ISSN 1392-0561

Informacijos mokslai

Mokslas • Ekonomika • Sanglauda
Europos Sąjunga
Europos socialinė fondas
Kuriame Lietuvos ateitį

ISSN 1392-0561

VILNIUS UNIVERSITY

INFORMATION SCIENCES

2013
Vol. 65

Research papers

Published since 1994

Vilnius University Publishing House
2013

ISSN 1392-0561

VILNIAUS UNIVERSITETAS

INFORMACIJOS MOKSLAI

2013
65 tomas

Mokslo darbai

Leidžiami nuo 1994 metų

Vilniaus universiteto leidykla
2013

Redaktorių kolegija (Editorial Board)

Vyriausioji redaktorė (Editor-in-Chief)

Prof. dr. *Zenona Atkočiūnienė* — Vilniaus universitetas, Lietuva (Komunikacija ir informacija 06H)
Vilnius University, Lithuania (Communication and Information 06H)

Vyriausiosios redaktorės pavaduotoja (Deputy Editor-in-Chief)

Prof. dr. *Elena Macevičiūtė* — Vilniaus universitetas, Lietuva; Švedijos bibliotekininkystės ir informacijos mokslų mokykla, Švedija (Komunikacija ir informacija 06H)
Vilnius University, Lithuania; The Swedish School of Libary and Information Studies Communication and information, Sweden (Communication and Information 06H)

Atsakingoji sekretorė (Executive Secretary)

Doc. dr. *Renata Matkevičienė* — Vilniaus universitetas, Lietuva (Komunikacija ir informacija 06H)
Vilnius University, Lithuania (Communication and Information 06H)

Nariai (Editors):

Prof. dr. *Arūnas Augustinaitis* — Kazimiero Simonavičiaus universitetas, Lietuva (Komunikacija ir informacija 06H)
Kazimieras Simonavicius University of Lithuania (Communication and Information 06H)

Prof. dr. *Auksė Balčytienė* — Vytauto Didžiojo universitetas, Lietuva (Komunikacija ir informacija 06H)
Vytautas Magnus University, Lithuania (Communication and Information 06H)

Prof. dr. *Saulius Gudas* — Vilniaus universiteto Kauno humanitarinis fakultetas, Lietuva (Informatikos inžinerija 07T; Informatika 09P)
Vilnius University, Kaunas Faculty of Humanities (Informatic Engineering 07T; Informatic 09P)

Prof. dr. *Renaldas Gudauskas* — Lietuvos nacionalinė Martyno Mažvydo biblioteka, Lietuva (Komunikacija ir informacija 06H)
Martynas Mažvydas National Library of Lithuania (Communication and Information 06H)

Prof. *Cees J. Hamelink* — Amsterdamo universiteto Komunikacijos ir žmogaus teisių centras, Nyderlandų Karalystė (Komunikacija ir informacija 06H)
University of Amsterdam, Centre of Communication and Human Rights (Communication and Information 06H)

Prof. *Lars Höglund* — Geteborgo universitetas, Švedija (Sociologija, Socialiniai mokslai 05S)
Göteborgs University, Sweden (Sociology, Social Sciences 05S)

Prof. *Epp Lauk* — Tartu universitetas, Estija (Komunikacija ir informacija 06H)
Tartu University, Estonia (Communication and Information 06H)

Prof. *Jaako Lehtonen* — Juveskiulės universitetas, Suomija (Komunikacija ir informacija 06H)
University of Jyväskylä, Finland (Communication and Information 06H)

Prof. dr. *Albinas Marčinskas* — Vilniaus universitetas, Lietuva (Vadyba ir administravimas 03S)
Vilnius University, Lithuania (Management and Administration 03S)

Doc. dr. *Audronė Nugaraitė* — Vytauto Didžiojo universitetas, Lietuva (Komunikacija ir informacija 06H)
Vytautas Magnus University, Lithuania (Communication and Information 06H)

Doc. dr. *Marija Prokopčik* — Vilniaus universiteto biblioteka, Lietuva (Edukologija 07S)
Library of Vilnius University, Lithuania (Educology 07S)

Prof. dr. *Marija Stonkienė* — Vilniaus universitetas, Lietuva (Komunikacija ir informacija 06H)
Vilnius University, Lithuania (Communication and Information 06H)

Prof. dr. *Marius Povilas Šaulauskas* — Vilniaus universitetas, Lietuva (Filosofija 01H)
Vilnius University, Lithuania (Philosophy 01H)

Prof. habil. dr. *Laimutis Telksnys* — Matematikos ir informatikos institutas, Lietuva (Fiziniai mokslai, Informatika 09P)
Institute of Mathematics and Informatics, Lithuania (Physical Sciences, Informatic 09P)

Prof. *Tom Wilson* — Šefildo universitetas, Didžioji Britanija (Komunikacija ir informacija 06H)
Sheffield University, Great Britain (Communication and Information 06H)

Prof. *Minka Zlateva* — Sofijos Šv. K. Ohridskio universitetas, Bulgarija (Komunikacija ir informacija 06H)
University of Sofia "St. Kl.Ohridski", Bulgaria (Communication and Information 06H)

Redakcijos adresas (Address):
„Informacijos mokslų" redaktorių kolegija (Editing board of "Informacijos mokslai")
Vilniaus universiteto Komunikacijos fakultetas (Faculty of Communication, Vilnius University)
Adresas (Address): Saulėtekio al. 9, I rūmai, 603 kab., LT-10222 Vilnius, Lithuania
Tel. (Ph.) (+370 5) 236 61 19; Faks. (Fax) (+370 5) 236 61 04
El. paštas (e-mail): zenona.atkociuniene@kf.vu.lt; informacijos.mokslai@kf.vu.lt
Elektroninio leidinio „Informacijos mokslai" (ISSN 1392-1487)
adresas internete (Electronic journal „Informacijos mokslai" (ISSN 1392-1487)
on the Internet): http://www.leidykla.eu/mokslo-darbai/informacijos-mokslai/

© Vilniaus universitetas, 2013
© Vilnius University, 2013

TURINYS

INFORMACINĖS VISUOMENĖS PLĖTRA

Saulė Jokūbauskienė. Lyderio gyvavimo organizacijoje proceso kompleksinės analizės modelis ir sistema 7

INFORMACINĖS TECHNOLOGIJOS IR KALBA

Pavel Stefanovič, Olga Kurasova. Tekstinių dokumentų panašumų paieška naudojant saviorganizuojančius neuroninius tinklus ir *k* vidurkių metodą 24

INTERNETINĖS TECHNOLOGIJOS

Dalė Dzemydienė, Arūnas Miliauskas. Vartotojo valdomos internetinių paslaugų automatinės ir pusiau automatinės kompozicijos atlikimo galimybės 34

Aurimas Rapečka, Virginijus Marcinkevičius, Gintautas Dzemyda. Rekomendacinės sistemos algoritmų veikimo elektroninio knygyno duomenų bazėje analizė 45

PROGRAMAVIMAS IR PROGRAMINĖ ĮRANGA

Ernestas Filatovas, Grigorijus Melničenko, Vytautas Jakštys. OpenGL ir DirectX technologijų spartos palyginimas programuojant trimačius vaizdus 56

DUOMENŲ ANALIZĖ IR VAIZDAVIMAS

Olga Kurasova, Virginijus Marcinkevičius, Viktor Medvedev, Aurimas Rapečka. Duomenų tyrybos sistemos, pagrįstos saityno paslaugomis 66

Kristina Lapin, Sigitas Dapkūnas. Duomenų apdorojimas ir vizualizavimas mobiliuosiuose įrenginiuose 75

Kotryna Paulauskienė, Olga Kurasova. Duomenų tyrybos sistemų galimybių tyrimas įvairių apimčių duomenims analizuoti 85

Gediminas Bazilevičius. Mobiliosios duomenų tyrybos paslaugų analizė 96

KOMPIUTERINIS MODELIAVIMAS

Jonas Mockus, Igor Katin, Joana Katina. Apie eksperimentinį investavimo strategijų tyrimą realiose ir virtualiose finansų rinkose 103

Audronė Lupeikienė, Jolanta Miliauskaitė, Albertas Čaplinskas. Trijų dominuojančių paslaugų kokybės modeliavimo būdų kritinė analizė ir vertinimas 111

Rūta Simanavičienė. Statistinių metodų taikymas daugiatikslių sprendimų patikimumui įvertinti 120

CONTENTS

INFORMATION SOCIETY DEVELOPMENT

Saulė Jokūbauskienė. The complex analysis model and the system of a leader's life cycle process in an organization 7

INFORMATION TECHNOLOGIES AND NATURAL LANGUAGE

Pavel Stefanovič, Olga Kurasova. Similarity analysis of text documents by self-organizing maps and *k*-means..... 24

INTERNET TECHNOLOGIES

Dalė Dzemydienė, Arūnas Miliauskas. Possibilities of automatic and semi-automatic end-user driven web service composition..... 34

Aurimas Rapečka, Virginijus Marcinkevičius, Gintautas Dzemyda. Analysis of the efficiency of recommendatory systems algorithms in an e-bookshop..... 45

INFORMATION SYSTEMS AND MODELING

Ernestas Filatovas, Grigorijus Melničenko, Vytautas Jakštys. Comparison of OpenGL and DirectX technologies in 3D programming..... 56

DATA ANALYSIS AND VISUALISATION

Olga Kurasova, Virginijus Marcinkevičius, Viktor Medvedev, Aurimas Rapečka. Data mining systems, based on Web services..... 66

Kristina Lapin, Sigitas Dapkūnas. Data Processing and Visualization on Mobile Devices..... 75

Kotryna Paulauskienė, Olga Kurasova. Investigation of the abilities of data mining systems to analyse various volume datasets..... 85

Gediminas Bazilevičius. Analysis of services for mobile data mining..... 96

COMPUTER-AIDED MODELING

Jonas Mockus, Igor Katin, Joana Katina. On the experimental investigation of investment strategies in the real and virtual financial markets..... 103

Audronė Lupeikienė, Jolanta Miliauskaitė, Albertas Čaplinskas. Critical analysis and elaboration of three prevailing approaches to model quality of service..... 111

Rūta Simanavičienė. Application of statistical methods to assess the reliability of the multiple attributes decisions..... 120

INFORMACINĖS VISUOMENĖS PLĖTRA

Lyderio gyvavimo organizacijoje proceso kompleksinės analizės modelis ir sistema

Saulė Jokūbauskienė

Vilniaus universiteto Komunikacijos fakulteto Informacijos ir komunikacijos katedros doktorantė
Departament of Information and Communication studies, Faculty of Communication, Vilnius University, Doctoral student
Saulėtekio al. 9, LT-10222 Vilnius
El. paštas: g.saule@gmail.com

Šio straipsnio tyrimų objektą sudaro lyderio gyvavimo organizacijoje procesas, jame dalyvaujančios ir savo tikslus norinčios įgyvendinti suinteresuotos grupės bei išorinė mikro-, mezo- ir makrolygmens aplinka kaip visuma. Siekiant atlikti šio tyrimų objekto išsamią analizę buvo adaptuoti projektų daugiakriterinės analizės metodai. Juos taikant buvo kompleksiškai įvertinti jo vadybiniai, organizaciniai, ekonominiai, socialiniai, teisiniai, emociniai, psichologiniai ir kiti aspektai. Straipsnyje pateiktas Lyderio gyvavimo organizacijoje proceso kompleksinės analizės modelis ir sistema bei šios sistemos taikymo praktinis pavyzdys.

Įvadas

Lyderio modelius savo darbuose analizavo įvairių šalių mokslininkai (Kim, 2012; Vargas, Torres, 2008; Chen et al., 2013, Day, Sin, 2011, Tavakoli, Vahidi, 2010, Liu et al., 2007; Ashtiani et al., 2013; Six, 2007; Sense, 2003). Šiuos mokslinius tyrimus galima skirstyti į keturias grupes, t. y. skirtus:

- lyderio gyvavimo proceso konkrečios stadijos konkretiems uždaviniams detaliai spręsti;
- konkrečiai problemai spręsti per visą lyderio gyvavimo procesą;
- lyderio gyvavimo proceso kompleksiniam efektyvumui didinti;
- lyderio gyvavimo proceso ar jo dalies efektyvumui didinti naudojantis naujausiomis intelektinėmis, biometrinėmis, išplėstinės ir virtualios realybės bei kitomis technologijomis.

Kuriamos įvairios lyderystės analizės informacinės (Cho et al., 2011; Dimitrios et al., 2013), intelektinės (Seah et al., 2010; Rao et al., 1994), žinių (Chun, Tak, 2009; McKenna et al., 2009), ekspertinės (Lehner, 1992; Oliveira et al., 2012) ir sprendimų paramos (Rees and Koehler, 2000; Lim et al. 1994) sistemos.

Šio straipsnio autorės atliekami darbai priklauso pirmajai (moksliniai tyrimai lyderio gyvavimo proceso konkrečios stadijos konkretiems uždaviniams detaliai spręsti) ir trečiajai (moksliniai tyrimai lyderio gyvavimo proceso kompleksiniam efektyvumui didinti) anksčiau analizuotoms mokslinių tyrimų grupėms. Galima pabrėžti, kad įvairių šalių mokslininkai nagrinėdami lyderio gyvavimo proceso ar jo sudedamųjų dalių kompleksinio projektavimo klausimus savo darbuose neanalizavo tokio tyrimo objekto kaip autorė: lyderio gyvavimo organizacijoje proceso, jame dalyvaujančių suinteresuotų grupių ir proceso efektyvumą veikiančios

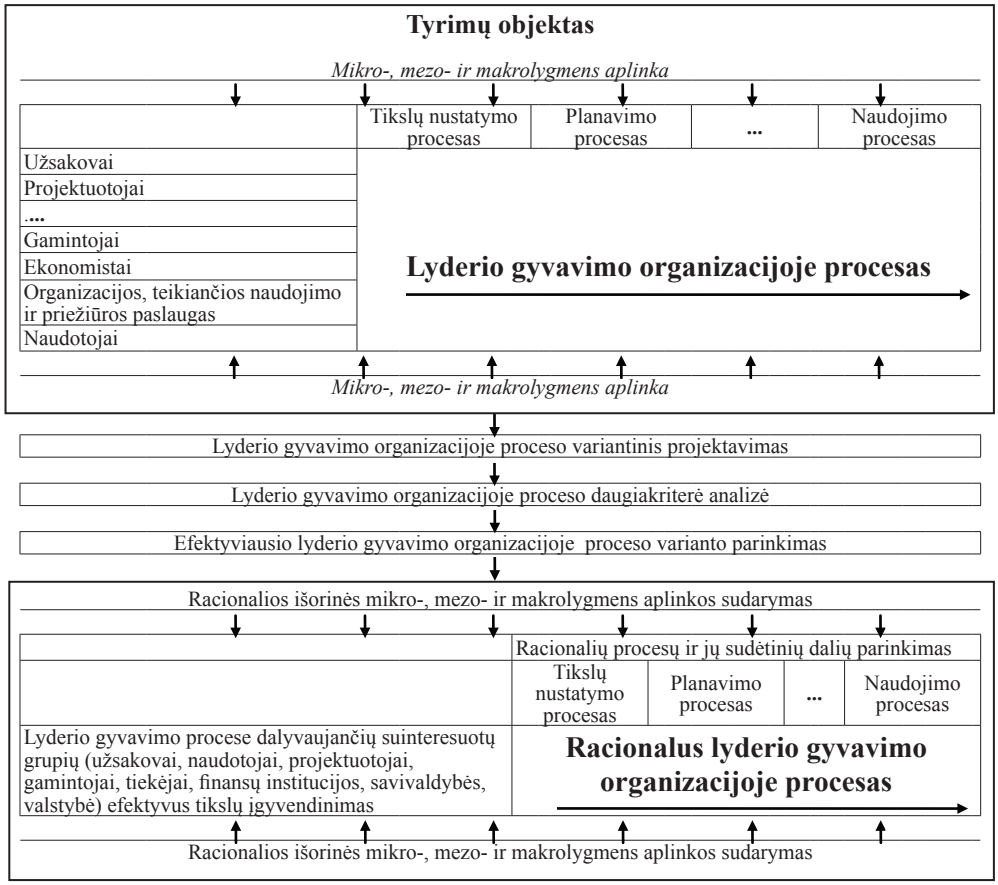
išorinės mikro-, mezo- ir makrolygmens aplinkos kaip visumos. Šiai tyrimų objekto kompleksinei analizei atlikti buvo pritaikyti projektų daugiakriterinės analizės metodai. Šiuo metu autorė kartu su kolegomis atlieka ketvirtajai grupei priklausančius teorinio lygmens mokslinius tyrimus.

Straipsnio pirmame skyriuje aprašomas pasiūlytas Lyderio gyvavimo organizacijoje proceso kompleksinės analizės modelis, kuriame ypatingas dėmesys kreipiamas į lyderio gyvavimo organizacijoje procesą bei jį veikiančią mikro-, mezo- ir makrolygmens aplinką. Antrame skyriuje analizuojama sukurta Daugiakriterinė lyderio gyvavimo ciklo organizacijoje analizės sprendimų paramos sistema. Šiame skyriuje detaliau aprašoma Sistemos duomenų bazė ir duomenų bazės valdymo sistema, modelių bazė ir modelių bazės valdymo sistema. Trečiame skyriuje pateikiamas Modelį ir Sistemą iliustruojantis praktinis pavyzdys. Straipsnio pabaigoje pateikiamos tyrimo išvados.

1. Lyderio gyvavimo organizacijoje proceso kompleksinės analizės modelis

Lyderio gyvavimo procesas turi būti projektuojamas ir vykdomas atsižvelgiant į jame dalyvaujančių suinteresuotų grupių tikslus, galimybes bei išorinę mikro-, mezo- ir makrolygmens aplinką, veikiančią organizacijos efektyvumą.

Siekiant įgyvendinti lyderio gyvavimo proceso organizacijoje tikslus buvo sukurtas Lyderio gyvavimo organizacijoje proceso kompleksinės analizės modelis (1 pav.), apimantis



1 pav. ***Lyderio gyvavimo organizacijoje proceso kompleksinės analizės modelis***

lyderio gyvavimo organizacijoje procesą, jame dalyvaujančių ir savo tikslus norinčių įgyvendinti suinteresuotų grupių bei išorinės aplinkos kaip visumos variantinį projektavimą, daugiakriterinę analizę ir efektyviausių variantų nustatymą.

Siekiant detaliau išanalizuoti pasiūlytą Modelį, būtina trumpai apžvelgti lyderio gyvavimo procesą organizacijoje, jį veikiančią mikro-, mezo- ir makrolygmens aplinką.

Lyderis nuolat vykdo aukštos kvalifikacijos ir kompetencijos reikalaujančius projektus. Šiame skyriuje projektas suprantamas plačiąja prasme, kaip visuma nuolatos besikeičiančių projektų, kuriuos vykdo tas pats asmuo per visą savo darbinės veiklos gyvavimo procesą (pavyzdžiui, projekto darbuotojas, projekto vadovas ir projekto direktorius).

Kadangi projekto skirtingų sudedamųjų dalių racionalumas dažnai priklauso nuo skirtingų *suinteresuotų grupių* (užsakovai, savininkai, vartotojai, finansų institucijos, gamintojai, tiekėjai, valstybė ir jos įgaliotos institucijos, savivaldybės ir t. t.), todėl tik atliekant projekto gyvavimo proceso kompleksinį projektavimą, kurio metu artimai sąveikauja pagrindinės suinteresuotos grupės, galima pasiekti gerų rezultatų. Tikslų nustatymo, projektavimo, gamybos ir naudojimo procesuose dalyvauja įvairios suinteresuotos grupės, sąveikaujančios tarpusavyje gana ilgą laiką.

Vienas iš svarbiausių uždavinių siekiant padidinti projekto gyvavimo proceso efektyvumą yra kuo geriau įgyvendinti jame dalyvaujančių suinteresuotų grupių tikslus. Daugelis suinteresuotų grupių projekto efektyvumo laipsnį supranta kaip santykį tarp savo indėlio (panaudotų išteklių, rizikos), įdėto didinant projekto racionalumą, ir dėl jo pasiektų tikslų (ekonominių, teisinių, socialinių) laipsnio. Siekiama, kad kiekvienos suinteresuotos grupės pastangomis padidintas projekto efektyvumas būtų tiesiogiai proporcingas jos pasiektiems tikslams ir patenkintiems poreikiams. Dažniausiai ir vyrauja būtent toks supratimas, jeigu suinteresuotos grupės traktuoja „dalijamą pyragą“ (pasiekti projekto tikslai) kaip kintamą dydį. Suinteresuotos grupės, norėdamos padidinti „dalijamą pyragą“, stengiasi padidinti projekto efektyvumą. Todėl „pyrago“ (įskaitant jo padidėjimą) dalybas galima traktuoti kaip teigiamą tendenciją. Vadinasi, tokiu būdu siekiama sukurti efektyvesnę suinteresuotų grupių organizavimo sistemą.

Pagrindinės suinteresuotos grupės dalyvauja esminėse projekto gyvavimo proceso stadijose. Todėl esama glaudžios sąveikos tarp įvairių suinteresuotų grupių ir tarp atskirų projekto gyvavimo proceso etapų. Kadangi tiktai visų projekto gyvavimo proceso etapų ir juose dalyvaujančių suinteresuotų grupių visuma išsamiai apibūdina projektą, todėl siekiant užtikrinti projekto efektyvumą būtina atlikti jo variantinį projektavimą ir kompleksinę analizę.

Lyderio gyvavimo proceso efektyvumo lygis priklauso nuo trijų tarpusavyje susijusių mikro-, mezo- ir makrolygmenų kintamųjų skaičiaus.

1.1. Makrolygmens veiksniai

Makrolygmens veiksniai (šalies teisinė, ekonominė, politinė, socialinė ir kultūrinė, religinė aplinka, demografinė situacija, nacionalinė ir religinė gyventojų sudėtis; tolerancija, tradicijos ir t. t.) veikia visos šalies ar pramonės efektyvumo lygį. Priklausomai nuo bendro šių makrolygmens veiksnių poveikio kinta lyderio gyvavimo proceso efektyvumo lygis. Susidomėjimas naujomis lyderių kompetencijomis kyla iš socialinių ir ekonominių pokyčių: rinkų globalizacijos ir pasaulinės konkurencijos, produktų plėtojimo ciklų trumpėjimo, demografinių pokyčių labiausiai industrializuotose pasaulio šalyse, žinių gyvavimo ciklo sumažėjimo ir kt. Kaip pavyzdys, toliau bus trumpai aptarta ekonominės, politinės, socialinės ir kultūrinės aplinkos, globalizacijos proceso įtaka lyderio gyvavimo proceso efektyvumui.

Naujausi Europos Sąjungos dokumentai „Europa 2020“ nubrėžė akivaizdžias žinių visuomenės plėtros kryptis bei socialinius iššūkius. Žinių visuomenė šalia saugios visuomenės ir konkurencingos ekonomikos yra pirmas iš trijų Lietuvos valstybės prioritetų, įtvirtintų Valstybės ilgalaikės perspektyvos tikslus

apibrėžiančiame dokumente – Valstybės ilgalaikės raidos strategijoje. Žinių visuomenėje išskirtinę reikšmę įgyja žinios ir kompetencija. Šiuolaikiniame pasaulyje siekis ugdyti žinių visuomenės kompetencijas, sukurti globalią žiniomis grįstą ekonomiką – vienas svarbiausių informacinės ir žinių visuomenės politikos tikslų (Rychen, Sagalnik, 2002).

Ir prieš ekonominę krizę, ir pokriziniu laikotarpiu laikomasi nuostatos, kad investicijos į žmones ir gerovės valstybės kūrimas yra pagrindiniai ES tikslai (*2009/C 184 E/06*). Žinių visuomenėje pasitikėjimas, atsakomybė, motyvacija ir pagarba – būtinos sąlygos užtikrintai žinių sklaidai socialinių sąveikų metu (Castells, 2007). Kūrybiškas požiūris į inovacijas, organizuoti žinių tinklai, kaupiama patirtis, ugdomos ir plėtojamos kompetencijos ir organizacijai, ir individui leidžia aktyviai dalyvauti socialinio kapitalo mainuose (Harman, Brelade, 2000).

Lisabonos strategijos (2000) siektini tikslai didinti Europos konkurencingumą išlieka, tačiau atsiranda ir naujų – pavyzdžiui, informacinė visuomenė visiems, e. įterpties mažinimo deklaracijos.

Žinių ekonomika žymi tam tikrą jos kokybę ir plėtros tarpsnį. Žinių ekonomikai būdinga tai, kad itin svarbiu gamybos veiksniu tampa žinios: gamybos technologijos pramonėje, pardavimo technologijos prekyboje, klientų aptarnavimo metodai teikiant paslaugas. Žinių ekonomikoje svarbi inovacijų – žinių virtimo verslo sprendimais problema, inovacijas suprantant plačiai, ne tik kaip aukštąsias technologijas.

Žinių ekonomikos sąlygomis smarkiai augantis žinių kiekis, fragmentacijos ir globalizacijos procesai daro įtaką dinamiškam kolektyviniam ir individualiam konkurencingumui (Probst, Raub, Romhardt, 2006). Kūrybingumas ir inovatyvumas – sumanios ekonomikos plėtros pagrindas (Lietuva 2030).

Vienas esminių vieningos Europos kaip sąjungos išlikimo būdų – keliant el. gebėjimų lygį išlikti konkurencinga žinių ekonomika pasaulyje. INSEAD teigimu, koreliacija tarp skaitmeninių gebėjimų ir globalaus konkurencingumo siekia 85 proc. Skaitmeninis raštingumas kaip gyvenimo gebėjimas yra esminis atramos taškas tęstinėje visuomenės socialinėje-ekonominėje sistemoje (Ferrari, 2012). Kritinės skaitmeninės kompetencijos, kurias akcentuoja informacijos ir ryšių technologijų politikos nuostatos, yra žinių visuomenės ateities darbų sėkmės veiksnys. Verslumo kompetencija rekomendacijose išskirta kaip viena esminių mokymosi visą gyvenimą kompetencijų (Barkauskaitė, Gudžinskienė, 2007, p. 25–29).

Postmodernios visuomenės kontekste dalykinės veiklos prigimtis iš esmės keičiasi, globalizacijos tendencijos paverčia pasaulį vadinamuoju globaliu kaimu, žmogiškosios veiklos pridėtinė vertė tampa vis labiau susijusi su dalykinės kompetencijos verte (Love, Fong, Irani, 2005). Globalizacijos kontekste naujovių sūkurys ir sparti plėtra skatina mokytis nuolat, t. y. visą gyvenimą.

Globalus konkurencingumas kuria darbo rinkos lankstumą ir atsakomybę, kurioje žinios ir gebėjimai yra labiausiai vertinami konkurenciniai pranašumai (Gurría, 2011). Globalizacija atspindi labai paplitusį požiūrį, kad pasaulis ekonominių ir socialinių jėgų yra sulydomas į socialinę erdvę ir kad procesai, vykstantys viename regione, gali turėti didžiulę įtaką kito Žemės regiono žmonėms ir bendruomenėms.

Efektyvios organizacinio mokymosi strategijos tampa vis svarbesnės dėl didėjančio aplinkos kompleksiškumo, neapibrėžtumo, pokyčių greičio. Veikla tokioje aplinkoje sudėtinga, tam reikia naujų įgūdžių, žinių, greitai prisitaikyti prie aplinkos ir panaudoti visas įmanomas kompetencijas (Armstrong, 2001).

Socialinės aplinkos poveikis turi įtakos individo gebėjimų raiškai. Modernus socialinis gyvenimas remiasi ne tradicijomis, o nuolat kuriamu ir atnaujinamu refleksyviu žinojimu (Rychen, Sagalnik, 2002). Ankstesnėse premoderniose arba tradicinėse visuomenėse socialiniams pasikeitimams darė įtaką struktūros, o rizikos visuomenėje, arba refleksyviosios modernybės visuomenėje, vyrauja individualizacijos procesai, kuriantys individualius, suasmenintus pasakojimus. Perspektyvių transformacijų identifikavimas refleksyvios modernybės kontekste

skatina individą naudoti taktines žinias, kurių išraiška kinta improvizuojant ir inovatyviai reaguojant į potencialias grėsmes (Gudauskas, Kaušylienė, 2010).

Kompetencijos itin priklauso nuo šalių lyderystės kultūros. JAV yra individualistinė visuomenė, pasižyminti menku netikrumo vengimu ir nedideliu galios atstumu, taip pat tai vyriška visuomenė. Tačiau JAV išsiskiria aukštu individualizmo balu. Tarp vertintų šalių JAV ne kartą gavo individualistiškiausios įvertinimą. Nors menkas netikrumo vengimas ir nedidelis galios atstumas skatintų dalyvaujamąjį ir palaikomąjį valdymo stilių, dėl aukšto individualizmo balo ir JAV būdingos labiau vyriškos nei moteriškos visuomenės galima tikėtis, kad didesnį bendradarbiavimą užtikrintų nurodomasis vadovavimo stilius.

Suomija yra individualistinė ir turi menką netikrumo vengimą, bet abiem atvejais tai ne kraštutiniai variantai. Galios atstumo balas labai mažas, o visuomenė labai moteriška ir tai rodo, kad čia labiau tiktų dalyvaujamojo ir palaikomojo vadovavimo stiliai, dėl itin moteriškos visuomenės pabrėžiant dalyvaujamąjį stilių.

Lietuva pagal individualizmo, netikrumo vengimo ir galios atstumo dėmenis yra ties viduriu, bet kontinuumuose linksta į individualistinius, menkai išreikšto netikrumo vengimo ir nedidelio galios atstumo kraštus. Lietuva išsiskiria labai vyriška visuomene, taigi galima tikėtis, kad šioje visuomenėje nurodomasis vadovavimo stilius būtų tinkamesnis siekiant didinti bendradarbiavimą. Dėl menko netikrumo vengimo galima tikėtis, kad dalyvaujamojo vadovavimo stilius turėtų skatinti didesnį bendradarbiavimą nei palaikomasis vadovavimo stilius.

1.2. Mikrolygmens ir mezolygmens kintamieji veiksniai

Lyderio gyvavimo organizacijoje proceso efektyvumo lygis taip pat priklauso nuo mikro- ir mezolygmens kintamųjų veiksnių (veiklos srities pasirinkimas, infrastruktūros, informacinės sistemos, organizacijų susivienijimų veikla, integruotas projekto gyvavimo procesas, tiekimo proceso efektyvumo didinimas naudojantis internetu, emocijos, lūkesčiai, požiūris, gebėjimai, motyvacija, efektyvios suinteresuotos grupės (gyventojai, žiniasklaida, užsakovai, naudotojai), aktyvi, sąmoninga, socialinio konflikto nesuskaidyta ir integruota vietos bendruomenė; lyderių geri asmeniniai ryšiai, tiesioginiai kontaktai; gebėjimas pritraukti gerus darbuotojus; darbo jėgos kvalifikacijos lygis, inovatyvumas, išskirtinės žinios ir gebėjimai; taikoma pažangi technika ir technologija, organizacija, vadyba; mokymosi ir studijų visą gyvenimą galimybės, gyventojų kūrybiškumas, talentas ir pan.), kurie savo ruožtu priklauso nuo makrolygmens veiksnių įtakos.

Pavyzdžiui, mokesčių sistema, nustatyta makrolygmeniu, vadovaujantis vyriausybės fiskaline politika, daro tiesioginį poveikį lyderio darbo užmokesčio dydžiui mikrolygmeniu (projekto lygmeniu). Valstybės požiūris į konkrečią veiklą (įvairūs įstatymai ir nutarimai, mokesčių lengvatos, dotacijos, subsidijos ir pan.) turi daug įtakos organizacijų efektyvumui. Įvairių suinteresuotų grupių (pavyzdžiui, užsakovo ir rangovo) santykius tiesiogiai reguliuoja įstatymai. Kaip pavyzdys, toliau trumpai bus pristatyta lyderio emocijų, lūkesčių, požiūrio, gebėjimų, motyvacijos, organizacijos gebėjimų ir požiūrio, besimokančios organizacijos įtaka projekto efektyvumui.

Rinkos veikėjams trūksta informacijos apie tam tikrų veiksmų keliamą riziką ir galimybes, todėl gali būti priimami iracionalūs sprendimai. Tokie sprendimai dažnai priimami net ir turint pakankamai informacijos. Tuomet vietoje optimalaus naudingumo gali būti priimama emocinės raiškos įtaka arba vadovaujamasi apskaičiavimo strategijos rodikliais (Slater, Tonkiss, 2004).

Vidutinių lūkesčių sąvoka suponuoja galimus vidutinius gebėjimus arba esmines kompetencijas, įvertinant jų laipsnį vidutiniškai. Tikėtinos naudos teorija, lošimų teorija operuoja strateginio racionalumo sąvokomis ieškant optimalaus sprendimo, kai siekiama paskirstyti žinių visuomenės išteklius, juos pritaikant prie esamo konteksto. Rizikos sąlygomis veikiantis individas yra pasirengęs priimti sprendimus.

Tačiau tikėtinos naudos teorijoje svarbi individo nuostata rizikos atžvilgiu – išaiškėja polinkis rizikuoti, rizikos vengimas arba neutralumas rizikai (Norkus, 2003). Tikslingų žinių potencialas socialinių struktūrų atžvilgiu leidžia individui ir kartu organizacijai išvengti iracionalios elgsenos nulemtų sprendimų.

Analitiškai, pragmatiškai, vizualiai, holistiškai, originaliai, kūrybiškai mąstantis individas turi tamsiąją kompetencijų pusę – tai yra rizika, kuri individo stiprybes gali paversti silpnybėmis. Atviras ir inovatyvus mąstymas skatina ieškoti originalių sprendimų ir kūrybiškai generuoja naujas idėjas, kurios gali pasižymėti nepraktiškumu ir patirti nesėkmę. Holistinis, visa apimantis, aplinkos supratimas leidžia nustatyti įvykių grandinės priežastinius ryšius, tačiau gali turėti įtakos baimei veikti. Rizika kaip atsitiktinumas kyla dėl savireferencijos, kai tam tikru lygmeniu mes žinome per daug, todėl atsitiktinė procesų rizika sustiprina nesėkmės faktoriaus galimybę. Naujos pasirinkimo galimybės atnaujina senų žinių aktualumą, kitaip tariant, vienoje situacijoje silpnybės gali virsti stiprybėmis, ir atvirkščiai. Pernelyg patikėjusiam savo sėkme individui kyla grėsmė būti egoistiškam ir arogantiškam, o tai, kaip ir emocinis fonas, trukdo racionaliai mąstyti. Vizualus išplėstinis socialinės sąveikos aplinkos matymas gali trukdyti susitelkti į strateginius tikslus (Apgar, 2007). Emocinis intelektas socialinės sąveikos metu skatina pasitikėjimą ir motyvuoja siekti aukštesnių tikslų (Harman, Brelade, 2000).

Netikrumas yra pažintinė būsena, kuri paprastai reiškia susidomėjimo ir pasitikėjimo veiksnių įtaką. Ryškūs netikrumo požymiai yra sumaištis ir chaosas, kurie susiję su netolygiu mąstymu kuria nors tema. Netikrumas sukelia prasmės supratimo trūkumą (Giddens, 2000). Mokymasis pats savaime yra nepakankamas konkurenciniam pranašumui įgyti. Kiekvienas mokymasis turi būti orientuotas į aktualią dominuojančią konkurenciją. Siekdama sukurti ir palaikyti konkurencinį pranašumą, organizacija turi įgyti kompetencijų kūrimo dinamiką.

Besimokanti organizacija pasižymi kultūra, kurioje individualus tobulėjimas yra prioritetas, o organizacijos tikslas ir vizija yra aiškiai suprantami ir palaikomi jos narių. Kaip besimokančias būtų galima apibrėžti tokias organizacijas, kuriose asmeninis ugdymasis yra integruota visos organizacijos dalis, mokymasis – nuolatinis procesas. Pagrindiniai besimokančios organizacijos objektai yra transformacija, pokytis, delegavimas, inovacija, darbo stiliaus keitimas, adaptacija, vadybos stilius, dalyvavimas, darbuotojų dalyvavimo skatinimas (Love, Fong, Irani, 2005).

Lyderio gyvavimo procesas

Lyderio gyvavimo procesas kaip pavyzdys pateikiamas 1 lentelėje. Čia lyderis yra projekto darbuotojas, projekto vadovas ir projekto direktorius. Lentelėje nurodoma, kaip keitėsi jam būtinos vadybos, laiko, išlaidų, kokybės ir žmogiškųjų išteklių vadybos kompetencijos. Kaip matyti iš pateiktų kompetencijų, lyderiui einant vis atsakingesnes pareigas, jo kompetencijos sudėtingėjo ir plėtėsi.

Norint užtikrinti vykdomo projekto efektyvumą, reikia atsižvelgti į mikro-, mezo- ir makrolygmens veiksnių nustatytas sąlygas. Keičiantis šiems veiksniams, keičiasi ir lyderiui keliami reikalavimai.

Akivaizdu, kad lyderio veiklos galimybės bei efektyvumo lygis labai priklauso nuo racionaliai parinktų (esamų) mikro-, mezo- ir makroaplinkos kintamųjų veiksnių. Kitaip tariant, esant konkretiems mikro-, mezo- ir makrolygmens kintamiesiems veiksniams, iš karto objektyviai apribojami lyderio kompetencijų reikalavimai. Suprantama, kai lyderio kompetencijų poreikis priklauso nuo konkrečios mikro-, mezo- ir makrolygmens aplinkos, geriausia savo veiklą plėtoti tose srityse, šalyse ir geografinėse vietovėse (sostinė, įvairūs šalies miestai ir rajonai, kaimo vietovės ir t. t.) ir su tomis suinteresuotomis grupėmis, kur visų šių grupių tikslai būtų maksimaliai įgyvendinti. Remdamiesi šiuo teiginiu, lyderiai stengiasi susikurti racionalias vidinės ir išorinės aplinkos veiklos sąlygas, kad kuo geriau tenkindami

klientų poreikius pelnytų geresnę reputaciją ir organizacija gautų daugiau pelno.

Vadovaujantis pateiktomis mintimis, galima aiškiau suformuluoti vieną iš šio tyrimo tikslų – pateikti efektyvaus lyderio gyvavimo proceso teorinio modelio projektą, remiantis racionalių mikro-, mezo- ir makrolygmens kintamųjų veiksnių paieška. Šiame darbe modelis suprantamas kaip žaidimų taisyklių sistema, kuriai esant lyderis geriausiai panaudotų savo potencialias galimybes. Tai siūloma atlikti nagrinėjant išsivysčiusių šalių patirtį bei žinias ir pritaikant Lietuvai. Analizuojant reikia sudaryti galimus lyderio strategijos variantus, juos įvertinti pagal daugiakriterinius analizės metodus bei išrinkti veiksmingiausius. Iš pradžių ekspertiniais metodais, nustatomi mikro-, mezo- ir makrolygmens veiksniai ir juos apibūdinančios kriterijų sistemos bei posistemiai, išsamiai apibūdinantys lyderio veiklą.

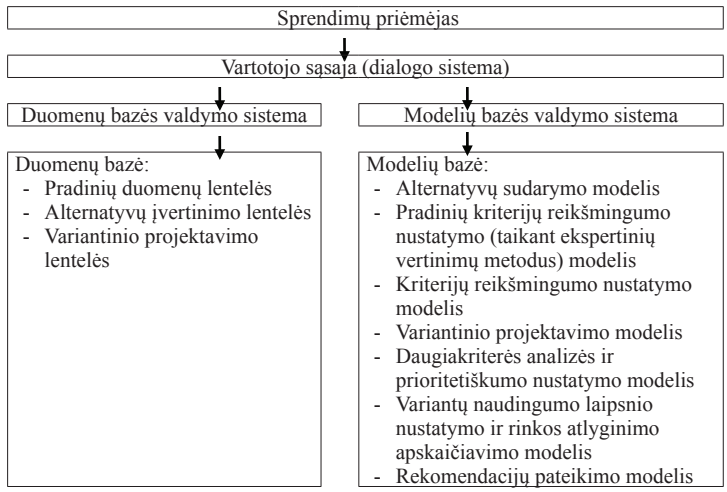
2. Daugiakriterinė lyderio gyvavimo ciklo organizacijoje analizės sprendimų paramos sistema

Remiantis atlikta informacinių, ekspertinių ir sprendimų paramos sistemų analize, sukurtu Lyderio gyvavimo proceso organizacijoje kompleksinės analizės modeliu, ankstesniame poskyryje aprašytais pagrindiniais principais ir metodais bei pirmame skyriuje pateiktais metodais, buvo sukurta Daugiakriterinė lyderio gyvavimo ciklo organizacijoje analizės sprendimų paramos (DLGCASP) sistema. Ši sistema sudaryta iš duomenų bazės ir duomenų bazės valdymo sistemos, modelių bazės ir modelių bazės valdymo sistemos, vartotojo sąsajos. Trumpai aprašysime DLGCASP sistemą.

2.1. Duomenų bazė

Lyderio gyvavimo organizacijoje proceso projekte dalyvauja suinteresuotos grupės (užsakovai, naudotojai, projektuotojai, gamintojai, eksploatavimo organizacijos, savivaldybės), kurių tikslai, galimybės, išsilavinimas ir patirtis skiriasi. Todėl priimant sprendimus dažnai nesutampa šių suinteresuotų grupių požiūriai. Siekiant visapusiškai apibūdinti nagrinėjamas alternatyvas ir priimti efektyvų kompromisinį sprendimą, dažnai būtina jas aprašyti remiantis ekonomine, teisine, socialine, estetine, komfortine, technine, technologine ir kitokia informacija (2 pav.). Ši informacija turi būti pateikiama vartotojui suprantamiausia forma.

DLGCASP sistemoje informacija, reikalinga sprendimams priimti, gali būti pateikta skaitmenine, tekstine, grafine (schemos, grafikai, diagramos, piešiniai, brėžiniai), formulių, fotografijos, garso, vaizdo, išplėstinės ir virtualios realybės bei kitokia forma (3 pav.). Pavyzdžiui, naudojant informacijos pateikimo skaitmeninę formą, pateikiama variantus išsamiai apibūdinanti kriterijų sistema, matavimo vienetai, reikšmės ir pradinis reikšmingumas. Naudojant informacijos tekstinę formą, pateikiamas alternatyvų ir jas išsamiai apibūdinančių kriterijų koncepcinis aprašymas, priežastys ir pagrindimas,



2 pav. ***Daugiakriterės lyderio gyvavimo ciklo organizacijoje analizės sprendimų paramos sistemos sudedamosios dalys***

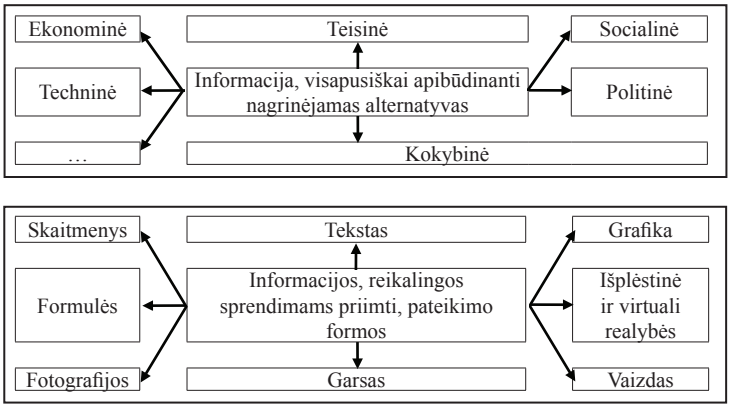
1 lentelė. ***Lyderio gyvavimo procesas organizacijoje ir kompetencijų kaita***

Projekto darbuotojas	Projekto vadovas	Projekto direktorius
Vadybos metodų taikymas	*Apimties planavimas ir vadyba*	*Apimtis: vadovavimas ir kontrolė*
Reikalingos tokios bendros žinios ir supratimas:		
• poreikis pradedant projektą nustatyti projekto apimtį • užduočių nustatymo svarba ir susiję metodai • vadybos vieta projekto gyvavimo ciklo kontekste • apimties pokyčių kontrolės priemonių išmanymas • produktų ir veiklos nustatymo metodai (pvz., paprasta darbų skirstymo struktūra).	• su projekto patvirtinimu, parengimu ir projekto vadovo paskyrimu susiję organizaciniai procesai • vadybos ir apimties kontrolės vieta projekto gyvavimo ciklo kontekste • poreikis apimtį nustatyti pradedant projektą ir per visą projekto gyvavimo ciklą • produktų ir veiklos nustatymo metodai (tarkime, darbo, organizacijos ir produktų suskirstymo struktūros) • apimties apibrėžimo komponentai • pokyčių kontrolės veikla	• vadybos vieta projekto gyvavimo ciklo kontekste, ypač ryšys tarp apimties, veiklos rezultatų, laiko, kainos ir kokybės • projekto pagrindimo ir patvirtinimo procesai • apimties nustatymo svarba, ypač pradedant projektą ir per visą projekto gyvavimo ciklą • užduočių nustatymo svarba ir metodai • projekto išteklių reikalavimų poveikis bendriems organizaciniams tikslams • pokyčių vadybos principai ir jų taikymas • tinkamos vadybos metodikos, metodai ir įrankiai, jų taikymas, galimybės ir trūkumai
Įgūdžiai:		
• planavimo • stebėjimo ir kontrolės • komandinio darbo ir bendravimo • įskaitant tokias savybes: ○ dėmesingumą smulkmenoms ○ tikslumą ○ kritinį mąstymą	• raštingumas verslo srityje (kurio pakaktų interpretuoti ir analizuoti projekto planus ir dokumentus) • planavimo ir organizavimo įgūdžiai (stebint apimtį ir reaguojant į potencialius ir realius pokyčius) • problemų sprendimo ir analitiniai įgūdžiai (sprendžiant projekto vadybos problemas ir sunkumus) • derybų įgūdžiai (su įvairiomis interesų grupėmis nagrinėjant apimties pasikeitimus)	• itin geras raštingumas (peržiūrint ir taisant projektų planus, pristatant sprendimus ir rašant ataskaitas) • technologiniai įgūdžiai (planuojant ir aprašant projekto apimtį) • bendravimo įgūdžiai (bendraujant su interesų grupėmis) • itin geri analitiniai, planavimo, organizavimo, problemų sprendimo ir vertinimo įgūdžiai (susiję su apimties vadyba) • asmeninės vadybos įgūdžiai (leidžiantys nustatyti veiksmų prioritetus siekiant sėkmingų rezultatų) • derybiniai įgūdžiai (leidžiantys kontroliuoti pasiūlytus apimties pakeitimus)
Laiko, kokybės ir išlaidų vadybos metodų taikymas	*Laiko, kokybės ir išlaidų planavimas ir vadyba*	*Bendrojo projekto tvarkaraščio vadyba / Kokybė ir išlaidos: vadovavimas ir valdymas*
Reikalingos tokios bendros žinios ir supratimas:		
• poreikis planuoti laiką ir išlaidas bei kokybę valdyti visoje projekto vadybos sistemoje • atitinkamos vadybos įrankių ir metodų taikymas asmeninės kompetencijos srityje • žinios apie procesus: kaip, kada ir kodėl diegiami tam tikros vadybos procesai • asmens indėlio į vadybos procesą svarba • darbų paskirstymo struktūros sąvokos, priklausomybės, išteklių paskirstymas, kritiniai keliai ir uždirbta vertė • sąvokos ir įgūdžiai (apibrėžiant ir išdėstant veiklą, apibrėžiant uždirbtą vertę, faktines ir planuotas išlaidas bei prognozavimą, sudarant pradinius projekto tvarkaraščius, išlaidų sąmatas ir biudžetą) • veiklai reikalingų išteklių ir kiekvienos veiklos rūšies trukmės nustatymas • išlaidų vadybos proceso duomenys, rezultatai, įrankiai ir metodikos • išlaidų ir kokybės vadybos vieta projekto gyvavimo ciklo ir kitų projektų vadybos kontekste	• poreikis laiką, išlaidas ir išteklius susieti su projekto ir projekto tvarkaraščiu • prisiimama atsakomybė už laiko, kokybės ir išteklių vadybą • projekto tvarkaraščių ir projekto biudžetų ir išlaidų prognozių sudarymas bei jų naudojimas kontrolei • laiko, kokybės ir išlaidų vadybos vieta projekto gyvavimo ciklo ir kitų projektų vadybos funkcijų kontekste • metodikos: tinkamos laiko, kokybės ir išlaidų vadybos metodikos, jų galimybės, trūkumai, taikymas ir poveikis • uždirbtos vertės ir kokybės vadybos principai ir veikla bei jų taikymas • išlaidų ir kokybės vadybos proceso duomenys, rezultatai, įrankiai ir metodai • sąvokos: išlaidų sąmatos ir biudžeto sudarymo, išlaidų kontrolės • planuota vertė, uždirbta vertė, faktinės išlaidos ir prognozavimas kontroliuojant išlaidas • kokybės audito procesai ir reikalavimai • kokybės procesai, kokybės planavimas, veiklos kokybės užtikrinimas, veiklos kokybės kontrolė • kokybės vadybos sistemų ir standartų naudojimas	• Principai ir jų taikymas: projektų, kokybės bei uždirbtos vertės vadybos, planuotos vertės, faktinių išlaidų ir prognozavimo principai kontroliuojant išlaidas, rizikos ir atlygio sutarčių administravimo principai • organizacinė politika, patarimai ir požiūris į laiko ir išlaidų vadybą • projektų vadybos sistemos ryšys su laiku, kaina ir ištekliais • atsakomybės už laiko ir išlaidų vadybą perdavimas ir vadyba • projekto tvarkaraščių ir biudžetų sudarymas ir jo naudojimas kontrolei • tinkamos laiko ir išlaidų vadybos ir įvertinimo metodikos, metodai ir įrankiai, jų galimybės ir trūkumai, taikymas ir poveikis • skirtumai tarp darbo turinio, rizikos, procesų, įrankių ir metodų, taikomų įvairiuose projekto gyvavimo ciklo etapuose • kokybės vadybos poreikis projekte ir jos vieta visoje organizacijoje • kokybės svarba renkantis kompromisus dėl laiko, kainos ir veiklos rezultatų, • kokybės politika ir standartai (taikomi projektų vadybos aplinkoje) • kokybės vadybos įrankiai, jų galimybės, trūkumai, taikymas ir poveikis • kokybės pasekmės, kritiniai sėkmės ir žlugimo kriterijai ir kokybiškos veiklos priemonės • pasauliniai kokybės standartai

Projekto darbuotojas	Projekto vadovas	Projekto direktorius
Laiko, kokybės ir išlaidų vadybos metodų taikymas	*Laiko, kokybės ir išlaidų planavimas ir vadyba*	*Bendrojo projekto tvarkaraščio vadyba / Kokybė ir išlaidos: vadovavimas ir valdymas*
Įgūdžiai:		

Įgūdžiai (reikalingi tik laiko vadybai):
• laiko vadybos
• planavimo
• stebėjimo ir kontrolės
• analizės ir vertinimo
• įskaitant tokias savybes:
  ○ dėmesingumą smulkmenoms
  ○ tikslumą
  ○ kritinį mąstymą
	• asmeninės vadybos įgūdžiai (užtikrinantys aiškią viziją ir prioritetus)
	• raštingumas (reikalingas skaitant, sudarant ir interpretuojant projekto tvarkaraščius, aprašant skirtumų priežastis ir rengiant finansines ataskaitas, kokybės uždavinius ir kriterijus, ataskaitas apie kokybės rezultatus) bei skaičiavimo ir biudžetų sudarymo įgūdžiai (stebint išlaidas ir valdant sąnaudas)
	• organizavimo įgūdžiai (tvarkaraštyje išdėstant užduotis ir rūpinantis uždavinių įvykdymu)
	• bendravimo įgūdžiai (perteikiant lūkesčius ir kitus informuojant apie pažangą, veiklos rezultatus, pristatant lūkesčius ir užtikrinant reikiamus rezultatus)
	• technologiniai įgūdžiai (naudoti reikiamą programinę įrangą sudarant projekto tvarkaraščius bei išlaidų registravimo ir finansinės atskaitomybės programinę įrangą)
	• analitiniai įgūdžiai (peržiūrint ir vertinant procesus, prognozuojant kainą, rekomenduojant patobulinimus, pasiektus projekto rezultatus lyginant su kokybės kriterijais)
		• itin geri lyderystės įgūdžiai (įkvepiantys visų interesų grupių pasitikėjimą ir pasikliovimą, padedantys laikytis kokybės etalonų)
		• raštingumas (reikalingas perduodant sprendimus ir rašant kokybiškas ataskaitas) bei skaičiavimo įgūdžiai (sudarant ir interpretuojant projekto biudžetus ir teikiant finansinę informaciją)
		• planavimo ir organizavimo įgūdžiai (užtikrinantys terminų laikymąsi visuose projektuose bei naudojami peržiūrint projekto išteklių poreikį ir įvertinant išlaidas)
		• bendravimo ir tarpasmeniniai įgūdžiai (perteikiant lūkesčius, derantis, sprendžiant konfliktus ir motyvuojant rezultatyvumą; šie įgūdžiai padeda laikytis kokybės etalonų)
		• technologiniai įgūdžiai (prižiūrint projekto tvarkaraščio vykdymą ir finansinius rezultatus)
		• laiko vadybos įgūdžiai (savo ir kitų)
		• problemų sprendimo įgūdžiai (sprendžiant laiko vadybos problemas bei kainų klausimus; šie įgūdžiai padeda užpildyti kokybės užtikrinimo spragas).
		• analizavimo, interpretavimo ir duomenų peržiūrėjimo įgūdžiai (vertinant pasiektus rezultatus pagal etalonus)

Žmogiškųjų išteklių vadybos metodų taikymas	*Žmogiškųjų išteklių planavimas ir vadyba*	*Žmogiškieji ištekliai: vadovavimas ir valdymas*
Reikalingos tokios bendros žinios ir supratimas:		
• žmogiškųjų išteklių ugdymo veikla: auklėjimas, praktinis mokymas ir pan. • poreikis žmogiškuosius išteklius valdyti visoje projekto vadybos sistemoje • žmogiškųjų išteklių vadybos vieta projekto gyvavimo ciklo ir kitų projektų vadybos funkcijų kontekste • žmogiškųjų išteklių vadybos įrankių ir metodų taikymas pareiškėjo kompetencijos srityje • kaip, kada ir kodėl diegiami žmogiškųjų išteklių vadybos procesai • asmens indėlio į žmogiškųjų išteklių vadybos procesą svarba • individų ir komandų indėlis, jų kuriama sinergija • konfliktų sprendimo principai ir veikla	• žmogiškųjų išteklų vadybos svarba projektų vadybos aplinkoje • žmogiškųjų išteklių ugdymo veikla: auklėjimas, praktinis mokymas ir pan. • įprasta organizacijos politika, standartai ir metodai, naudojami siekiant žmogiškųjų išteklių vadybos tikslų • pripažintų žmogiškųjų išteklių atrankos, paskyrimo, mokymo, rezultatų vertinimo ir motyvacijos įrankių naudojimas • konfliktai ir stresas, kurį patiria projektams vadovaujantys asmenys, ypač žmogiškųjų išteklių matricos vadybos aplinkoje • žmogiškųjų išteklių vadybos rezultatai, kritiniai sėkmės ir žlugimo kriterijai, žmogiškųjų išteklių veiklos rezultatų vertinimo matai • tarpasmeninių įgūdžių taikymas • tarpasmeninių stiprybių ir silpnybių įvertinimas • skirtumai tarp darbo turinio, procesų ir rizikos, kuri veikia žmogiškųjų išteklų vadybos reikalavimus įvairiuose projekto etapuose	• žmogiškųjų išteklių vadybos svarba projektų vadybos aplinkoje • žmogiškųjų išteklių vadybos rezultatams pasiekti reikalinga politika, standartai ir metodai • žmogiškųjų išteklių atrankos, paskyrimo, mokymo, rezultatų vertinimo ir motyvacijos įrankių naudojimas, jų galimybės, trūkumai, taikymas ir poveikis • konfliktų ir streso problemos (kurias patiria projektams vadovaujantys asmenys, ypač žmogiškųjų išteklių matricos vadybos aplinkoje) • žmogiškųjų išteklių vadybos rezultatai, kritiniai sėkmės ir žlugimo kriterijai, žmogiškųjų išteklių rezultatyvumo vertinimo matai • tarpasmeninių įgūdžių taikymas, tarpasmeninių silpnybių ir stiprybių įvertinimas • reikiamų strategijų taikymas (pvz., lyderystės, sprendimų priėmimo, grupių dinamikos, pokyčių vadybos, mokymosi ir lavinimo) • interesų grupių vadybos klausimai ir strategijos vyresniesiems vadovams • skirtumai tarp darbo turinio, procesų ir rizikos • skirtingų lyderystės stilių bei būdų poveikio išmanymas • organizacijos kultūros ir kultūros skirtumų poveikis lyderystės stiliui • gebėjimas taikyti komandų vystymosi ir dinamikos žinias • grupių elgsenos principai • profesionalus etinės lyderystės taikymas • profesionalus įmonės atsakomybės taikymas
Įgūdžiai:		
• komandinis darbas • geri bendravimo įgūdžiai • analitiniai įgūdžiai	• planavimo įgūdžiai (leidžiantys nustatyti, kokių įgūdžių reikia, ir darbuotojams paskirstyti projekto pareigas) • bendravimo ir lyderio įgūdžiai (padedantys motyvuoti darbuotojus, perteikti lūkesčius ir užtikrinti rezultatų pasiekimą) • tarpasmeniniai įgūdžiai (sprendžiant konfliktus) • praktinio mokymo ir auklėjimo įgūdžiai (padedantys didinti našumą) • analitiniai įgūdžiai (peržiūrint projektą ir nustatant tobulinimo būdus)	• planavimo įgūdžiai (leidžiantys žmogiškųjų išteklių reikalavimus nustatyti visai programai) • itin geri lyderystės įgūdžiai (įkvepiantys komandų, vadybininkų ir interesų grupių pasitikėjimą ir pasikliovimą) • bendravimo ir tarpasmeniniai įgūdžiai (perteikiant lūkesčius, derantis, sprendžiant konfliktus ir motyvuojant rezultatyvumą) • problemų sprendimo įgūdžiai (sprendžiant žmogiškųjų išteklių vadybos problemas visoje programoje) • treniravimo ir auklėjimo įgūdžiai • raštingumas (perduodant sprendimus ir rašant ataskaitas)



3 p a v. ***Sprendimams priimti reikalingos informacijos pateikimo formos***

kuriais remiantis buvo nustatytas konkretus kriterijų reikšmingumas bei reikšmės ir pan.

Tokiu būdu DLGCASP sistema sudaro sąlygas sprendimų priėmėjui gauti įvairiapusę išsamią kiekybinę ir kokybinę informaciją apie lyderio veiklos alternatyvas iš duomenų bazės bei remiantis modelių baze šiuos faktorius lanksčiai analizuoti ir priimti sprendimą.

Analizuojant duomenų bazės struktūrų tinkamumą sprendimų paramos sistemoms, pagal sprendžiamos problemos tipą galima išskirti skirtingą jų naudojimo efektyvumo lygį. Duomenų bazė turi tris fundamentines struktūras: hierarchinę, tinklinę ir reliacinę. DLGCASP sistemoje naudojama reliacinė duomenų bazės struktūra. Reliacinėje duomenų bazėje informacija saugoma lentelių pavidalu. DLGCASP sistemos duomenų bazę sudaro tokios lentelės:

- Pradinių duomenų lentelės. Jose pateikiama bendra informacija apie analizuojamą lyderio gyvavimo ciklą organizacijoje. Taip pat pateikiami projekto tikslai ir jų reikšmingumas, projektui skirta pinigų suma.
- Lyderio gyvavimo ciklo organizacijoje sprendimų įvertinimo lentelės. Šiose lentelėse pateikiama kiekybinė ir koncepcinė informacija apie alternatyvius sprendimų variantus. Toliau pateiktose 2 ir 3 lentelėse DLGCASP sistemai pateikti pradiniai duomenys, reikalingi daugiakriterinei analizei atlikti.
- Variantinio projektavimo lentelės. Šiose lentelėse pateikiama kiekybinė ir koncepcinė informacija apie lyderio gyvavimo ciklo organizacijoje elementų tarpusavio ryšius, suderinamumus ir galimas kombinacijas.

2.2. Modelių bazė

Kadangi lyderio gyvavimo ciklo alternatyvų efektyvumas dažnai vertinamas iš socialinių, kultūrinių, etinių, psichologinių, religinių, pasitikėjimo, nedarbo lygio, emocinių, vadybinių, organizacinių, techninių, teisinių, politinių, ekonominių ir kitokių pozicijų, todėl tarp sprendimų paramos sistemos modelių turi būti tokių, kurie padeda sprendimų priėmėjui atlikti šių variantų kompleksinę analizę ir priimti sprendimą. DLGCASP sistemoje šią funkciją atlieka tokie modelių bazę sudarantys modeliai:

- Alternatyvų sudarymo modelis;
- Pradinių kriterijų reikšmingumo nustatymo (taikant ekspertinių įvertinimų metodus) modelis;
- Kriterijų reikšmingumų nustatymo modelis;
- Variantinio projektavimo modelis;
- Daugiakriterinės analizės ir prioritetiškumo nustatymo modelis;
- Variantų naudingumo laipsnio nustatymo ir rinkos atlyginimo apskaičiavimo modelis;
- Rekomendacijų pateikimo modelis.

DLGCASP sistema, remdamasi šiais modeliais, automatizuotai sudaro alternatyvius lyderio gyvavimo ciklo variantus, atlieka daugiakriterinę jų analizę, nustato naudingumo laipsnį, apskaičiuoja rinkos atlyginimą ir išrenka efektyviausius variantus.

Naudojantis modelių bazės valdymo sistema pagal vartotojo poreikį taikomi įvairūs modeliai. Naudojant modelių bazės valdymo sistemą vienų modelių (pradinių kriterijų reikšmingumo nustatymo) skaičiavimo rezultatai tampa kitų modelių (variantinio projektavimo, daugiakri-

terinės analizės) pradiniais duomenimis, o šių modelių rezultatai – dar kitų modelių (variantų naudingumo laipsnio nustatymo, rekomendacijų pateikimo) pradiniais duomenimis.

3. Praktinis pavyzdys

Tyrimo metu apklausti 57 ekspertai (iš jų 27 Lietuvos ir 30 užsienio ekspertų). Apibendrinus ekspertų pateiktus atsakymus buvo nustatytas kiekvienos lyderio kompetencijos reikšmingumas ir prioritetiškumas, svertinis koeficientas, nurodantis kompetencijos svarbą analizuojant alternatyvas. Juo didesnis svertinis koeficientas, tuo didesnis kompetencijos svoris. Remiantis daugiakriterinės analizės metodu (šiuo metodu nagrinėjamų variantų prioritetiškumas ir reikšmingumas tiesiogiai ir proporcingai priklauso nuo alternatyvas adekvačiai apibūdinančių kriterijų sistemos, kriterijų reikšmių ir reikšmingumų dydžių. Kriterijų sistemą nustato, o kriterijų reikšmes ir pradinius reikšmingumus apskaičiuoja ekspertai. Visą šia informaciją gali keisti suinteresuotos grupės (užsakovas, vartotojas ir pan.), atsižvelgdamos į siekiamus savo tikslus) sudarant sprendimų priėmimo matricas buvo įvertinti asmenų alternatyvūs variantai. Iš 2–3 lentelėse pateiktų rezultatų matyti, kad siekiant įgyvendinti pagrindinius organizacijos tikslus pagal prioritetiškumą geriausios ir organizacijai svarbiausios yra lyderystės ir gebėjimo vadovauti, atvirumo naujoms idėjoms ir drąsos, bendradarbiavimo ir dalijimosi žiniomis, atsakomybės ir motyvacijos, kūrybiškumo, inovacijų taikymo ir plėtros, efektyvesnio inovacijos srautų valdymo, komandinio darbo, globalaus mąstymo, motyvavimo, bendravimo, tikslų įgyvendinimo, vertybinės orientacijos, verčių kūrimo bei konkurencingumo ir integruotų užduočių kompetencijos. Mažiausiai svarbios užduočių struktūrizavimo (0,001) ir darbuotojų kontrolės (0,001) kompetencijos. Sudarytų matricų 2–3 lentelėse rodiklis „+/-“ rodo palankesnę kompetencijos reikšmę, kai jis yra maksimizuojamas / minimizuojamas.

Sudarant kriterijų sistemą, buvo integruotos ir tyrimo metu gautos rekomendacijos / pasiūlymai, ir ekspertų komentarai, pavyzdžiui, prie kriterijaus „skaitmeninė kompetencija“ pateikiamas komentaras: „Svarbu yra holistinis ir kompleksinis požiūris, kuris neturėtų būti ribojamas tik skaitmeninio kompleksiškumo. Suvokimas, kad technologijos, aplinka, energijos resursai ir kt. dalykai susiję, ir tai neatsiejamai formuoja tiek gyvenimo stilių, tiek verslo modelius, tiek kitus dalykus“.

Viena didelė tarptautinė organizacija paskelbė konkursą vadovo pareigoms. Organizacija, atsižvelgdama į savo nuostatas, pagrindinius tikslus ir strategiją, numatė, kokių kompetencijų turinčių lyderių organizacijai reikės šiuo metu ir per artimiausius penkerius metus.

Du aukščiausios profesinės kompetencijos asmenys pateikė prašymus užimti šią vietą. Remiantis anksčiau pateiktu daugiakriterinės analizės metodu buvo įvertintos šių potencialių asmenų kompetencijos sudarant sprendimų priėmimo matricas. Potencialių kandidatų alternatyvos ir šių asmenų kompetencijos pateiktos 2 ir 3 lentelėse (bei kitose 8 lentelėse tinklapyje http://iti.vgtu.lt/imitacijosmain/simpletable.aspx?sistemid=518) ir kelios iš jų (lyderystės ir gebėjimo vadovauti, atvirumo naujoms idėjoms ir drąsos, bendradarbiavimo ir dalijimosi žiniomis, atsakomybės ir motyvacijos, kūrybiškumo, inovacijų taikymo ir plėtros, efektyvesnio inovacijos srautų valdymo, komandinio darbo, globalaus mąstymo, motyvavimo, bendravimo, tikslų įgyvendinimo, vertybinės orientacijos, verčių kūrimo bei konkurencingumo ir integruotų užduočių kompetencijos), t. y. svarbiausios organizacijai, kompetencijos toliau aprašytos išsamiau.

Pirmo analizuojamo **kriterijaus** „Penkios svarbiausios, lemiančios artimiausių 5-erių metų sėkmę, XXI a. žinių visuomenės lyderių kompetencijos“ (2 lentelė, X_1) patys svarbiausi pokriterijai (kompetencijos) yra Lyderystė ir gebėjimas vadovauti (X_{11}, q_1=0,18) bei Atvirumas naujoms idėjoms ir drąsa (X_{12}, q_2=0,17). Šie pokriterijai toliau trumpai aprašomi 1-ojo ir 2-ojo lyderio pavyzdžiu.

Vertinant pokriterijį (kompetenciją) **„Lyderystė ir gebėjimas vadovauti“ (X_{11}, q_1=0,18)** nustatyta, kad **1 lyderis** turi geras lyderystės ir gebėjimo vadovauti kompetencijas. Lyderis visada iniciatyvus ir pasirengęs siekti galimybių.

2 lentelė. ***Duomenys, reikalingi XXI a. žinių visuomenės lyderiui būtinų kompetencijų daugiakriterei analizei atlikti***

Su alternatyvomis susijusi kiekybinė ir kokybinė informacija					
Vadovo alternatyvas apibūdinančios kompetencijos	*	**Matavimo vienetai**	**Kriterijaus svoris (svertinis koeficientas)**	**Palygintos alternatyvos**	
				1 lyderis	**2 lyderis**
Lyderystė ir gebėjimas vadovauti	+	Balai	0,18	27	23
Atvirumas naujoms idėjoms ir drąsa	+	Balai	0,17	25	20
Gebėjimas daryti globalias ir dinamiškas įžvalgas	+	Balai	0,11	17	14
Savirefleksija ir motyvacija	+	Balai	0,1	13	16
Skaitmeninio kompleksiškumo suvokimas	+	Balai	0,1	12	16
Bendradarbiavimas ir organizacijos pozicionavimas	+	Balai	0,1	14	14
Turbulentiškų aplinkų valdymas	+	Balai	0,08	17	7
Žmogiškųjų išteklių valdymas	+	Balai	0,07	10	9
Skaitmeninė kompetencija	+	Balai	0,06	8	8
Rizikos kontrolė ir konkurencingumas	+	Balai	0,04	7	3
Skaitmeninė kompetencija	+	Balai	0,015	8	8
Rizikos valdymo kompetencijos	+	Balai	0,014	7	3
Socialinė kompetencija	+	Balai	0,019	5	9
Kokybės valdymo kompetencijos	+	Balai	0,017	9	7
Užduočių struktūrizavimas	-	Balai	0,001	4	9

* Simbolis „+(-)“ parodo, kad didesnė (mažesnė) kriterijaus reikšmė atitinka didesnį reikšmingumą vartotojui (suinteresuotoms grupėms)

Stipriai orientuotas į pasiekimus. Įkvepia darbuotojus siekti rezultatų, juos motyvuoja, perteikia lūkesčius ir užtikrina, kad rezultatai bus pasiekti. Numato, atpažįsta ir patenkina klientų poreikius. Užtikrina produkto pagaminimo kokybę, kad jis atitiktų reikalavimus, kiekį, suderintą kainą ir aptartą laiką. Asmuo laiką planuoja efektyviai, užtikrina produktyvumą, tikslai, uždaviniai ir rezultatai pasiekiami be įtampos ir nuovargio. Turi kelerių metų vadovavimo patirtį. Puikios finansinės ataskaitos ir kontrolė, gerai vykdomi verslo ir finansiniai planai, gera įmonės struktūra.

O štai **2 lyderis** sunkiau vadovauja asmenims ir grupėms, silpnesnė įtikinėjimo ir darbuotojų motyvavimo taktika. Kartais trūksta iniciatyvos, mažiau orientuotas į pasiekimus, užduočiai atlikti išnaudojama daugiau energijos. Moka vadovauti, tinkamos finansinės ataskaitos ir kontrolė. Paprastai laikomasi verslo ir finansinio plano. Kartais atsiranda nenumatytas lėšų poreikis. Nepastebėta organizacinių nesklandumų.

Vertinant **1 lyderio Atvirumo naujoms idėjoms ir drąsos (X_{12}, q_2=0,17)** kompetenciją nustatyta, kad nors komandos narių, kurių nelaiko autoritetais, idėjoms 1 lyderis nėra itin atviras, tačiau pats ieško iššūkių, lengvai perpranta ir aktyviai diegia naujoves, drąsiai, bet valdydamas riziką priima sprendimus. **2 lyderio** ši kompetencija silpnesnė, t. y. šis lyderis neieško iššūkių, jo vadovavimas labiau inertiškas ir konservatyvus, tačiau yra atviras kitų keliamoms naujoms idėjoms.

Antro analizuojamo **kriterijaus** „Svarbiausios organizacijos žinių darbuotojų kompetencijos siekiant pagrindinių organizacijos tikslų“ (3 lentelė, X_2) patys svarbiausi pokriterijai (kompetencijos) yra Bendradarbiavimas (X_{21}, q_1=0,12), Atsakomybė ir motyvacija (X_{22}, q_2=0,1) bei Kūrybiškumas (X_{23}, q_3=0,1).

Vertinant **Bendradarbiavimo (X_{21}, q_1=0,12)** kompetenciją nustatyta, kad 2 lyderis bendradarbiauti geba geriau nei 1 lyderis. **1 lyderis** suvokia bendradarbiavimo svarbą ir paprastai geba bendradarbiauti, tačiau kartais priima vienašališkus sprendimus. **2 lyderis** turi gerą bendradarbiavimo kompetenciją, leidžia darbuotojams realizuoti savo potencialą, t. y. dalyvauti priimant sprendimus, pavestą darbą atlieka kokybiškai, siekia nuolat tobulėti.

3 l e n t e l ė. ***Duomenys, reikalingi svarbiausių organizacijos žinių darbuotojų kompetencijų siekiant pagrindinių organizacijos žinių daugiakriterinei analizei atlikti***

Su alternatyvomis susijusi kiekybinė ir kokybinė informacija					
Vadovo alternatyvas apibūdinančios kompetencijos	*****	**Matavimo vienetai**	**Kriterijaus svoris (svertinis koeficientas)**	**Palygintos alternatyvos**	
				1 lyderis	**2 lyderis**
Bendradarbiavimas	+	Balai	0,12	16	17
Atsakomybė ir motyvacija	+	Balai	0,1	15	14
Kūrybiškumas	+	Balai	0,1	14	14
Analitinis, globalus mąstymas	+	Balai	0,09	15	11
Gebėjimas atlikti kompleksines užduotis	+	Balai	0,09	10	15
Gebėjimas greitai mokytis	+	Balai	0,09	14	11
Skaitmeninis raštingumas	+	Balai	0,06	14	14
Gebėjimas spręsti problemas	+	Balai	0,06	7	3
Lankstumas	+	Balai	0,05	3	11
Informacijos valdymas: atranka ir analizė	+	Balai	0,05	8	6
Efektyvumas, rezultatų pasiekimas	+	Balai	0,05	10	6
Gebėjimas kurti ir dalytis turiniu	+	Balai	0,04	4	8
Tikslų pasiekimas	+	Balai	0,04	10	5
Gebėjimas prisiimti riziką	+	Balai	0,04	6	4
Komunikabilumas	+	Balai	0,03	6	6
Individualus darbas	-	Balai	0,01	2	1

Analizuojant **Atsakomybės ir motyvacijos (X_{22}, q_2=0,1)** kompetenciją nustatyta, kad **1 lyderis** yra atsakingas. Jis pasitiki savimi, įvertina save, savo stiprybes ir galimybes. Visada yra motyvuotas ir prisiima atsakomybę už kokybišką ir efektyvų tikslų, uždavinių įgyvendinimą, rezultatų pasiekimą. Numato, atpažįsta ir patenkina klientų poreikius. Užtikrina produkto pagaminimo kokybę, kad jis atitiktų reikalavimus, kiekį, suderintą kainą ir aptartą laiką. **2 lyderis** yra motyvuotas, tačiau jam kartais trūksta atsakomybės įgyvendinti užsibrėžtus tikslus ir uždavinius, pasiekti planuotus rezultatus: jis identifikuoja klientų poreikius, tačiau kartais neužtikrina tų poreikių patenkinimo, t. y. produkto pagaminimo kokybės, kad jis atitiktų reikalavimus, kiekį, suderėtą kainą ir aptartą laiką.

Kūrybiškumas (X_{23}, q_3=0,1) – viena iš vadybos kompetencijų, padedančių spręsti sudėtingus aplinkos galvosūkius. **Abu lyderiai (1 lyderis ir 2 lyderis)** yra kūrybiški vadovaudami organizacijai: priima ir diegia naujoves, kuria praktinę patirtį, darbe taiko „sumanias“ technologijas.

Galima pabrėžti, kad skirtingoms suinteresuotoms grupėms atliekant to paties lyderio gyvavimo ciklo organizacijoje variantinį projektavimą ir daugiakriterinę analizę dažnai gaunami skirtingi rezultatai. Kadangi skirtingų suinteresuotų grupių siekiamų tikslų visuma ne visai sutampa, todėl savaime suprantama, kad jų pateikti pradiniai reikalavimai ir galutiniai rezultatai skirsis. Suinteresuotų grupių siekiami tikslai ir esamos finansinės galimybės išreiškiami kiekybine forma ir pateikiami kaip pradiniai skaičiavimo duomenys. Šie pradiniai duomenys turi būti susieti su kitomis lentelėmis.

Lyderio gyvavimo ciklo organizacijoje sprendimų įvertinimo lentelėse pateikiami galimi variantai, variantų kiekybinis ir koncepcinis aprašymas.

Norint suprojektuoti ir įgyvendinti efektyvų lyderio gyvavimo ciklo organizacijoje variantą, būtina išnagrinėti galimus alternatyvius sprendimus. Siekiant atlikti alternatyvų automatizuotą variantinį projektavimą, būtina sudaryti lyderio gyvavimo ciklo organizacijoje elementų ir sprendimų tarpusavio ryšio, suderinamumo, kombinacijų ir variantinio projektavimo lenteles. DLGCASP sistema, remdamasi pradiniais duomenimis, pagal pasiūlytą variantinio projektavimo

4 l e n t e l ė. ***XXI a. žinių visuomenės lyderiui būtinų kompetencijų daugiakriterio vertinimo rezultatai***

Su alternatyvomis susijusi kiekybinė ir kokybinė informacija					
Vadovo alternatyvas apibūdinančios kompetencijos	*****	**Matavimo vienetai**	**Kriterijaus svoris (svertinis koeficientas)**	**Palygintos alternatyvos**	
				1 lyderis	**2 lyderis**
Lyderystė ir gebėjimas vadovauti	+	Balai	0,18	0,0972 AVG MIN	0,0828 AVG MIN
Atvirumas naujoms idėjoms ir drąsa	+	Balai	0,17	0,0944 AVG MIN	0,0756 AVG MIN
Gebėjimas daryti globalias ir dinamiškas įžvalgas	+	Balai	0,11	0,0603 AVG MIN	0,0497 AVG MIN
Savirefleksija ir motyvacija	+	Balai	0,1	0,0448 AVG MIN	0,0552 AVG MIN
Skaitmeninio kompleksiškumo suvokimas	+	Balai	0,1	0,0429 AVG MIN	0,0571 AVG MIN
Bendradarbiavimas ir organizacijos pozicionavimas	+	Balai	0,1	0,05 AVG MIN	0,05 AVG MIN
Turbulentiškų aplinkų valdymas	+	Balai	0,08	0,0567 AVG MIN	0,0233 AVG MIN
Žmogiškųjų išteklių valdymas	+	Balai	0,07	0,0368 AVG MIN	0,.0332 AVG MIN
Skaitmeninė kompetencija	+	Balai	0,06	0,03 AVG MIN	0,03 AVG MIN
Rizikos kontrolė ir konkurencingumas	+	Balai	0,04	0,028 AVG MIN	0,012 AVG MIN
Skaitmeninė kompetencija	+	Balai	0,015	0,0075 AVG MIN	0,0075 AVG MIN
Rizikos valdymo kompetencijos	+	Balai	0,014	0,0098 AVG MIN	0,0042 AVG MIN
Socialinė kompetencija	+	Balai	0,019	0,0068 AVG MIN	0,0122 AVG MIN
Kokybės valdymo kompetencijos	+	Balai	0,017	0,0096 AVG MIN	0,0074 AVG MIN
Užduočių struktūrizavimas	-	Balai	0,001	0,0003 AVG MIN	0,0007 AVG MIN
Suma svertinių koeficientų, normalizuotų, maksimizuotų (projekto pliusų), susijusių su alternatyvomis				0,5748	0,5002
Suma svertinių koeficientų, normalizuotų, minimizuotų (projekto minusų), susijusių su alternatyvomis				0,0003	0,0007
Alternatyvos reikšmingumas (Q_j)				0,5755	0,5005
Alternatyvos prioritetiškumas				**1**	2
Alternatyvos naudingumo laipsnis (N_j), (%)				100%	86,97%

metodą gali sudaryti daugelį alternatyvių variantų. Nustatoma, ar parengti variantai atitinka jiems keliamus reikalavimus. Neatitinkantis šių reikalavimų variantas toliau nenagrinėjamas. Atliekant lyderio gyvavimo ciklo organizacijoje variantinį projektavimą iškyla kriterijų reikšmingumo suderinamumo problema. Šiuo atveju kompleksiškai vertinant alternatyvas konkretaus kriterijaus reikšmingumo dydis priklauso nuo visumos vertinamų kriterijų, jų reikšmių ir pradinių reikšmingumų.

Remiantis 2 ir 3 lentelėse (bei kitose aštuoniose lentelėse tinklapyje http://iti.vgtu.lt/imitacijosmain/simpletable.aspx?sistemid=518) pateiktais duomenimis, su DLGCASP sistema buvo atlikta dviejų alternatyvių vadovų daugiakriterinė analizė. DLGCASP sistema nustatė alternatyvų prioritetiškumą. Juo didesnis Q_j, tuo didesnis vadovo efektyvumas (prioritetiškumas). Suformuotas apibendrintas (redukuotas) kriterijus Q_j tiesiogiai ir proporcingai priklauso nuo lyginamų kriterijų reikšmių x_{ij} ir

reikšmingumų q_i santykinės įtakos galutiniam rezultatui.

Vadovo a_j reikšmingumas Q_j rodo organizacijos tikslų pasiekimo lygį. Efektyviausio vadovo reikšmingumas Q_{max} visada bus didžiausias. Visų kitų likusių vadovų reikšmingumas yra mažesnis už Q_{max}, t. y. organizacijos suminiai tikslai tenkinami blogiau negu geriausio vadovo atveju.

Tačiau praktiškai organizaciją labiau domina ne nagrinėjamų vadovų reikšmingumas ir prioritetiškumas, bet tai, koks yra vadovų naudingumo laipsnis ir kokia iš to išplaukianti jų rinkos vertė. Kitaip sakant, organizaciją labiausiai domina vadovas, kuris geriausiai patenkins jos tikslus, mažiausiai kainuos ir mažiausiai pažeis kitų suinteresuotų grupių poreikius ir tikslus. Šio įvertinimo rezultatams išreikšti vartojamos vadovo naudingumo laipsnio ir rinkos vertės sąvokos.

Siekiant tiksliau įvertinti nagrinėjamų vadovų gerąsias ir blogąsias savybes, jas palyginti su organizacijos tikslais ir visa tai suderinti su vadovo atlyginimu, būtina nustatyti vadovų naudingumo laipsnius ir vertinamo vadovo rinkos atlyginimą.

Toliau kaip pavyzdys pateikti XXI a. žinių visuomenės lyderiui būtinų kompetencijų daugiakriterinio vertinimo rezultatai (4 lentelė).

Remiantis pirmiau pateiktais lyderių daugiakriterinės analizės rezultatais, alternatyvų reikšmingumu bei jų prioritetiškumu matyti, kad pirmasis lyderis yra tinkamesnis paskelbtai didelės tarptautinės organizacijos vadovo vietai.

Išvados

Remiantis modelių ir sistemų analize, buvo sukurti Lyderio gyvavimo organizacijoje proceso kompleksinės analizės modelis ir sistema. Originali Daugiakriterinė lyderio gyvavimo organizacijoje ciklo analizės sprendimų paramos sistema sudaro sąlygas kompleksiškai analizuoti lyderystės gyvavimo procesą, jo sudedamąsias dalis, jame dalyvaujančias suinteresuotas grupes ir jį veikiančią išorinę mikro-, mezo- ir makrolygmens aplinką. Sudarytos lyderio gyvavimo proceso ir jo sudedamųjų dalių kompleksinės duomenų bazės, kuriose alternatyvos kompleksiškai aprašomos vadybiniais, organizaciniais, ekonominiais, teisiniais, emociniais, psichologiniais ir kitais aspektais. Remiantis šiomis duomenų bazėmis, sukurtos sistemos sudaro sąlygas kompleksiškai analizuoti variantus kiekybine ir koncepcine formomis.

LITERATŪRA

APGAR, D. (2007). *Rizikos intelektas. Kaip išmokti valdyti tai, ko nežinome*. Vilnius. 211 p. ISBN 978-9955-460-48-0.

ARMSTRONG, M. (2001). *A hadbook of Human Resource Management Practice*. 8th edition. Kogan Page, US. 981 p. ISBN 0 7494 3393 0.

ASHTIANI, Milad Gorji; MAKUI, Ahmad; RAMEZANIAN, Reza (2013). A robust model for a leader–follower competitive facility location problem in a discrete space. *Applied Mathematical Modelling*, vol. 37, iss. 1–2, p. 62–71.

BARKAUSKAITĖ, M; GUDŽINSKIENĖ, V. (2007). *Inovatyvinių šiuolaikinių studijų technologijų kriterijai ir aprašas*: mokymo metodinė medžiaga. Vilnius, 160 p. ISBN 978-9955-668-90-9.

CASTELLS, M. (2007). *Tinklaveikos visuomenės raida. Informacijos amžius. Ekonomika, visuomenė ir kultūra.* III tomas: *Tūkstantmečio pabaiga*. Kaunas. 416 p. ISBN 978-9986-850-58-8.

CHEN Mingli; GOU, Xueqiang; DU, Yaping (2013).The effect of ground altitude on lightning striking distance based on a bi-directional leader model. *Atmospheric Research*, vol. 125–126, p. 76–83.

CHO, Jeewon; PARK, Insu; MICHEL, John W. (2011). How does leadership affect information systems success? The role of transformational leadership. *Information & Management*, vol. 48, iss. 7, p. 270–277.

CHUN, Young Woo; TAK, Kwang-Il (2009). Songgye, a traditional knowledge system for sustainable forest management in Choson Dynasty of Korea.

Forest Ecology and Management, vol. 257, iss. 10, p. 2022–2026.

DAY, David V.; SIN, Hock-Peng (2011). Longitudinal tests of an integrative model of leader development: Charting and understanding developmental trajectories. *The Leadership Quarterly*, vol. 22, iss. 3, p. 545–560.

DIMITRIOS, Nasiopoulos K.; SAKAS, Damianos P.; VLACHOS, D.S. (2013). The Role of Information Systems in Creating Strategic Leadership Model. *Procedia – Social and Behavioral Sciences*, vol. 73, p. 285–293.

FERRARI, A. (2012). *Digital Competence in Practice: An Analysis of Frameworks*. European Commision. Joint Research Center. Institute for Prospective Technological Studies. Luxembourg. ISBN 978-92-79-25093-4 (pdf).

GIDDENS, A. (2002). *Modernybė ir asmens tapatumas. Asmuo ir visuomenė vėlyvosios modernybės amžiuje.* Vilnius. 315 p. ISBN 9986-943-60-4.

GUDAUSKAS, R.; KAUŠYLIENĖ, A. (2010). *Žinių vadyba ir edukacinės inovacijos*. Vilnius. 48 p. ISBN 978-9955-20-569-2.

GURRIA, A. (2011). *OECD PIAAC (Programme for the International Assessment of Adult Competencies)* [žiūrėta 2011 m. rugsėjo 23 d.]. Secretary-General. Prieiga internetu: <http://www.oecd.org/>.

HARMAN, Chris; BRELADE, Sue (2000). *Knowledge management and the role of HR. Securing competetitive advantage in the knowledge economy*. 106 p. ISBN 0 273 644 564.

KIM, Sungwook (2012). Multi-leader multi-follower Stackelberg model for cognitive radio spectrum sharing scheme. *Computer Networks*, vol. 56, iss. 17, p. 3682–3692.

LEHNER, Franz (1992). Expert systems for organizational and managerial tasks. *Information & Management*, vol. 23, iss. 1, p. 31–41.

LIETUVA 2030. *Lietuvos pažangos strategijos projektas „Lietuva 2030“* [žiūrėta 2011 m. gruodžio 15 d.]. Prieiga internetu: <www.lietuva2030.lt >.

LIM, Lai-Huat; RAMAN, K. S.; WEI, Kwok-Kee (1994). Interacting effects of GDSS and leadership. *Decision Support Systems*, vol. 12, iss. 3, p. 199–211.

LISABONOS STRATEGIJA (2000).

LIU, Shi-Cai; TAN, Da-Long; LIU, Guang-Jun (2007). Robust Leader-follower Formation Control of Mobile Robots Based on a Second Order Kinematics Model. *Acta Automatica Sinica*, vol. 33, iss. 9, p. 947–955.

LOVE, P.; FONG, P. S. W.; IRANI, Z. eds. (2005). *Management of Knowledge in Project Environments*. Elsevier Limited. 242 p. ISBN 0 7506 6251 4.

McKENNA, Bernard; ROONEY, David; BOAL, Kimberley B. (2009). Wisdom principles as a meta-theoretical basis for evaluating leadership. *The Leadership Quarterly*, vol. 20, iss. 2, p. 177–190.

NORKUS, Zenonas (2003). *Max Weber ir racionalus pasirinkimas*. Vilnius. 512 p. ISBN 9986-09-255-8.

OLIVEIRA, Marco A. de; POSSAMAI, Osmar; DALLA VALENTINA, Luiz V. O.; FLESCH, Carlos A. (2012). Applying Bayesian networks to performance forecast of innovation projects: A case study of transformational leadership influence in organizations oriented by projects. *Expert Systems with Applications*, vol. 39, iss. 5, p. 5061–5070.

PROBST, G; RAUB, S; ROMHARDT, K. (2006). *Žinių vadyba. Sėkmės komponentai*. Vilnius. 352 p. ISBN 9955-443-26-X.

RYCHEN, D. S.; SALGANIK, L. (2002). *DeSeCo Discussion Paper.* Definition and Selection of Competencies.

RAO, Ming; DONG, Randy; MAHALEC, Valdimir (1994). Intelligent system for safe process startup. *Engineering Applications of Artificial Intelligence*, vol. 7, iss. 4, p. 349–360.

REES, Jackie; KOEHLER, Gary J. (2000). Leadership and group search in group decision support systems. *Decision Support Systems*, vol. 30, iss. 1, p. 73–82.

SEAH, Melody; HSIEH, Ming Huei; WENG, Pu-Dong (2010). A case analysis of Savecom: The role of indigenous leadership in implementing a business intelligence system. *International Journal of Information Management*, vol. 30, iss. 4, p. 368–373.

SENSE, Andrew J. (2003). A model of the politics of project leader learning. *International Journal of Project Management*, vol. 21, iss. 2, p. 107–114.

SIX, Frédérique E.; BAKKER, Frank G.A. de; HUBERTS, Leo W.J.C. (2007). Judging a Corporate Leader's Integrity: An Illustrated Three-Component Model. *European Management Journal*, vol. 25, iss. 3, p. 185–194.

SLATER, D; TONKISS, F. (2004). *Rinkos visuomenė: sociologinė studija*. Vilnius. 341 p. ISBN 9986-39-365-5.

TAVAKOLI, M. R. Bank; VAHIDI, B. (2010). Shielding failure rate calculation by means of downward and upward lightning leader movement models: Effect of environmental conditions. *Journal of Electrostatics*, vol. 68, iss. 3, p. 275–283.

VARGAS, Mauricio; TORRES, Horacio (2008). On the development of a lightning leader model for tortuous or branched channels – Part I: Model description. *Journal of Electrostatics*, vol. 66, iss. 9–10, p. 482–488.

2009/C 184 E/06. *Europos Parlamento rezoliucija dėl 2008 m. pavasarį vyksiančiam Europos Vadovų Tarybos susitikimui perduodamos nuomonės apie Lisabonos strategiją*.

„Dėl valstybės ilgalaikės raidos strategijos. LRS nutarimas. 2002-11-12, Nr. IX-1187. Žin., 2002, Nr. 113-5029.

Komisijos komunikatas 2020 m. Europa. Pažangaus, tvaraus ir integracinio augimo strategija KOM (2010) 2020 galutinis.“

THE COMPLEX ANALYSIS MODEL AND THE SYSTEM OF A LEADER'S LIFE CYCLE PROCESS IN AN ORGANIZATION

Saulė Jokūbauskienė

S u m m a r y

The objective of the article is a Leader‘s life cycle process in an organization and the parties involved and concerned to realize their aims, as well as outside micro-, mezzo- and macro- level environment as a whole. According to the objective of the research a proper and complex analysis has been made with a view to adapt the methods of multicriteria analysis. As a result of the above mentioned methods used the objective was evaluated according to the leader‘s abilities in managerial, organizational, economic, social, legal, emotional, psychological and other aspects. The article presents the model of the complex analysis, the system and its application in practice for the Leader‘s life cycle in an organization.

INFORMACINĖS TECHNOLOGIJOS IR KALBA

Tekstinių dokumentų panašumų paieška naudojant saviorganizuojančius neuroninius tinklus ir *k* vidurkių metodą

Pavel Stefanovič

Vilniaus universiteto Matematikos ir informatikos instituto doktorantas
Vilnius University, Institute of Mathematics and Informatics, Doctoral student
Akademijos g. 4, LT-08663 Vilnius
El. paštas: pavel.stefanovic@mii.vu.lt

Olga Kurasova

Vilniaus universiteto Matematikos ir informatikos instituto vyresn. mokslo darbuotoja
Vilnius University, Institute of Mathematics and Informatics, Senior researcher
Akademijos g. 4, LT-08663 Vilnius
El. paštas: olga.kurasova@mii.vu.lt

Straipsnyje nagrinėjama dokumentų panašumų paieška naudojant du populiarius metodus: saviorganizuojančius neuroninius tinklus (SOM) ir k vidurkių metodą. Vienas iš šių metodų tikslų – suskirstyti duomenis į klasterius pagal jų panašumą. Analizuota tekstinių dokumentų matricos sudarymo faktorių įtaka gautiems rezultatams. SOM kokybei įvertinti pasiūlyti du nauji matai, skirti klasifikuotiems duomenims, kurių reikšmės parodo susidariusių klasterių išsidėstymą SOM žemėlapyje. Pirmasis matas parodo, kaip gerai tos pačios klasės duomenys išsidėsto žemėlapyje vienas šalia kito, antrasis matas – kaip toli yra skirtingų klasių centrai. K vidurkių metodu gautų rezultatų kokybei įvertinti skaičiuota suma nuo klasterio centro iki klasterio narių bei įvertintas klasių nesutapimas su klasteriais. Eksperimentiniams tyrimams atlikti pasirinkti tekstiniai dokumentai, paimti iš Lietuvos Respublikos Seimo dokumentų bazės.

Įvadas

Kelis dešimtmečius saviorganizuojantis neuroninis tinklas (SOM) buvo naudojamas įvairiems skaitiniams duomenims klasterizuoti, klasifikuoti ir vizualizuoti. Tačiau gausėjant įvairialypei informacijai, saviorganizuojantys neuroniniai tinklai pradėti taikyti tekstinei (Kohonen, Xing, 2011), grafinei (Sjoberg, Laaksonen, 2011), vaizdo (Maia, Barreto, Coelho, 2011), garso (Mayer, 2011) ir kitokio pobūdžio informacijai analizuoti.

Šis straipsnis skiriamas tekstinei informacijai nagrinėti. Internetinėje erdvėje gausu įvairaus pobūdžio tekstinės informacijos: dokumentų, tekstų svetainėse, mokslinių straipsnių, todėl natūralu, jog atsiranda poreikis šią informaciją apdoroti siekiant gauti naudingų žinių. Gausybė informacijos internete mums leidžia rasti vieną ar kitą dalyką labai greitai, tačiau ji dažnai būna nenaudinga, iškraipyta ar neesminė. Todėl vis dažniau atsiranda įvairiausių duomenų tyrybos metodų jai analizuoti ir sisteminti (Marinai, 2011; Alsmadi, Saleh, 2012; Sihag, Kumar, 2013).

Saviorganizuojantys neuroniniai tinklai yra sėkmingai taikomi įvairiems uždaviniams spręsti: dokumentų plagijavimui tikrinti (Chow, Rahman, 2009), internetinėse svetainėse pateiktiems straipsniams analizuoti (Mayer, Rauber, 2011) ir kt. Kitas būdas duomenims klaste-

rizuoti ir klasifikuoti yra *k* vidurkių metodas (MacQueen, 1967). Šis metodas taip pat sėkmingai naudojamas tekstinių duomenų analizei (Steinbach, Karypis, Kumar, 2000).

Darbe (Sihag, Kumar, 2013) analizuojamas *k* vidurkių metodas tekstiniams dokumentams klasterizuoti. Pasiūlytas grafais grįstas būdas pradiniams klasterių centrams skaičiuoti. Palyginti rezultatai, kai pradiniai centrai inicijuojami kaip įprasta ir pasiūlytuoju būdu. Rezultatai parodė naujojo būdo pranašumą. Hierarchinio klasterizavimo ir *k* vidurkių metodų, taikomų tekstiniams dokumentams klasterizuoti, lyginamoji analizė atlikta darbe (Steinbach, Karypis, Kumar, 2000). Rezultatai parodė, kad *k* vidurkių metodas yra pranašesnis už hierarchinį klasterizavimą. Saviorganizuojančiu neuroniniu tinklu grįstas metodas, taikytas tekstams iš senų šaltinių atkurti, tirtas darbe (Marinai, 2011). Atlikta tekstinių dokumentų panašumo analizė, siekiant nustatyti mokslinių straipsnių plagiatus (Alsmadi, Saleh, 2012). Dokumentų panašumui nustatyti naudoti kosinusų, Euklido, Dice ir miesto kvartalų atstumų matai.

Norint duomenis analizuoti saviorganizuojančiu neuroniniu tinklu, *k* vidurkių ar kitu klasterizavimo metodu, jie turi būti pateikiami skaitine išraiška, todėl iškyla problema, kaip tinkamai paversti tekstinę informaciją į skaitinę. Tinkamai parinkus faktorius, keičiančius duomenis iš įvairių tekstinių duomenų formatų į skaitinį, galima tiksliau vertinti ir analizuoti gautus rezultatus. Visuose minėtuose straipsniuose šių faktorių analizė neatlikta. Šiame straipsnyje tiriama minėtų faktorių įtaka rezultatams, kurie gauti saviorganizuojančiu neuroniniu tinklu ir *k* vidurkių metodu, siekiant nustatyti tekstinių dokumentų – kelių Lietuvos ministerijų įsakymų panašumus.

1. Saviorganizuojantys neuroniniai tinklai

Pagrindinis saviorganizuojančio neuroninio tinklo tikslas – išlaikyti duomenų topologiją (Dzemyda, Kurasova, Žilinskas, 2008). SOM tinklai gali būti naudojami duomenims klasterizuoti ir vizualizuoti, taip pat ieškant daugiamačių duomenų projekcijų į mažesnio skaičiaus matmenų erdvę. Saviorganizuojantis neuroninis tinklas yra neuronų, išdėstytų dvimačio tinklelio (žemėlapio, lentelės) mazguose, masyvas $M = \{M_{kl}, k = 1,...,r, l = 1,...,s\}$ (Kohonen, 2001); čia r – tinklo eilučių skaičius, s – tinklo stulpelių skaičius, M_{kl} – vektorius, kurio matmenų skaičius n toks pat kaip mokymo duomenų vektorių. SOM tinklas mokomas mokymo be mokytojo būdu (angl. *unsupervised*). Neuroniniam tinklui daug kartų pateikiama skirtingų objektų, nusakomų n-mačiais vektoriais $X_1, X_2, ..., X_N$. Kiekviename mokymo žingsnyje (iteracijoje) vienas mokymo aibės vektorius $X_p \in \{X_1, X_2,..., X_N\}$ pateikiamas į tinklą. Mokymo pradžioje neuronų (vektorių) M_{kl} komponenčių pradinės reikšmės dažniausiai nustatomos atsitiktinai (Stefanovič, Kurasova, 2009). Vektorius X_p palyginamas su visais neuronais M_{kl}. Dažniausiai skaičiuojamas Euklido atstumas $\| X_p - M_{kl} \|$ tarp šio vektoriaus X_p ir kiekvieno neurono M_{kl}. Randama, iki kurio neurono $M_{k_w l_w} \in \{M_{kl}, k = 1,...,r, l = 1,...,s\}$ atstumas yra mažiausias; rastas neuronas $M_{k_w l_w}$ vadinamas neuronu (vektoriumi) nugalėtoju (angl. *winner*). Visų neuronų komponentės keičiamos naudojantis iteracine formule (1).

$$M_{kl}(t+1) = M_{kl}(t) + \\ + h(k,l,k_w,l_w,t)\left(X_p - M_{kl}(t)\right). \quad (1)$$

Čia t yra iteracijos numeris, $h(k, l, k_w, l_w, t)$ – kaimynystės funkcija. Remiantis ankstesniais tyrimais, eksperimentiniuose tyrimuose naudota Gausso kaimynystės funkcija (Stefanovič, Kurasova, 2011):

$$h(k,l,k_w,l_w,t) = \\ = \alpha(t)\exp\left(-\left\| R_{k_w l_w} - R_{kl} \right\|^2 / 2(\eta(k,l,k_w,l_w,t))^2\right).$$

Čia $\alpha(t)$ – mokymo greitis, $\eta(k, l, k_w, l_w, t)$ yra neuronų M_{kl} ir $M_{k_w l_w}$ kaimynystės eilė, dvimačiai vektoriai R_{kl} and $R_{k_w l_w}$ sudaryti iš neuronų M_{kl} ir $M_{k_w l_w}$ indeksų, parodančių neuronų vietą SOM tinkle. Eksperimentiniuose tyrimuose naudotas mokymo greitis $\alpha(t) = (1 - t/T)$; čia T – iteracijų skaičius.

SOM kokybei nustatyti siūlomi du nauji matai, kurie naudojami tik klasifikuotiems duomenims analizuoti. Pirmasis matas (2) parodo, kaip arti žemėlapyje tos pačios klasės neuronai išsidėsto šalia vienas kito. Kuo mato reikšmė yra mažesnė, tuo rezultatas geresnis.

$$E_c = \frac{1}{N_c} \sum_{i=1}^{n_c - 1} \sum_{j=1}^{n_c} \left(\| Z_i^c - Z_j^c \| k_i^c k_j^c + b \right). \quad (2)$$

Čia c – klasės numeris, N_c – c-osios klasės vektorių skaičius, n_c – žemėlapio neuronų, į kuriuos pateko c-osios klasės vektoriai, skaičius, Z_i^c – žemėlapio langelių indeksai, į kuriuos pateko c-osios klasės neuronai, k_i^c – c-osios klasės vektorių, patekusių į Z_i^c langelį, skaičius. Gali būti atvejų, kai į tuos pačius žemėlapio langelius patenka skirtingų klasių nariai, tuomet pagal (3) formulę skaičiuojama bauda b. Jei viename langelyje yra tik tos pačios klasės vektoriai, tai $b = 0$.

$$b = \frac{l_i^{c'}}{k_i} + \frac{l_j^{c'}}{k_j}. \quad (3)$$

Čia k_i (k_j) – Z_i^c (Z_j^c) langelyje esančių vektorių skaičius, $l_i^{c'}$ ($l_j^{c'}$) – kitų nei c-osios klasių vektorių, esančių Z_i^c (Z_j^c) langelyje, skaičius.

Antrasis pasiūlytas matas (5) įvertina, kaip toli yra išsidėstę skirtingų klasių centrai. Šiuo atveju kuo mato reikšmė yra didesnė, tuo rezultatas geresnis. Iš pradžių randami visų žemėlapyje pavaizduotų klasių centrų indeksai Y^c (4).

$$Y^c = \frac{1}{n_c} \sum_{i=1}^{n_c} Z_i^c. \quad (4)$$

Tuomet antrojo mato reikšmė skaičiuojama pagal formulę:

$$E_{center} = \frac{1}{n} \sum_{c=1}^{n-1} \sum_{d=c+1}^{n} \left\| Y^c - Y^d \right\|. \quad (5)$$

Čia n – klasių skaičius.

2. *K* vidurkių metodas

Naudojant k vidurkių metodą stengiamasi duomenų aibę padalinti į nesusikertančius klasterius (Molytė, 2011). Šiame klasterizavimo algoritme minimizuojama tam tikra kriterijaus funkcija. Ji turi būti tokia, kad minimizuojant klasterių panašumą, klasterių skirtumas būtų maksimizuojamas. Tai gali būti vidutinis atstumas tarp klasterių. K vidurkių metodo etapai yra šie: 1) atsitiktinai iniciuojami klasterių centrai; 2) kiekvienas analizuojamos duomenų aibės vektorius yra priskiriamas tam klasteriui, iki kurio atstumas nuo centro yra mažiausias; 3) perskaičiuojami kiekvieno klasterio centrai; 4) skaičiuojama kvadratinė paklaida tarp klasterio centro ir klasteriui priskirtų duomenų; 5) 2–4 etapai kartojami tol, kol analizuojami duomenys nebepasiskirsto kitiems klasteriams.

Įprastai k vidurkių metodo kokybei įvertinti skaičiuojama gautų klasterių atstumų nuo klasterio centro iki jam priskirtų duomenų suma. Šiame straipsnyje analizuoti duomenys, priskirti kelioms klasėms, todėl svarbu įvertinti, ar gauti klasteriai atitinka klases. Pradžioje duomenys klasterizuojami į tiek klasterių, kiek yra klasių. Nustatomas klasių ir klasterių atitikimas, t. y. tariama, kad tam tikros klasės duomenys turi būti priskirti klasteriui, į kurį pateko dauguma tos klasės narių. Tuomet suskaičiuojama, kiek kiekvienos klasės narių pateko ne į savo klasterį. Šio įverčio taikyti SOM rezultatui negalima, nes čia nėra griežtai išreikštų klasterių, juos tik galima stebėti vizualiai.

3. Tekstinių dokumentų matricos sudarymas

Kad galėtume analizuoti tekstinius dokumentus įvairiais klasterizavimo metodais, būtina dokumentus paversti skaitine išraiška. Tuo tikslu sudaromas tekstinių dokumentų žodynas, o iš jo – tekstinių dokumentų matrica (6).

$$\begin{pmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1n} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & x_{N3} & \dots & x_{Nn} \end{pmatrix} \quad (6)$$

Čia x_{pl} – l-ojo žodžio pasikartojimų p-ajame dokumente skaičius, $l = 1,...,n$, $p = 1,...,N$. N – analizuojamų dokumentų skaičius, o n – žodžių skaičius dokumentų žodyne. Kiekviena matricos (6) eilutė atitinka vektorių, sudarytą iš n komponenčių. Šie vektoriai (duomenys) bus klasterizuojami SOM ir *k* vidurkių metodu. Pradžioje būtina sudaryti tekstinių dokumentų žodyną, atsižvelgiant į pasirinktus faktorius. Maksimalaus žodžių ilgio apribojimas leidžia atmesti žodžius, kurie yra per trumpi ar per ilgi. Kiekviename dokumente galima rasti trumpų žodelių, pavyzdžiui, prielinksnių ar jungtukų: į, iš, o ir kt., todėl sudarant žodyną juos reikia atmesti. Kad nebūtų iškraipomi rezultatai, būtina atmesti skaičius, numeraciją ir formules, nes tai neteikia esminės informacijos apie analizuojamą dokumentą. Labai svarbu parinkti tinkamą žodžių pasikartojimų dokumente skaičių. Nurodant per mažą skaičių, į žodyną gali būti įtraukiami visai neesminiai žodžiai. Nurodant per didelį skaičių, dokumentai gali būti atmesti, nes gali būti taip, kad juose nebus tiek kartų pasikartojančio žodžio.

Sudarant tekstinių dokumentų žodyną, galima neįtraukti dažnai vartojamus neesminius, bet visuose dokumentuose pasitaikančius žodžius iš sudaryto sąrašo. Pavyzdžiui, tokie trumpi žodeliai (jungtukai, prielinksniai, įvardžiai ir kt.): čia, kur, kada, šie, ten ir t. t. Jie visai necharakterizuoja analizuojamo dokumento. Teksto dokumentų matricai sudaryti naudotas „Text To Matrix Generator“ (TMG) įrankis (*Zeimpekis, Gallopoulos, 2005*).

4. Eksperimentinių tyrimų rezultatai

4.1. Tyrimų duomenys

Eksperimentiniuose tyrimuose nagrinėtos aštuonios dokumentų aibės, paimtos iš Lietuvos Respublikos Seimo duomenų bazės (LRS, 2013). Atsitiktiniu būdu pasirinkta po 15 panašaus ilgio skirtingų ministerijų įsakymų: Finansų, Kultūros, Susisiekimo, Sveikatos, Švietimo, Ūkio, Vidaus reikalų ir Žemės ūkio. Iš šių ministerijų įsakymų sudaryti trys duomenų rinkiniai: $X^1 = \{X_1^1, X_2^1, ..., X_{60}^1\}$, $X^2 = \{X_1^2, X_2^2, ..., X_{60}^2\}$ ir $X^3 = \{X_1^3, X_2^3, ..., X_{60}^3\}$.

Pirmąjį rinkinį sudaro: vektoriai $X_1^1, X_2^1, ..., X_{15}^1$ – I klasė (Sveikatos ministerija), vektoriai $X_{16}^1, X_{17}^1, ..., X_{30}^1$ – II klasė (Švietimo ministerija), vektoriai $X_{31}^1, X_{32}^1, ..., X_{45}^1$ – III klasė (Vidaus reikalų ministerija) ir vektoriai $X_{46}^1, X_{47}^1, ..., X_{60}^1$ – IV klasė (Žemės ūkio ministerija).

Antrąjį rinkinį sudaro: vektoriai $X_1^2, X_2^2, ..., X_{15}^2$ – I klasė (Finansų ministerija), vektoriai $X_{16}^2, X_{17}^2, ..., X_{30}^2$ – II klasė (Kultūros ministerija), vektoriai $X_{31}^2, X_{32}^2, ..., X_{45}^2$ – III klasė (Susisiekimo ministerija) ir vektoriai $X_{46}^2, X_{47}^2, ..., X_{60}^2$ – IV klasė (Ūkio ministerija).

Trečiąjį rinkinį sudaro: vektoriai $X_1^3, X_2^3, ..., X_{15}^3$ – I klasė (Finansų ministerija), vektoriai $X_{16}^3, X_{17}^3, ..., X_{30}^3$ – II klasė (Ūkio ministerija), vektoriai $X_{31}^3, X_{32}^3, ..., X_{45}^3$ – III klasė (Vidaus reikalų ministerija) ir vektoriai $X_{46}^3, X_{47}^3, ..., X_{60}^3$ – IV klasė (Žemės ūkio ministerija).

Eksperimentų metu vektorių matmenų erdvė kiekvienu atveju yra skirtinga, nes ji priklauso nuo susidariusio tekstinio dokumentų žodyno ilgio.

4.2. Tekstinių dokumentų matricos kūrimas eksperimentiniams tyrimams

Kaip jau minėta, svarbu parinkti tinkamus faktorius verčiant tekstinę informaciją į skaitinę. Dokumentuose pateikti skaičiai, numeracija nesuteikia jokios esminės informacijos, todėl sudarant tekstinių dokumentų matricą šie duomenys nebuvo įtraukiami. Tyrime analizuojama žodžių pasikartojimo ir dažniausiai vartojamų žodžių sąrašo naudojimo įtaka rezultatams. Atlikus pirminius eksperimentus pastebėta, kad tiriamai dokumentų aibei tinkamiausias maksimalus žodžių pasikartojimų skaičius dokumente yra penki. Pasirinkus didesnį skaičių sudarant tekstinių dokumentų žodyną, dokumentai buvo atmetami, nes juose tiek kartų pasikartojančių žodžių nebuvo. Tirtas žodžių pasikartojimas nuo 1 iki 5, naudojant dažniausiai vartojamų žodžių sąrašą arba ne. Iš viso kiekvienam duomenų rinkiniui

1 lentelė. ***SOM rezultatai pirmajam duomenų rinkiniui***

Pasikartojimas / Matas	Mokymo aibė					Testavimo aibė				
	1	2	3	4	5	1	2	3	4	5
Nenaudojant dažniausiai vartojamų žodžių sąrašo										
E_1	17,84	17,78	19,32	20,34	18,94	0,75	1,22	2,70	2,44	5,17
E_2	28,50	28,29	28,25	26,76	29,69	2,59	3,75	3,24	3,28	4,95
E_3	15,17	17,62	22,20	17,99	19,68	2,35	2,50	4,46	2,82	4,51
E_4	24,30	30,13	26,08	31,94	30,63	2,02	1,60	1,80	2,59	1,68
E_{center}	6,25	5,36	5,42	4,77	5,06	6,14	6,87	5,85	5,45	5,26
Naudojant dažniausiai vartojamų žodžių sąrašą										
E_1	20,72	18,63	16,56	14,10	17,47	1,20	0,93	0,58	1,54	3,14
E_2	28,56	31,97	29,67	32,11	28,15	3,67	3,60	4,02	5,79	3,91
E_3	15,31	16,04	15,35	23,03	23,62	2,02	2,77	2,96	4,49	6,78
E_4	26,78	21,38	27,32	25,69	29,35	2,93	2,97	3,19	1,41	1,27
E_{center}	5,92	6,42	5,98	5,11	4,95	5,93	7,29	7,05	5,78	4,83

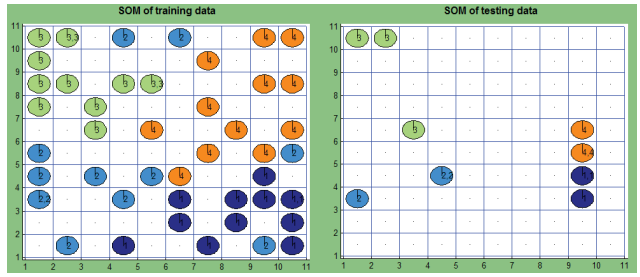
sudaryta po 10 tekstinių dokumentų matricų, kurios analizuotos saviorganizuojančių neuroninių tinklų ir *k* vidurkių metodu.

4.3. Rezultatai, gauti naudojant SOM tinklą

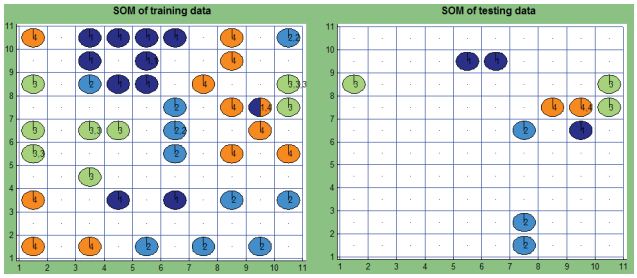
Pasirinktas 10×10 SOM žemėlapio dydis ($r = s = 10$). 80 proc. duomenų priskirti mokymo aibei, kiti 20 proc. testavimo aibei. Naudota sistema, pasiūlyta darbe (Stefanovič, Kurasova, 2011). Kiekvienas bandymas kartotas penkis kartus, esant skirtingoms žemėlapio neuronų pradinėms reikšmėms. Penkiais bandymais gautų matų reikšmių vidurkiai pateikiami 1–3 lentelėse. Kaip matome 1 lentelėje, pirmajam duomenų rinkiniui rezultatai yra gana panašūs naudojant dažniausiai vartojamų žodžių sąrašą ir jo nenaudojant. Mažiausi atstumai tarp klasių narių gauti I ir III klasėms (E_1 ir E_3), tai reiškia, kad jų nariai žemėlapyje yra išsidėstę arti vienas kito. Priešingai, II ir IV klasės atstumai (E_2 ir E_4) yra kur kas didesni ir tai rodo, jog šie duomenys pasklidę po žemėlapį plačiau. Didžiausias skirtumas tarp klasių centrų (E_{center}) gautas, kai sudarant žodyną naudojamas dažnai vartojamų žodžių sąrašas (6,42) ir žodžių pasikartojimų skaičius ne mažesnis nei 2. Testavimo aibėje atstumas siekia 7,29. Nurodant vis didesnį žodžių pasikartojimų skaičių tiek mokymo aibei, tiek testavimo aibei, atstumas tarp centrų E_{center} turi tendenciją mažėti. Tai reiškia, jog žemėlapyje pavaizduoti duomenys vis labiau susimaišo.

Pirmame paveiksle pateikiamas SOM žemėlapis pirmajam duomenų rinkiniui, iš visų bandymų išrinkus tą, kuriam matų reikšmės buvo geriausios, t. y. $E_1 - E_4$ reikšmė maža, o E_{center} – didelė. Žemėlapis, kai matų reikšmės blogiausios, vaizduojamas 2 paveiksle. SOM žemėlapis gautas (1 pav.), kai žodžių pasikartojimų skaičius yra 2 ir sudarant žodyną atmetami dažniausiai vartojami žodžiai. Paveiksle spalvos ir skaičiukai žymi klases.

Žemėlapyje matyti (1 pav.), jog I ir III (Sveikatos ir Vidaus reikalų ministerijos) klasių duomenys sudaro du atskirus klasterius, o klasės nariai išsidėstę arti vienas kito, tai ir parodė 1 lentelėje pateikti rezultatai. II ir IV (Švietimo



1 pav. ***Geriausias SOM žemėlapis pirmajam duomenų rinkiniui***



2 p a v. ***Blogiausias SOM žemėlapis pirmajam duomenų rinkiniui***

ir Žemės ūkio ministerijos) klasių duomenys plačiau pasklidę po žemėlapį, todėl jų klasių atstumų matų reikšmės (E_2 ir E_4) gautos kur kas didesnės. Dešinėje pateikti testavimo rezultatai rodo, jog duomenys išsidėsto grupelėmis maždaug tose pačiose vietose, kur ir mokymo aibės žemėlapyje pateikti duomenys. SOM žemėlapis, kuriame matų reikšmės yra mažiausios, gautas, kai pasirinktas žodžių pasikartojimų skaičius 4 ir nenaudotas dažniausiai vartojamų žodžių sąrašas (2 pav.).

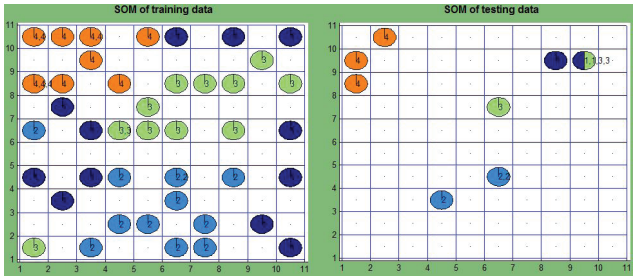
Šiuo atveju matome, jog duomenys plačiai išsibarsto po visą žemėlapį. Susiformavo tik nedideli I klasės klasteriai viršuje, III klasės kairėje, II klasės nariai apatiniame kampe ir IV klasės nariai viršutiniame dešiniajame kampe. Į vieną langelį patenka po vieną I ir IV klasių narį.

Atlikus antrojo duomenų rinkinio tyrimus matyti (2 lentelė), kad mažiausios pirmojo mato reikmės (E_2 ir E_4) gautos II ir IV klasei (Kultūros ir Ūkio ministerijos). Šių klasių duomenys žemėlapyje išsidėsto arčiau vieni kitų tiek mokymo, tiek testavimo aibei. I ir III (Finansų ir Susisiekimo ministerijos) klasių duomenų rezultatai gerokai blogesni. Kaip ir pirmojo duomenų rinkinio analizės atveju, didinant žodžių pasikartojimų skaičių, atstumai tarp centrų (E_{center}) turi tendenciją mažėti, duomenys pasiskirsto plačiau. Palyginus atstumus tarp centrų mokymo aibei matyti, kad geresni (didesnė mato reikšmė) rezultatai gaunami, kai sudarant dokumentų žodyną dažniausiai vartojamų žodžių sąrašas nenaudojamas.

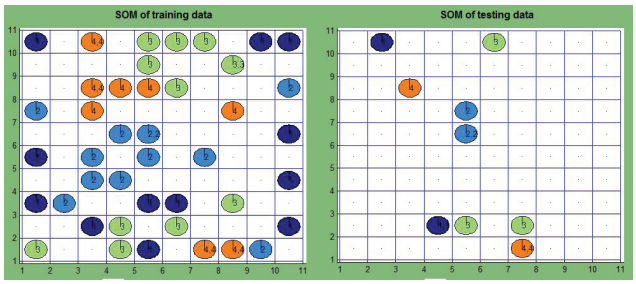
Geriausias SOM žemėlapis pagal pasiūlytus matus gautas, kai žodžių pasikartojimas lygus 1 ir sudarant žodyną naudojamas dažniausiai vartojamų žodžių sąrašas (3 pav.). Mokymo aibės kairiajame viršutiniame žemėlapio kampe išsidėsto visi IV klasės nariai (Ūkio ministerija), apačioje dauguma II klasės narių (Kultūros ministerija), per vidurį III klasės nariai (Susisiekimo ministerija), o I klasės nariai (Finansų ministerija) išsibarsto po visą SOM žemėlapį. Pagal 2 lentelėje pateiktus rezultatus, mažiausi atstumai (E_2 ir E_4) yra tarp IV ir II klasių narių ir tai matoma pateiktame žemėlapyje (3 pav.). Plačiausiai išsidėsto I klasės duomenys.

2 l e n t e l ė. ***SOM rezultatai antrajam duomenų rinkiniui***

Pasikartojimas	Mokymo aibė					Testavimo aibė				
Matas	1	2	3	4	5	1	2	3	4	5
Nenaudojant dažniausiai vartojamų žodžių sąrašo										
E_1	28,02	34,40	32,36	28,82	33,38	3,45	4,33	3,44	4,17	4,26
E_2	19,86	20,88	19,91	22,88	22,52	1,56	2,06	1,56	0,67	0,83
E_3	28,35	25,84	25,14	26,40	26,62	5,06	4,79	5,10	4,48	5,02
E_4	14,95	15,12	16,98	22,24	21,26	1,49	2,30	2,88	4,12	3,59
E_{center}	5,98	4,92	4,99	4,61	3,81	7,01	7,11	6,29	6,06	4,96
Naudojant dažniausiai vartojamų žodžių sąrašą										
E_1	31,97	29,26	30,64	29,52	29,93	2,97	4,06	4,96	5,27	4,44
E_2	19,31	22,12	18,70	20,97	23,26	1,51	1,70	1,92	0,78	1,00
E_3	22,51	24,66	25,71	24,51	29,07	4,84	4,73	4,11	3,24	5,70
E_4	14,64	20,13	18,89	18,94	20,78	1,68	3,31	3,33	2,16	3,60
E_{center}	6,36	5,49	5,18	5,99	4,60	7,39	6,77	4,86	6,70	4,99



3 p a v. ***Geriausias SOM žemėlapis antrajam duomenų rinkiniui***



4 p a v. ***Blogiausias SOM žemėlapis antrajam duomenų rinkiniui***

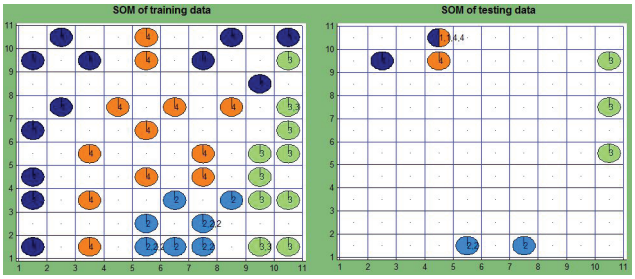
Blogiausias SOM rezultatas pagal tiriamus matus gautas, kai pasikartojimų skaičius 5 ir nenaudojamas dažniausiai vartojamų žodžių sąrašas. Matome (4 pav.) tik nedidelius skirtingų klasių klasterius, tačiau iš esmės visų klasių duomenys plačiai išsidėsto įvairiose žemėlapio vietose.

Atlikus eksperimentus naudojant trečiąjį duomenų rinkinį matyti, kad mažiausios matų reikšmės (E_2 ir E_3) gautos II ir III klasėms (Ūkio ir Vidaus reikalų ministerijos). I ir IV klasėms (Finansų ir Žemės ūkio ministerijos) matų reikšmės (E_1 ir E_4) yra didžiausios. Didinant pasikartojimo skaičių mokymo ir testavimo aibėms, analogiškai kaip antrajam duomenų rinkiniui, atstumai tarp centrų E_{center} turi tendenciją mažėti. Šiuo atveju, kai pasikartojimų skaičius nuo 1 iki 3, didesni atstumai tarp centrų E_{center} gaunami nenaudojant dažniausiai vartojamų žodžių sąrašo, o kai pasikartojimų skaičius 4 ir 5, naudojant sąrašą rezultatai yra geresni.

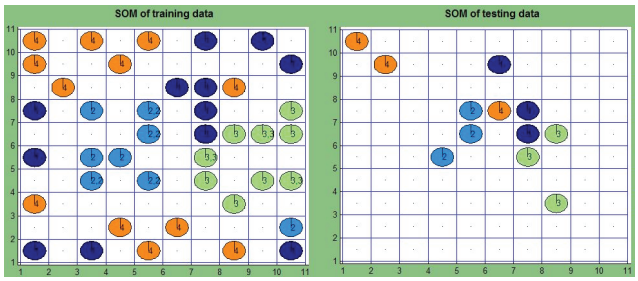
Šiuo atveju geriausias pagal pasiūlytų matų reikšmes SOM žemėlapis gautas, kai žodžių pasikartojimų skaičius 1 ir dažniausiai vartojamų žodžių sąrašas nenaudotas. Aiškiai matomi mokymo aibės žemėlapio apačioje išsidėstę II klasės nariai (Ūkio ministerija), dešinėje – III klasė (Vidaus reikalų ministerija), per vidurį išilgai nuo apačios iki viršaus – IV klasės nariai (Žemės ūkio ministerija) ir kairėje pusėje – dauguma I klasės narių (Finansų ministerija).

3 l e n t e l ė. ***SOM rezultatai trečiajam duomenų rinkiniui***

Pasikartojimas / **Matas**	**Mokymo aibė**					**Testavimo aibė**				
	1	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
Nenaudojant dažniausiai vartojamų žodžių sąrašo										
E_1	29,98	31,65	30,11	30,22	32,77	1,93	5,13	4,26	3,96	4,36
E_2	11,49	18,47	17,41	21,78	21,45	1,65	3,08	2,96	4,03	4,24
E_3	17,18	22,67	19,75	19,63	23,60	2,57	3,06	3,25	2,66	5,84
E_4	26,00	26,29	29,64	30,11	27,87	1,54	4,85	2,69	3,22	1,28
E_{center}	6,68	5,32	5,14	4,62	4,33	7,86	6,06	4,89	3,84	4,93
Naudojant dažniausiai vartojamų žodžių sąrašą										
E_1	30,63	33,42	33,44	32,65	30,43	4,80	3,94	2,75	4,32	3,99
E_2	12,52	14,46	15,82	16,86	18,46	1,95	2,30	1,84	2,94	3,85
E_3	14,48	17,47	21,25	21,76	26,09	2,32	1,98	3,22	4,23	6,35
E_4	27,92	30,09	30,52	29,43	30,21	3,08	1,26	3,27	2,85	1,78
E_{center}	6,15	5,15	3,89	4,83	4,41	7,33	6,38	5,03	5,00	3,98



5 p a v. ***Geriausias SOM žemėlapis trečiajam duomenų rinkiniui***



6 p a v. ***Blogiausias SOM žemėlapis trečiajam duomenų rinkiniui***

Blogiausi rezultatai pagal pasiūlytus matus gauti, kai žodžių pasikartojimų skaičių 3 ir naudojamas dažniausiai vartojamų žodžių sąrašas. Kadangi II klasės nariai atitinka Ūkio ministerijos dokumentus, o IV klasės – Žemės ūkio ministerijos, galima pamatyti, kad šios klasės tiek 5, tiek 6 paveiksluose yra išsidėsčiusios greta, tai rodo šių dokumentų panašumą. Lygiai taip pat I klasės (Finansų ministerija) ir III klasės (Vidaus reikalų ministerija) dokumentuose nagrinėjami panašūs klausimai, todėl natūralu, jog I ir III duomenys yra greta.

4.4. Rezultatai, gauti naudojant k vidurkių metodą

Tyrimams *k* vidurkių metodu atlikti naudota *Matlab* sistemos funkcija *k-means*. Imant vis kitas pradines klasterių centrų koordinates, bandymai buvo kartoti po 10 kartų. Kaskart buvo vertinamas klasių priskyrimas klasteriams (4 lentelėje tai nurodoma „Neteisingai priskirti“) bei atstumų tarp klasterių centrų iki jiems priskirtų duomenų sumos (4 lentelėje – AKS). Dešimties bandymų metu gautų įverčių vidurkiai pateikiami 4 lentelėje.

Iš pateiktų rezultatų (4 lentelė) matyti, kad nagrinėjant visus tris duomenų rinkinius, didinant žodžių pasikartojimų skaičių tekste ir sudarant žodyną nenaudojant dažniausiai vartojamų žodžių sąrašo, pastebimas neteisingai priskirtų duomenų pagal klases skaičiaus didėjimas, tačiau klasterizavimo rezultatai pagal AKS kaskart gerėja, neskaitant vieno atvejo antrajam duomenų rinkiniui, kai pasikartojimų skaičius lygus 4. Todėl naudojant *k* vidurkių metodą sunku pasakyti, kuris rezultatas yra geriausias, nes reikia atsižvelgti tiek į klasterizavimo kokybę, tiek į klasių suskirstymą klasteriams. Atlikus eksperimentus, kai sudarant žodyną naudojamas dažniausiai vartojamų žodžių sąrašas, matoma, kad klasterizavimo rezultatai gerėja didinant pasikartojimų skaičių, tačiau klasių priskyrimo klasteriams rezultatai yra įvairūs.

4 l e n t e l ė. ***K vidurkių metodu gauti eksperimentiniai rezultatai***

Pasikartojimas / Matas	Nenaudojant dažniausiai vartojamų žodžių sąrašo					Naudojant dažniausiai vartojamų žodžių sąrašą				
	1	2	3	4	5	1	2	3	4	5
Pirmajam duomenų rinkiniui										
Neteisingai priskirti	18,3	23,6	26,6	30,7	33,2	22,8	20,1	21,5	27,3	32,1
AKS	43608	40513	35885	31221	26923	40112	37259	32887	29115	24972
Antrajam duomenų rinkiniui										
Neteisingai priskirti	21,9	23,3	24,7	28,8	28	25	25,2	23,9	26,5	28,9
AKS	36796	34209	30580	26469	22719	35732	32302	28056	24934	21585
Trečiajam duomenų rinkiniui										
Neteisingai priskirti	19,9	21,9	27,6	32,5	35,1	21,7	27,7	26,8	32,7	34,2
AKS	39383	37593	33090	28965	24125	37124	34765	32004	26021	21524

Lyginant rezultatus, kai sudarant žodyną naudojamas dažniausiai vartojamų žodžių sąrašas ir ne, matoma, kad analizuojant visus tris duomenų rinkinius gaunama mažesnė mato AKS reikšmė, kai vartojami dažniausi žodžiai. Gaunami geresni klasterizavimo rezultatai. Nagrinėjant klasių priskyrimus klasteriams matyti, jog pirmajam duomenų rinkiniui, nurodant pasikartojimų skaičių nuo 2 iki 5, klasių priskyrimas tinkamiems klasteriams yra geresnis, kai naudojamas dažniausiai vartojamų žodžių sąrašas. Antrojo ir trečiojo duomenų rinkinių atvejais klasių priskyrimo klasteriams rezultatai yra gana panašūs.

Išvados

Straipsnyje nagrinėjama tekstinių dokumentų panašumo paieška naudojant du populiarius metodus – saviorganizuojantį neuroninį tinklą ir *k* vidurkių metodą. Tirta kelių tekstinių dokumentų matricos sukūrimo faktorių įtaka gautiems rezultatams. SOM kokybei įvertinti pasiūlyti du nauji matai. *K* vidurkio metodo kokybei įvertinti skaičiuotos atstumų nuo klasterių centrų iki klasterių narių sumos ir nustatomas klasių ir klasterių atitikimas. Klasių ir klasterių atitikimą SOM tinkle vienareikšmiškai įvertinti neįmanoma, nes čia negalima duomenų vienareikšmiškai priskirti klasteriams, kaip įmanoma *k* vidurkių metodu, klasterius galima tik stebėti vizualiai. Tekstinių duomenų panašumo analizė taikant saviorganizuojantį neuroninį tinklą yra informatyvesnė nei taikant *k* vidurkių metodą, nes naudojant SOM gaunami ne tik skaitiniai įverčiai, bet ir žemėlapiai, kuriuose matyti tekstinių dokumentų išsidėstymas, o tai leidžia vizualiai vertinti dokumentų panašumus.

Atlikti tyrimai parodė, jog geresni rezultatai pagal pasiūlytus matus gaunami, kai sudarant žodyną naudojamas dažniausiai vartojamų žodžių sąrašas. Tiek SOM, tiek *k* vidurkių metodas pateikė geresnius rezultatus. Teksto žodžių pasikartojimų skaičius dokumente taip pat yra svarbus. Tyrimai parodė, jog per didelis ar per mažas žodžių pasikartojimų skaičius rezultatus blogina. Nagrinėjant tekstinius duomenis, apie kuriuos turima mažai informacijos, sudarant tekstinę dokumentų matricą rekomenduotume naudoti dažnai vartojamų žodžių sąrašą ir žodžių pasikartojimų skaičių parinkti ne mažesnį nei 3. Naudojant saviorganizuojančius neuroninius tinklus, pirmojo mato reikšmės ($E_1 - E_4$) labiau nesiskiria priklausomai nuo žodžių pasikartojimų skaičiaus, tačiau didinant šį skaičių atstumas tarp klasių centrų E_{center} mažėja, o tai reiškia, kad rezultatai gaunami blogesni. Kai žodžių pasikartojimų skaičius didėja, klasterizavimo *k* vidurkių metodu rezultatai AKS mato prasme gaunami geresni, t. y. mažesnės sumos tarp klasterių centrų ir jų narių. Be to, naudojant dažniausiai vartojamų žodžių sąrašą gaunami geresni rezultatai. Tačiau, didėjant žodžių pasikartojimų skaičiui, klasių priskyrimų tinkamiems klasteriams rezultatai blogėja.

Atlikus eksperimentus naudojant trečiąjį dokumentų rinkinį matyti, kad nors dokumentai yra skirtingų klasių, tačiau pagal sudarytą tekstinių dokumentų matricą Finansų ir Vidaus reikalų ministerijų tekstiniai dokumentai yra panašūs. Taip pat panašumų matoma ir tarp Ūkio ir Žemės ūkio ministerijų dokumentų. Naudojant pirmąjį ir antrąjį duomenų rinkinius, aiškių panašumų tarp skirtingų klasių negauta.

Padėka. Šis tyrimas atliktas Europos socialinio fondo finansuojamo projekto „Paslaugų interneto technologijų kūrimo ir panaudojimo našių skaičiavimų platformose teoriniai ir inžineriniai aspektai“ (Nr. VP1-3.1-ŠMM-08-K-01-010) lėšomis. Straipsnio autoriai dėkoja recenzentams už pastabas ir pasiūlymus, kurie leido iš esmės pagerinti straipsnį.

LITERATŪRA IR ŠALTINIAI

ALSMADI, Izzat; SALEH, I. Zakaria (2012). Documents Similarities Algorithms for Research Papers Authenticity. *ICCIT*, p. 210–214.

CHOW, W. S. Tommy; RAHMAN, M. K. M. (2009). Multilayer SOM With Tree-Structured Data for Efficient Document Retrieval and Plagiarism Detection. *IEEE Transactions on Neural networks*, vol. 20, no. 9.

DZEMYDA, Gintautas; KURASOVA, Olga; ŽILINSKAS, Julius (2008). *Daugiamačių duomenų vizualizavimo metodai*, p. 108–127.

KOHONEN, Teuvo (2001). *Self-organizing Maps*. 3rd ed. Springer series in information sciences. Berlin: Springer-Verlag. 506 p. ISBN 3540679219.

KOHONEN, Teuvo; XING, Hongbing (2011). Contextually Self-Organized Maps of Chinese Words. In: J. Laaksonen, T. Honkela, eds. Advances in Self-Organizing Maps – WSOM 2011, *Lecture Notes in Computer Science*, vol. 6731, Heidelberg: Springer Verlag, p. 16–29.

LIETUVOS RESPUBLIKOS SEIMAS (LRS) [interaktyvus] [žiūrėta 2013 m. gegužės 2 d.]. Prieiga per internetą: <http://www3.lrs.lt/dokpaieska/forma_l.htm>.

MACQUEEN, J. (1967). Some methods for classification and analysis of multivariate observations. In: Le Cam, L. M. and Neyman, J., editors. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Statistics.* Berkeley and Los Angeles: University of California Press, p. 281–297.

MAIA, E. B. Jose; BARRETO, A. Guilherme; COELHO, V. Andre Luis (2011). Evolving a Self-Organizing Feature Map for Visual Object Tracking. In: J. Laaksonen, T. Honkela, eds. Advances in Self-Organizing Maps – WSOM 2011, *Lecture Notes in Computer Science*, vol. 6731, Heidelberg: Springer Verlag, p. 121–130.

MARINAI, Simone (2011). Text from early printed books. *International Journal on Document Analysis and Recognition – IJDAR*, vol. 14, no. 2, p. 117–129.

MAYER, Rudolf (2011). Analysing the Similarity of Album Art with Self-organizing maps. In: J. Laaksonen, T. Honkela, eds. Advances in Self-Organizing Maps – WSOM 2011, *Lecture Notes in Computer Science*, vol. 6731, Heidelberg: Springer Verlag, p. 357–366.

MAYER, Rudolf.; RAUBER, Andreas (2011). On Wires and Cables: Content Analysis of WikiLeaks Using Self-Organising Maps. In: J. Laaksonen, T. Honkela, eds. Advances in Self-Organizing Maps – WSOM 2011, *Lecture Notes in Computer Science*, vol. 6731, Heidelberg: Springer Verlag, p. 238–246.

MOLYTĖ, Alma (2011). *Vektorių kvantavimo metodų jungimo su daugiamatėmis skalėmis analizė*: Daktaro disertacija. Vilniaus universitetas.

SIHAG, K. Vikas; KUMAR, Subhash (2013). Graph based Text Document Clustering by Detecting Initial Centroids for k-means. *International Journal of Computer Applications* (0975–8887), vol. 62, no. 19, p. 1–4.

SJOBERG, Mats; LAAKSONEN, Jorma (2011). Analysing the Structure of Semantic Concepts in Visual Databases. In: J. Laaksonen, T. Honkela, eds. Advances in Self-Organizing Maps – WSOM 2011, *Lecture Notes in Computer Science*, vol. 6731, Heidelberg: Springer Verlag, p. 338–347.

STEFANOVIČ, Pavel; KURASOVA, Olga (2009) Saviorganizuojančių neuroninių tinklų sistemų lyginamoji analizė. *Informacijos mokslai*, t. 50, p. 334–339.

STEFANOVIČ, Pavel; KURASOVA, Olga (2011). Visual analysis of self-organizing maps. *Nonlinear Analysis: Modeling and Control*, vol. 16(4), p. 488–504.

STEFANOVIČ, Pavel; KURASOVA, Olga (2011). Influence of Learning Rates and Neighboring Functions on Self-Organizing Maps. In: J. Laaksonen, T. Honkela, eds. Advances in Self-Organizing Maps – WSOM 2011, *Lecture Notes in Computer Science*, vol. 6731, Heidelberg: Springer Verlag, p. 141–150.

STEINBACH, Michael; KARYPIS, George; KUMAR, Vipin (2000). Comparison of document clustering techniques. *KDD Workshop on Text Mining.*

ZEIMPEKIS, Dimitrios; GALLOPOULOS, Efstratios (2005). *TMG: A Matlab Toolbox for Generating Term-Document Matrices from Text Collections.* Technical Report HPCLAB-SCG 1/01-05, University of Patras, GR-26500, Patras, Greece.

SIMILARITY ANALYSIS OF TEXT DOCUMENTS BY SELF-ORGANIZING MAPS AND *K*-MEANS

Pavel Stefanovič, Olga Kurasova

S u m m a r y

In this paper, we try to find similarities of different text documents by the self-organizing map (SOM) and *k*-means method. One of the main goals of these methods is to cluster a dataset. Using SOM, the similarities of documents can be observed visually. Both methods can be used only for numerical information, so we analyse the different options by converting text data on to numerical in order to get better results. To estimate the SOM quality, when the classified data are analysed, we propose two new measures: distances between SOM cells, corresponding to data items assigned to the same class, and the distance between centres of SOM cells, corresponding to different classes. We also analyse the results of visualization by self-organizing maps. In order to estimate the *k*-means quality, we calculate the sum of distances between cluster centres and class members and also we estimate assignment of the data from particular classes to the clusters. The experiments have been carried out using three datasets ocquired from the document database of Seimas of the Republic of Lithuania.

INTERNETINĖS TECHNOLOGIJOS

Possibilities of automatic and semi-automatic end-user driven web service composition

Dalė Dzemydienė

Mykolas Romeris University, Institute of Communication and Informatics, Professor, Dr.
Mykolo Romerio universiteto Komunikacijos ir informatikos instituto profesorė, daktarė
Ateities str. 20, LT-08303 Vilnius
E-mail: daledz@mruni.eu

Arūnas Miliauskas

Vilnius University, Institute of Mathematics and Informatics Doctoral student
Vilniaus universiteto Matematikos ir informatikos instituto doktorantas
Akademijos str. 4, LT-08663 Vilnius
E-mail: arunas.miliauskas@mii.vu.lt

Our research work relates to the main principles, means and current limitations of the end-user driven automatic and semi-automatic web service composition. It analyses automatic and semi-automatic composition approaches found in literature and classifies them as workflow-based, template-based and automatic methods. The aim of this research is to provide a proposal how to construct semi-automatic or automatic end-user driven web service composition. An approach is illustrated by the multi-complexity of service composition in travel domain. We analyze a conceptual solution that covers the whole composition process: from an end-user submitting composition requirements until the presentation of the a composition execution results. Some methods of an artificial intelligence (AI) planning research field were used in proposed web service composition approach.

Introduction

Nowadays methods and tools for web service composition are becoming the mainstream information technology. Web services can provide a more helpful way for accessing functionality over the network. When implementing new software solution there is often a good chance that some functions are already implemented and can be used by accessing them via web services. But rarely a single service can provide all needed functionality. Therefore the composition of several web services is needed for realization of multi-complex end-user needs.

We would like to use the definition of web services composition and execution as *the process of realization of the requirements of new services using the capability specification of the existing component services.* This definition is proposed by (Agarwal et al., 2008) and helps us to explain which web service composition method is more useful in implementation.

The web service composition is not a trivial task. Some issues should be taken into account when dealing with service composition:

- Number of web services increases and service repositories expected to grow large (Rao and Su, 2005). Therefore, a discovery of suitable services for a particular task can be difficult.
- Web services can be updated and existing compositions may become obsolete. For composition that uses many web services it may require a lot of effort to keep it up to date.

- Services are developed by different organizations, based on different conceptual models. Even for a software developer it can be a difficult task that often leads to many errors or takes a long time to understand the concrete meaning of data in different services.
- Current web services description languages like Web Services Description Language (WSDL) describe only syntax (not semantics) (Akkiraju et al., 2005). Therefore, a manual work is required for a discovery, invocation, and composition of these services.

All the mentioned issues show that web service composition is based on current technologies, architectures, languages, protocols targeted to software developer and is time-consuming, expensive, error prone and leads to very constrained solutions.

One of potential solutions tackling these problems is an automatic or semi-automatic service composition. A level of automation here may vary depending on level of involvement of the software engineers or end users in this process. Here by a semi-automatic service composition we mean that an end-user has to select abstract process and most relevant and suitable services are used from the proposed list of compatible services. Those services are discovered automatically, and a user is not concerned about data mapping or integration. In full automation scenarios composite process is synthesized and services selected based on user goal and service descriptions, without any user intervention.

The aim of this research is analyze possibilities and provide brief description of current trends, activities, issues and problems in automatic, semi-automatic web service composition. The review is presented by comparing main characteristics of different web service composition approaches. We provide classification of these approaches, identifying and selecting some methods which are most suitable and promising for semi-automatic end-user driven web service composition. An approach is illustrated by an example useful in travel domain and shows the end-user driven automatic web service composition possibilities. As a result the conceptual solution of system architecture for semi-automatic web service composition is proposed.

Possibilities of automatic and semi-automatic web service composition

The idea of automatic or semi-automatic web service composition is very promising. It was envisioned that Semantic Web movement could solve this problem because its goal is *"a web of data that can be processed directly and indirectly by machines"* (Berners-Lee, 2001). However, this idea was not realized to its full potential yet. This is shown by the statement created by one of the co-authors of this idea: *"This simple idea... remains largely unrealized."* (Shadbolt et al., 2006). Therefore a lot of research activities are carrying on and many research papers have been published for developing the automatic web service integration and composition possibilities.

The classification of automatic web service composition approaches is presented by (Kim et al., 2009; Rao and Su, 2005; Diagiampetri et al., 2007). Following some approaches, we can identify some commonalities and differences in the process of production of a service composite workflow:

- In workflow-based approaches the composite process is viewed as a workflow and presented as a directed acyclic graph. The process requires domain knowledge and developer involvement.
- In template-based approaches templates are predefined, created and an end-user can select one of them to create a workflow. User empowerment depends on template extensibility.
- In approaches based on AI planning it is a way to generate process automatically from problem specification. In most cases of AI planning implementations user domain knowledge and developer intervention are not required.

Based on the classification proposed by (Rao and Su, 2005) we would like to classify service composition approaches by using slightly different groups of definitions and provide arguments for that.

In (Kim et al., 2009; Rao and Su, 2005) an attempt was made to define the process of automatic service composition. We select one defined in (Kim et al., 2009). It consists of the following phases:

- Specification phase: provide an easy way for a user to specify task goals, requirements and constraints without extensive domain knowledge;
- Planning phase: provide an automatic way to compose an abstract workflow based on the specification;
- Validation phase: provide techniques to ensure that the composite process realized via an abstract workflow satisfies the user's stated task goals;
- Discovery phase: provide a way to discover services that satisfy task specifications in the workflow;
- Execution with monitoring phase: provide a framework for monitoring executing service and provide automatic fault-handling mechanisms.

Automatic or semi-automatic composition can be classified according different dimensions. Here we analyze 3 groups of approaches based on methods how an end-user (without any programming background) is involved in the creation of a service composition workflow:

- Workflow-based: an end-user is responsible for creating a composition workflow.
- Template-based: an end-user doesn't create a workflow himself / herself, the most relevant template can be chosen to simplify composition.
- Automatic: a user provides high-level goals and based on that composition is synthesized.

In workflow-based approaches composition is seen as workflow (Casati et al., 2001). Therefore web services are composed by creating control and data flow among them. Workflow-based approaches split into two groups (Rao and Su, 2005): static or dynamic workflow generation.

The difference between them lies on level of user involvement. In static approach the user creates an abstract process model and only web service selection and binding are done automatically. On the other hand, in the dynamic approach a process model and service selection are made automatically. Combined approaches are also mentioned.

One of the approaches discussed is template-based service composition. Using this method, a workflow template is selected and it is bound to specific web services. Most papers promote this approach as the most practical one (Kim et al., 2009; Mehandjiev et al., 2010). One of the advantages of this approach is a compromise between flexibility, user empowerment and encapsulation of complexity that is crucial for a user without any programming background. Template-based approaches differ in level of abstraction of templates and languages used for template specification. Most common languages are OWL-S (Kim et al., 2009; Sirin et al., 2005), BPEL with WSDL (Geebelen et al., 2008; Jie et al., 2008). In most cases languages are extended with special constructs to be suitable for template specification.

SOA4All project (2008-2011) aimed at integrating SOA with the Web 2.0 and Semantic Web (Domingue et al., 2009) and one of the addressed challenges was automated web service composition. The project research results are related to the topic of the template based composition (Mehandjiev et al., 2010; Lécué et al., 2010). Mehandjiev et al. (2010) proposes an approach of assisted service composition for end users. Successful compositions are saved into reusable templates where technical details are hidden from the user. Semantic technologies are used to hide service composition complexity such as control or data dependencies. The user is qualified to select a template from a list and choose a preferred service. Still we couldn't find information what language was used to specify a composition template. Another paper (Lécué et al., 2010) describes an approach where composition templates are generated from log files.

A composition is synthesized from generated and already existing templates using a parametric design procedure with a blackboard based multi agent system and later is optimized by changing services to the most relevant ones. Again, no information regarding template format is provided.

Other approaches (Geebelen et al., 2008; Jie et al., 2008) modify BPEL with a special annotation to define templates. Geebelen et al. (2008) proposes a method where BPEL is rendered using a similar approach as dynamic web page content is generated and send to a browser. A model–view-controller (MVC) pattern with a ruby-on-rails framework is used. The template in this approach represents a view in this pattern and is BPEL with special dynamic annotations, and is created at a runtime the same way as HTML web page is created for a user. In another research (Jie et al., 2008) BPEL is extended with special tags for a template definition. In the template abstract services are marked with partner tags which during template instantiation are replaced with real web services definitions. In all BPEL based approaches template instantiation produces an executable BPEL workflow.

Another workflow template based approach is presented in a paper (Sirin et al., 2005). It adheres to an earlier work of (Sirin et al., 2004) where a method of translation OWL-S process model to SHOP2 domain was presented. But in addition to the previous work, the following one (Sirin et al., 2005) explored possibilities of creating workflow templates and using them to create flexible compositions. OWL-S is seen as a specification language for these templates. It provides enough control flow elements like *Perform, Sequence, Choice,* etc. for defining *Composite_Process* from lower grain components - atomic processes. But this is not enough for creating abstract workflows. Therefore OWL-S was extended with the new type of process – the so called *Abstract_Process*. This *Abstract_Process* is not bound to concrete web services and is used for an abstract template specification. This template is mapped to the Hierarchical Task Network (HTN) domain, where a non-primitive task represents an abstract activity. In the HTN planning (Ghallab et al., 2004) the tasks are decomposed by methods to subtasks (and these further) until all tasks are broken down to primitive tasks. HTN-DL is an extended HTN formalism and is used to compose a process from template definition.

A web service composition using Situation calculus was presented by McIlraith and Son (2002). The situation calculus extends first order logic with actions and situations. Golog is a language built on top of situation calculus. It provides constructs (like sequences, nondeterministic choices, conditionals, loops) which enable defining a complex action and describing how they are comprised of primitive actions. Since originally Golog was created without information gathering – sensing actions, these types of actions are also added to this approach. User requirements are specified as preferences, defining which action is desired in a particular situation. A plan is synthesized using A^* planner originated from the Simple Breadth First Planner (Reiter, 2001).

Automatic web service composition

Under the automatic web service category fall approaches in which no pre-specified control flow exists, and a workflow is synthesized from user's goals, inputs outputs, pre-conditions and effects (IOPE) of individual services. Many papers (Kim et al., 2009, Rao and Su 2005) call these methods "AI planning" because in the majority of works AI planning methods are used to create compositions. But we see this definition ambiguous because some of template based composition approaches also use AI planning methods, for example, HTN (Sirin et al., 2005).

Here we accept AI planning definition provided by (Ghallab et al., 2004) where AI planning is defined as computational study of a deliberation process (planning process) that, aiming to achieve some pre-stated objectives, chooses and organizes actions by anticipating their outcomes. There is no statement that templates or any other predefined organization of actions cannot be used in AI planning. Therefore we do not refer to AI planning in our classification.

Based on a paper (Digiampietri et al., 2007) we define automatic web service composition as a process where based on user provided goals the system automatically creates a composition that satisfies these goals.

Sirin et al. (2004) suggests using HTN planning for an automatic web service composition. The paper shows how OWL-S service descriptions can be transformed to SHOP2 domain. But for this approach OWL-S composite service definition must exists. Also, this method cannot handle OWL-S composite processes where Split, Split – Join constructs are used. This is due to the fact that SHOP2 planner cannot handle concurrency. Another constraint is that atomic processes in OWL-S must be either world altering or information providing service, but not both. The reason is that the method uses interleaved planning, where, during planning process, information-providing services are executed and world-altering services are simulated (based on input, output, preconditions and effects).

Another approach was presented by Pistore et al. (2005). Here composite service is synthesized from abstract BPEL4WS processes. These abstract BPEL4WS processes define an interaction protocol with component web services. Each of them is translated to a state transition system (STS). Then based on composition requirements new STS (for composite process) is generated which is automatically translated to a composite executable BPEL4WS program. Synthesis approach is based on symbol model checking and uses system MBP that is built on top of symbol model checker named NuSMV. Composition requirements are expressed in the EAGLE language. This method shows that it is possible to create composition having only composition requirements and a definition of abstract processes.

The automatic composition (when no additional information is used, apart from composition requirements, goals and IOPE is required from end-user) has the following benefits and advantages:

- Extreme flexibility is mentioned by (Svatek and Vacura, 2005).
- The only approach when predefined templates do not exist and end-user lacks domain knowledge (Kim et al., 2009).

Despite that, the following disadvantages of automatic web service composition have been reported:

- Unpredictable results if all conditions are not perfectly specified (Svatek and Vacura, 2005).
- Composition effort is estimated bigger than in template based approach due to higher search space (Agarwal et al., 2008). Therefore it is difficult to synthesize a correct composition and this approach is not trusted in real-world scenarios yet (Kim et al., 2009).
- Lack of user empowerment and interaction (Kim et al., 2009).

Here we argue that if knowledge about relations between tasks, goal and services exists then composition can be efficient as in a template-based scenario. Such functions can be provided by domain ontology. Ontology is specified by domain experts and by using it we can avoid software developer involvement and give more empowerment to an end-user.

An example of end-user driven composition in travel domain

In order to illustrate the main features of user driven web service composition possibilities we provide an example. We choose a travel domain in which we would like to demonstrate multi complexity of web service composition requirements and then provide composition walkthrough. For instance, John, an end user, lives in Lithuania, and decides to have a leisure weekend in Paris. In order to arrange this travel himself he needs to:

- Select appropriate means of transport. That may include various types of transport, like:
  - finding plane tickets;
  - renting a car;
  - finding a bus to get from the airport.
- Choose potential points of interest (libraries, monuments, restaurants etc.).

- Find accommodation (a hotel, a hostel, a camping site, etc.). Accommodation type depends on the traveller's preferences and type of travel (group of students travelling across Europe will have different requirements than a business person going to arranged meeting).

All required activities can be accomplished by the user via internet, most likely all transport, accommodation service providers have web sites with all important information and all needed services can be booked and purchased online.

However, arranging this requires a lot of manual work: searching numbers of web sites for relevant information, purchasing the appropriate tickets at different web stores.

Some travel, accommodation, transport service providers created web services and the tasks currently performed by end user could potentially be accomplished by a software agent. However, these compositions are currently created by a software developer and are restricted to very simple scenarios (e. g., business travel).

Ability for the end user based on preferences, context to create a web service composition, execute it without any assistance of a software developer, is still a promising and yet unresolved issue. We provide our vision of the solution in the scenario walkthrough.

Based on arguments above, we select an automatic composition as the most relevant in provided scenario. We make an assumption that travel and accommodation service providers have created semantically annotated services. There are tools for this purpose like web service modeling ontology (WSMO) or ontology in web ontology language (OWL) for semantic web service specification (OWL-S). These specifications are included into domain ontology by domain experts.

Domain ontology is one of the main prerequisites for this approach. The planning and travel domain should be specified by domain experts.

Based on the arguments by Sirin et al. (2004) we select an HTN planning method and planner SHOP2 for web service composition.

Capturing the end-user travel requirements

In order to create a composition an end-user must be able to submit composition requirements in a convenient way. Many papers support an idea providing requirements as goals, but few of them emphasize distinction between planning goals and composition high-level goals (Portchelvi et al., 2012). We agree that it would be impossible for an end-user without any software engineering knowledge and define a planning problem in an AI planner (e. g., SHOP2) understandable format.

In order to translate high end-user requirements to the planning problem we adhere to a Goal-Based Service Framework (GSF) proposed approach (da Silva Santos et al., 2009). There should exist a travel domain specification (created by domain experts) based on Goal-Based Service Ontology (GSO). Domain specification supports an end-user goal, task decomposition and mapping tasks with services. Apart from domain concepts, the ontology may formally specify their relationships, dependencies, and example cases (Dzemydienė et al., 2011).

In our travel domain example we envision that an end-user does not provide all travel parameters, options, but rather provides a goal – this is based on a predefined travel template. Thus in our travel scenario John decides to have a leisure type weekend trip. He provides a goal – to have a leisure weekend trip, the type of trip that is achieved by fulfilling *Leisure_weekend_trip_task* – the structure of which is defined by *Leisure_weekend_trip_pattern*. That means the trip will have same structure (stop points, transport types, accommodation etc.) as defined in the leisure weekend trip pattern.

Houda et al. (2010) provides public transportation ontology for the travel domain and defines journey patterns, like *Leisure_journey_pattern*. Using a similar approach we argue that there should exist similar patterns for a trip, like *Leisure_trip_pattern, Leisure_weekend_trip_pattern* (the one that is chosen by John in our example) or *Business_trip_pattern*.

Here we do not use goal decomposition to sub-goals but use a high-level complex task that is further decomposed into lower level subtasks.

Travel pattern translation to planning domain

In order to use traveling patterns in terms of goals in GSO, we must map the GSF approach (da Silva Santos et al., 2009) with travel domain definitions (Houda et al., 2010). We provide an example for this in a diagram below (Fig. 1).

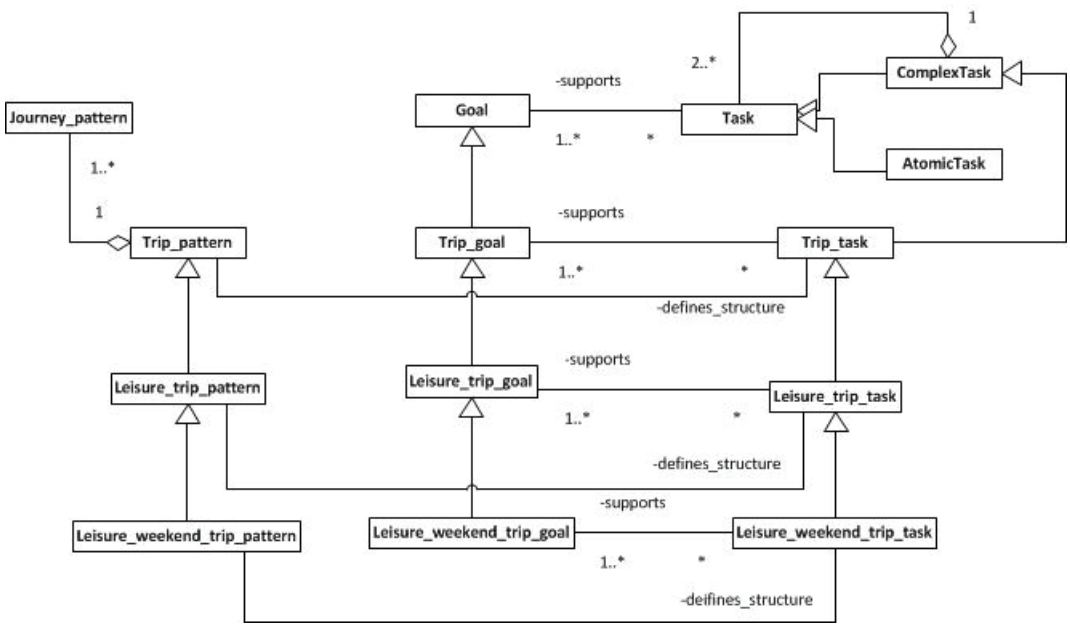
We envision that there exist different types of travel patterns. For a public transportation Houda et al. (2010) provide travel patterns like *Service journey pattern, Interesting journey pattern*, *Shopping journey pattern* taxonomy. We distinguish *journey* and *trip* concepts. *Trip* is an act of going to another place and returning back. *Journey* is one piece of travel, going from one place to another. So, the *leisure_weekend_trip* pattern is a composition of several *journeys.*

These patterns need to be defined and mapped with GSO before planning. The structure of the selected pattern can drive task decomposition to lower grained tasks which at the end will be mapped to web services.

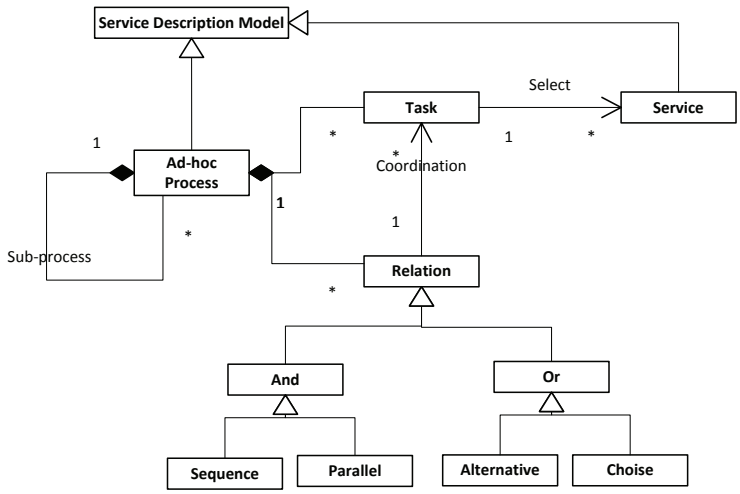
The entity *Journey_pattern* is from public transportation ontology (Houda et al., 2010). Entities *Goal, Task, CompexTask, AtomicTask* are from GSO. Others are our proposed entities (see Fig. 1).

Here we specify different types of goals (specializations of the goal in GSO) for each kind of pattern. Also, we suggest creating specializations for GSO Complex Task in order to achieve different types of goals. Using this approach a high-level goal (like "Have leisure weekend trip") can be translated to an AI planning problem definition.

Since we propose the usage of SHOP2 planner, the selected pattern description in OWL must be translated to a SHOP2 domain. SHOP2, like all HTN planners, uses a notion of an HTN method to capture how tasks are decomposed to subtasks. So an OWL pattern should be translated to the HTN method. Xiao et al. (2011) shows how ad-hoc processes could be created from ontology descriptions. An ad-hoc process is defined as a set of tasks that have to be per-



F i g. 1. ***Description of a travel goal domain based on GSO***



F i g. 2. ***An ad-hoc process model***

formed without strict order. The ad-hoc process model shows that it consists of simple flow control structure elements.

Sirin et al. (2004) shows how OWL-S process models can be encoded as SHOP2 domains. An OWL-S composite process uses the following control structures: Sequence, Unordered, Choice, If-Then-Else, Iterate, Repeat-Until, Repeat-While, Split and Split+Join. Therefore we conclude that any ad-hoc process can be transformed to an OWL-S composite process which, based on Sirin et al. (2004), can be translated to a SHOP2 planning domain. It is worth mentioning that SHOP2 cannot handle concurrency, therefore a solution how concurrent action must be handled should be presented.

Here we provide an idea how travel patterns can be defined and translated to an HTN planning domain. In order to prove this a more detailed research on this topic is required and is planned in the future.

Generating a plan for travel web service composition

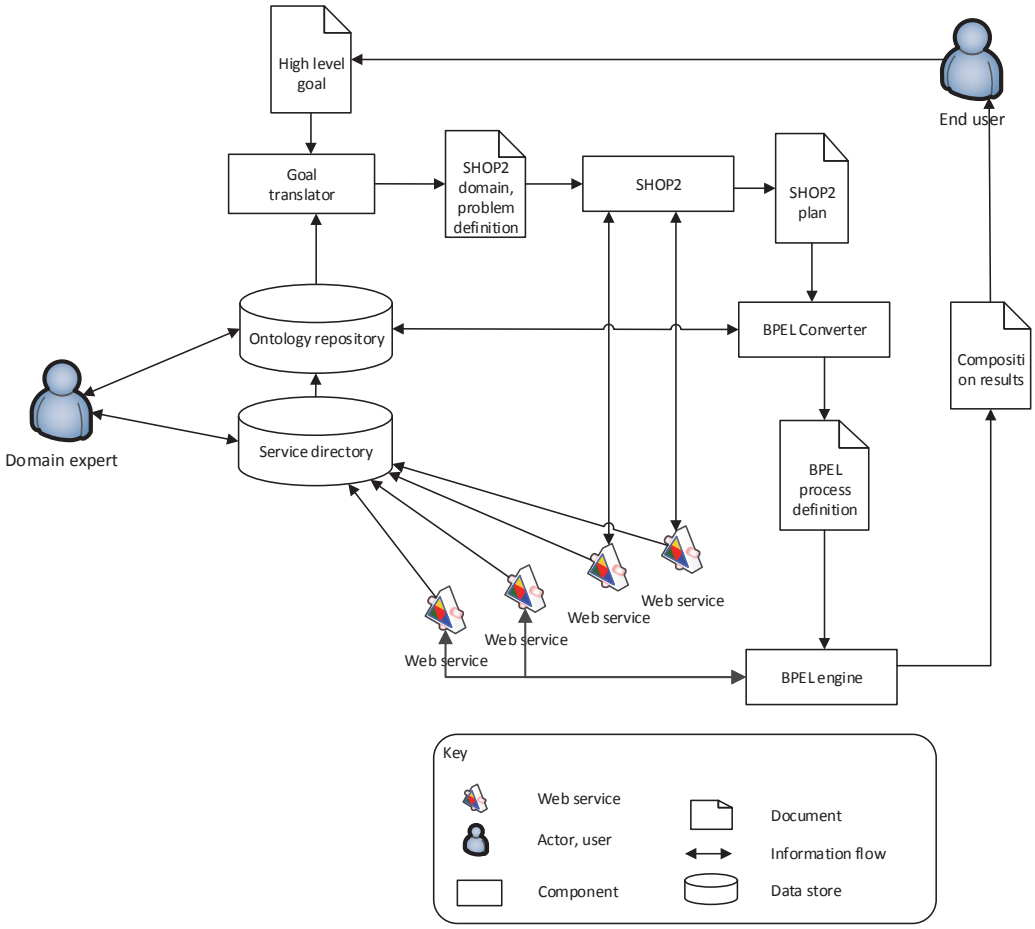
Having a planning domain and a problem provided a SHOP2 planner can search for a plan. Sirin et al. (2004) demonstrated that during the planning process information gathering actions could be executed.

In the example, during the planning travel information services must be requested online. The received information regarding flight schedules, available accommodation, points of interest must be used in travel planning. Based on that, a travel plan can be synthesized.

The plan contains world altering primitive actions – this represents calls to web services which state changes, for example, booking flight tickets.

We must transform the plan to executable workflow. The generated abstract plan must be converted to an executable description. Business process execution language (BPEL) is a good candidate for the composite, executable workflow as today there are many BPEL engines in the industry. Sirin et al. (2004) provides an example how an abstract plan (generated by SHOP2) can be transformed to the OWL-S process definition. Due to the similarities between OWL-S and BPEL we envision that a similar approach could be used for creating a BPEL composite workflow. GSO and a domain definition will be a primary source of knowledge for this process. We plan to test this approach in practice in the future.

We propose architecture as a structure of components where each component is either an already existing software component (SHOP2, BPEL Engine) or an integration component (Goal translator, BPEL Converter). One of the main reasons for this is decomposition of automatic web service composition problem to smaller problems that potentially already have solutions. Each sub-problem is solved by a specific component and the interchange between components is made using messages (documents). Data-flow between components and main actions is provided in Fig. 3.



F i g. 3. ***Conceptual solution of end-user driven service composition represented by data flow processes***

Goal translator is a component that by using domain ontology creates planning domain and problem definitions in a SHOP2 format from high-level end-user goals.

SHOP2 is an existing AI planner that can be extended with functions for an information sensing action execution (Nau et al., 2003). Information queries to external web services are made during planning. SHOP2 produces an abstract plan.

BPEL converter is a component that transforms an abstract AI plan to a web service composition in BPEL.

BPEL engine is a component that executes the process defined in BPEL. Currently there are plenty of BPEL engines in industry. Composition execution results are presented to an end-user.

Domain expert is an important role because the person is responsible for creating a domain definition in GSO that will be the core of the whole service composition process.

Conclusions

The main possibilities of the approaches for automatic and semi-automatic web service composition are analyzed paying more attention to end-user divining composition. The analysis shows that there is still a lack of methods and tools that focus on multi-complex possibilities of specifying end-user's requirements in a conve-

nient way. Most of approaches are related to the plan synthesis and the major part seeks to adopt an AI planning for the web service composition. However, there are approaches that try to solve a web service composition problem without AI planning methods.

Despite many papers published there still isn't a strict and clear classification of end-user driven composition approaches. We showed that defining categories of the template-based and AI planning-based is ambiguous.

Most of web service composition approaches propose template-based methods as the most feasible way of problem realization. We argue that an automatic composition when there is a domain knowledge specified by the ontology can be as practical as a template-based one.

The AI planning methods are suitable for end-user driven web service composition realization. Despite that a web context places additional constraints, requirements that classical AI planning cannot cope with these problems. Therefore, AI planning methods that allow relaxing of some of restrictive assumptions of a classical AI planning (example like partial state observability) are useful.

We provided an example that shows how this problem of the automatic service composition can be broken down to smaller ones, and how which can be handled separately. We provided a conceptual architecture of the system that is oriented to reuse existing assets in this research fields.

REFERENCES

AGARWAL, V., CHAFLE, G.; MITTAL, S.; SRIVASTAVA, B. (2008). Understanding approaches for web service composition and execution. In: *Proceedings of the 1st Bangalore Annual Computer Conference*. ACM, p. 1–8.

AKKIRAJU, R.; FARRELL, J.; MILLER, J.; NAGARAJAN, M.; SCHMIDT, M.; SHETH, A.; VERMA, K. (2005). Web service semantics – WSDL-S.

BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. (2001). The semantic web. *Scientific American*, 284, 5, p. 28–37.

CASATI, F.; SAYAL, M.; SHAN, M.C. (2001). Developing e-services for composing e-services. In: *Advanced Information Systems Engineering*. Springer Berlin Heidelberg, p. 171–186.

DA SILVA SANTOS, L. B.; GUIZZARDI, G.; GUIZZARDI, R. S. S.; DA SILVA, E. G.; PIRES, L. F.; van Sinderen, M. (2009). GSO: Designing a well-founded service ontology to support dynamic service discovery and composition. In: *Proceedings of the Enterprise Distributed Object Computing Conference Workshops, 2009. EDOCW 2009. 13th*. IEEE, p. 35–44.

DIGIAMPIETRI, L. A.; PÉREZ-ALCÁZAR, J. J.; MEDEIROS, C. B. (2007). AI Planning in Web Services Composition: a review of current approaches and a new solution. In: *VI ENIA-Proceedings of the XXVII Brazilian Computer Society Conference (CSBC2007)*.

DOMINGUE, J.; FENSEL, D.; DAVIES, J.; GONZÁLEZ-CABERO, R.; PEDRINACI, C. (2009). The Service Web: a Web of Billions of Services. *Towards the Future Internet*, p. 203.

DZEMYDIENĖ, D.; JASIŪNAS, E.; KRIAUČIUKAS, V.; MILIAUSKAS, A. (2011). Possibilities of managing of informational resources in the adaptive e-service providing system by using domain ontology. In: *Proceedings of the XV Computer Science Conference*. LIKS. Vilnius. Žara, p. 57–63.

GEEBELEN, K.; MICHIELS, S.; JOOSEN, W. (2008). Dynamic reconfiguration using template based web service composition. In: *Proceedings of the 3rd Workshop on Middleware for Service Oriented Computing*. ACM, p. 49–54.

GHALLAB, M.; NAU, D.; TRAVERSO, P. (2004). *Automated planning: theory and practice*. Morgan Kaufmann, ISBN 1-55860-856-7

HOUDA, M.; KHEMAJA, M.; OLIVEIRA, K.; ABED, M. (2010). A public transportation ontology to support user travel planning. In: *Proceedings of the Fourth International Conference on Research Challenges in Information Science (RCIS)*. IEEE, p. 127–136.

JIE, G.; BO, C., JUNLIANG, C.; LEI, Z. (2008). A template-based orchestration framework for hybrid services. In: *Proceedings of the Fourth Advanced International Conference on Telecommunications. (AICT'08)*. IEEE, p. 315–320.

KIM, A.; KANG, M.; MEADOWS, C.; IOUP, E.; SAMPLE, J. (2009). *A framework for automatic web service composition*. Naval Research Lab Washington DC Center For High Assurance Computing Systems (CHACS).

LÉCUÉ, F.; GORRONOGOITIA, Y.; GONZALEZ, R.; RADZIMSKI, M.; VILLA, M. (2010). SOA4All: an innovative integrated approach to services composition. In: *Proceedings of IEEE International Conference on Web Services (ICWS)*. IEEE, p. 58–67.

LI, X.; WU, C. (2009). Research on OWL-S Service Automatic Composition Based on Planning. In: *Proceedings of International Conference on Information Engineering and Computer Science. (ICIECS),* IEEE, p. 1–4.

MCILRAITH, S.; SON, T. C. (2002). Adapting Golog for composition of semantic Web services. In: *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*. Morgan Kaufmann Publishers; p. 482–496.

MEHANDJIEV, N.; LECUE, F.; WAJID, U.; NAMOUN, A. (2010). Assisted Service Composition for End Users. In: *Proceedings of the IEEE 8th European Conference on Web Services (ECOWS)*. IEEE, p. 131–138.

NAU, D. S.; AU, T. C.; ILGHAMI, O.; KUTER, U.; MURDOCK, J. W.; WU, D.; YAMAN, F. (2003). SHOP2: An HTN planning system. *Artif. Intell. Res. (JAIR)*, vol. 20, p. 379–404.

PISTORE, M.; TRAVERSO, P.; BERTOLI, P.; MARCONI, A. (2005). Automated synthesis of composite BPELWS web services. In: *Proceedings of the IEEE International Conference on Web Services, (ICWS)*. IEEE, p. 293–301.

PORTCHELVI, V.; VENKATESAN, V. PRASANNA; SHANMUGASUNDARAM, G. A. (2012). Study on composition goal in automated web services composition approaches. *International Journal of Computer Applications*, vol. 55, issue 11, p. 4–8.

RAO, J.; SU, X. (2005). A survey of automated web service composition methods. In: *Semantic Web Services and Web Process Composition*. Springer Berlin Heidelberg, p. 43–54.

REITER, R. (2001). *Knowledge in action: logical foundations for specifying and implementing dynamical systems*. Cambridge: MIT press, ISBN 0-26218218-1

SHADBOLT, N.; HALL, W.; BERNERS-LEE, T. (2006). The semantic web revisited. *Intelligent Systems, IEEE*, 21.3, p. 96–101.

SIRIN, E.; PARSIA, B.; WU, D.; HENDLER, J.; NAU, D. (2004). HTN planning for web service composition using SHOP2. *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 1, no 4, p. 377–396.

SIRIN, E.; PARSIA, B.; HENDLER, J. (2005). Template-based composition of semantic web services. In: *AAAI Fall Symposium on Agents and the Semantic Web,* p. 85–92.

SVÁTEK, V.; VACURA, M. (2005). Automatic Composition of Web Analysis Tools: Simulation on Classification Templates. In: *First International Workshop on Representation and Analysis of Web Space (RAWS-05)*. Online: http://CEUR-WS.org.

XIAO, H.; ZOU, Y.; TANG, R.; NG, J.; NIGUL, L. (2009). An automatic approach for ontology-driven service composition. In: *Proceedings of the IEEE International Conference on Service-Oriented Computing and Applications (SOCA' 2009)*. IEEE, p. 1–8.

VARTOTOJO VALDOMOS INTERNETINIŲ PASLAUGŲ AUTOMATINĖS IR PUSIAU AUTOMATINĖS KOMPOZICIJOS ATLIKIMO GALIMYBĖS

Dalė Dzemydienė, Arūnas Miliauskas

S a n t r a u k a

Šio mokslinio tyrimo tematika nagrinėja internetinių paslaugų kompozicijos atlikimo priemones ir būdus. Straipsnyje aprašomi vartotojo poreikiams pritaikytų internetinių paslaugų kompozicijos automatiniai ir pusiau automatiniai kūrimo būdai ir metodai, plačiau nagrinėjamos šių metodų galimybės ir apribojimai. Analizuojami moksliniuose straipsniuose pateikiami internetinių paslaugų kompozicijos atlikimo metodai ir išskiriami trys pagrindiniai šių paslaugų kompozicijos būdai: darbų srautų modeliais grindžiamas, paslaugų šablonais grindžiamas ir automatinis paslaugų kompozicijos metodas. Tyrimo tikslas – pateikti pasiūlymą, kuris leistų automatiniu ar pusiau automatiniu būdu kurti internetinių paslaugų kompozicijas pagal vartotojų poreikius. Pusiau automatinio internetinių paslaugų komponavimo uždavinio sprendimo būdą iliustruoja sudėtingas kelionės planavimo pavyzdys. Internetinių paslaugų kompozicijai atlikti siūloma taikyti dirbtinio intelekto planavimo metodus. Pateikiama tokio uždavinio sprendimo koncepcija, kuri grindžiama fragmentiniais kituose projektuose gautais paslaugų komponavimo rezultatais ir bando sujungti visą internetinių paslaugų kompozicijos procesą: nuo vartotojo keliamų kompozicijos reikalavimų įvedimo iki tinkamo paslaugų kompozicijos rezultatų pateikimo.

Rekomendacinės sistemos algoritmų veikimo elektroninio knygyno duomenų bazėje analizė

Aurimas Rapečka

Vilniaus universiteto Matematikos ir informatikos instituto doktorantas
Vilnius University, Institute of Mathematics and Informatics, PhD student
Akademijos g. 4, LT-08663 Vilnius
El. paštas: Aurimas.Rapecka@mii.vu.lt

Virginijus Marcinkevičius

Vilniaus universiteto Matematikos ir informatikos instituto daktaras
Vilnius University, Institute of Mathematics and Informatics, PhD, Senior researcher
Akademijos g. 4, LT-08663 Vilnius
El. paštas: Virginijus.Marcinkevicius@mii.vu.lt

Gintautas Dzemyda

Vilniaus universiteto Matematikos ir informatikos instituto profesorius
Vilnius University, Institute of Mathematics and Informatics, Professor, Principal researcher
Akademijos g. 4, LT-08663 Vilnius
El. paštas: Gintautas.Dzemyda@mii.vu.lt

Straipsnis skiriamas rekomendacinių sistemų algoritmų veikimo konkrečioje elektroninės parduotuvės duomenų bazėje analizei. Analizės tikslas – pagal pasirinktus įverčius rasti rekomendacinių sistemų algoritmus, efektyviausiai veikiančius turimoje duomenų bazėje. Šiame straipsnyje palyginti nemokamos rekomendacinių sistemų programinės įrangos paketai, aprašytas su pasirinkta programine įranga atliktas rekomendavimo algoritmų efektyvumo turimoje duomenų bazėje eksperimentinis tyrimas siekiant nustatyti geriausiai ir prasčiausiai veikiančius algoritmus.

Įvadas

Daugėjant interneto vartotojų vis didesnį mastą įgauna ir prekyba internete. Įvairios elektroninės parduotuvės siūlo platų prekių asortimentą ir standartinis vartotojas čia dažnai susiduria su prekių pasirinkimo keblumais. Skirtingai nuo parduotuvių realiame pasaulyje, kuriose daiktą galima paimti į rankas ar net išbandyti, elektroninės interneto parduotuvės pirkėjas turi mažiau galimybių pats įvertinti siūlomus produktus.

Šiame straipsnyje analizuojamas pavyzdys – elektroninė knygų parduotuvė, turinti tūkstančius knygų, kurių pavartyti fiziškai neįmanoma, todėl neužtenka surūšiuoti pagal žanrą, autorių, kainą ar naujumą. Čia reikalinga rekomendacinė sistema (RS), kurios algoritmai gebėtų greitai padėti potencialiems klientams susigaudyti didelėje esamų ir naujų knygų pasiūloje ir pasiūlyti tik konkrečiam vartotojui tinkamiausias knygas.

Nors egzistuoja daug rekomendacinėse sistemose naudojamų algoritmų (Vozalis, Margaritis, 2003), tačiau mokslinėje literatūroje aprašomuose tyrimuose dažniausiai dėmesys telkiamas į vieną ar du (siūlomą ir palyginimo) algoritmus (Onuma, Tong, Faloutsos, 2009; Gong, 2010). Šiame straipsnyje žvelgiama plačiau – eksperimentiškai vertinamas 18 rekomendavimo algoritmų veikimas konkrečioje duomenų bazėje. Vienas iš rekomendavimo algoritmų efektyvumo palyginimo pavyzdžių galėtų būti 2006–2009 metais internetinėje erdvėje vykęs

Netflix Prize konkursas (jo pagrindinis prizas 1 mln. JAV dolerių), kuriame siekta sukurti efektyviausiai *Netflix* filmų duomenų bazėje veikiantį rekomendavimo algoritmą. Konkursas sulaukė apie 50 000 dalyvių (Netflix, 2009), o nugalėtoju tapo algoritmas, plačiau aptartas šaltinyje (Toscher, Jahrer, Bell, 2009).

Šio straipsnio tikslas – atlikti elektroninės parduotuvės pirkimų įrašų duomenų bazės analizę, pagal iš anksto parinktus kriterijus atlikti atvirojo kodo rekomendacinių sistemų lyginamąją analizę ir eksperimentiškai įvertinti esamų rekomendavimo algoritmų efektyvumą konkrečios duomenų bazės atveju.

Rekomendacinės sistemos ir jų naudojami algoritmai

Nors yra sukurta įvairių rekomendacinių sistemų, tačiau išskiriami du labiausiai paplitę rekomendacinių sistemų tipai: kuriančios prognozes rekomendacinės sistemos ir generuojančios rekomendacijas rekomendacinės sistemos (Ricci, Rokach, Shapira & Kantor, 2010).

Kuriančių prognozes rekomendacinių sistemų pateikiama prognozė išreiškiama skaitiniu pavidalu r_{ij}, kuris reiškia prognozuojamą vartotojo a_i įvertinimą produktui b_j. Kuriant prognozes vartotojams reikalinga įverčių produktų aibė. Dažniausiai šiuos įverčius vartotojai palieka sąmoningai – atitinkamu balu įvertindami patikusius ar nepatikusius produktus.

Rekomendacijas kuriančių RS generuojama rekomendacija išreiškiama kaip vartotojui a_i tinkamiausių produktų sąrašas N_i. Kuriant prognozes užtenka turėti dvejetainę vartotojų įvertintų produktų įverčių aibę. Šiuos įverčius vartotojai palieka nesąmoningai, pavyzdžiui, pirkdami produktus.

Straipsnyje aprašomi eksperimentai atliekami su turima dvejetainių įverčių duomenų baze, todėl aptariami tik su rekomendacijų generavimu susiję aspektai. Šiuo metu yra sukurta apie 20 labiau paplitusių rekomendacijų generavimo algoritmų.

Vienas paprasčiausių rekomendavimo algoritmų – populiariausių prekių (angl. *MostPopular*) rekomendavimas visiems pirkėjams (Huang, Zeng, Chen, 2007). Tai labai paplitęs ir dažnai praktikoje taikomas rekomendavimo algoritmas. Prekės reitingas apskaičiuojamas paprasta formule:

$$Reitingas = \frac{Vartotojų,\ pirkusių\ prekę,\ skaičius}{Vartotojų\ duomenų\ rinkinyje\ skaičius}$$

Pirmiausia rekomenduojami tie produktai, kurių reitingas didžiausias. Šis algoritmas yra labai konservatyvus dėl to, kad rekomendacijos neišeina už erdvės, su kuria vartotojas jau susipažinęs, ribos (McFee et al., 2009).

Kitas populiarus algoritmas – *k* artimiausių kaimynų (angl. *KNN*, *UserKNN*) algoritmas. Tai klasikinis rekomendacinių sistemų algoritmas. Reitingai arba prekės prognozuojamos remiantis *k* panašiausių vartotojų praeities įverčiais.

Du esminiai šio algoritmo, realizuoto rekomendacinėse sistemose, veikimo aspektai – panašumo koeficiento skaičiavimas ir panašumo slenksčio nustatymas. Populiariausios vartotojo panašumo metrikos yra Pearsono koreliacija (angl. *Pearson correlation)* (Rashid, 2006) (1) ir kosinusinis panašumas (*angl. cosine similarity*) (Ziegler, 2013) (2).

$$panašumas(g,r) = \frac{\sum_p \left(g(p)-\bar{g}\right)\left(r(p)-\bar{r}\right)}{\sqrt{\sum_p \left(g(p)-\bar{g}\right)^2 \sum_p \left(r(p)-\bar{r}\right)^2}} \quad (1)$$

$$panašumas(g,r) = \frac{\sum_p g(p)r(p)}{\sqrt{\sum_p g(p)^2} \times \sqrt{\sum_p r(p)^2}} \quad (2)$$

Formulėse (1) ir (2) g – vartotojas, gaunantis rekomendaciją, r – vartotojas, kurio panašumas tikrinamas. Sumos ženklas formulėje rodo, kad panašumas skaičiuojamas naudojant visus produktus p, kuriuos abu vartotojai yra įvertinę. $g(p)$ ir $r(p)$ yra vartotojų įverčiai konkrečiam produktui p.

Formulėje (1) esantys kintamieji $\bar{g}$ ir $\bar{r}$ – atitinkamai g ir r vartotojų suteiktų įverčių jau vertintiems produktams vidurkiai. Formulės (1) skaitiklis yra vartotojų bendrų įvertintų produk-

tų įverčių nuokrypio nuo įverčių vidurkio sandaugų suma, o formulės vardiklis normalizuoja gautąjį rezultatą, kad jis priklausytų intervalui [-1;1]. Kuo rezultatas artimesnis 0, tuo vartotojai mažiau panašūs.

Savo ruožtu (2) formulės skaitiklis yra vartotojų bendrų įvertintų produktų įverčių sandaugų suma, o formulės vardiklis normalizuoja gautąjį rezultatą tam, kad jis priklausytų intervalui [0;1]. Kuo šis rezultatas artimesnis 0, tuo vartotojai mažiau panašūs, ir atvirkščiai.

KNN tipo algoritmai remiasi kiekvieno vartotojo panašumo koeficientų skaičiavimu, todėl esant didesniam vartotojų ir produktų įverčių skaičiui, šie algoritmai reikalauja didelių skaičiavimo ir laiko išteklių.

Kaip jau minėta, rekomendacinėse sistemose naudojamų algoritmų yra ir daugiau. Eksperimentinėje šio straipsnio dalyje atlikti eksperimentai su *Random* (atsitiktinis), *Zero* (visuomet grąžinantis 0 reitingą), *MostPopular* (populiariausių produktų rekomendavimo), *MostPopularByAttributes* (populiariausių kategorijose produktų rekomendavimo), *BPRMF* (Rendle et al., 2009), *BPRLinear* (Gantner et al., 2011), *LeastSquareSLIM* (Ning, Karypis, 2011), *ItemAttributeSVM* (Min, Han, 2005), *SoftMarginRankingMF* (Rendle, 2011), *WRMF* (Pan et al., 2008), *WeightedBPRMF* (Gantner et al., 2009), *MultiCoreBPRMF* (Rendle et al., 2009), *CLiMF* (Shi et al., 2012), *BPRSLIM* (Ning, Karypis, 2011), *UserKNN*, *ItemKNN* ir *ItemAttributeKNN* (visi *KNN – k* artimiausių kaimynų pagrindu veikiantys).

Rekomendacinių sistemų efektyvumo įverčių matai

Norint įvertinti, kaip tiksliai rekomendacinė sistema nustato, ar vartotojas įsigis (įsigijo) produktą, naudojami rekomendacinės sistemos efektyvumo įverčiai.

Produktus rekomenduojanti rekomendacinė sistema veikia kaip klasifikatorius, kuris nurodo, ar konkreti prekė bus (nebus) pasirinkta pirkėjo (Rajaraman, Leskovec, Ullman, 2013). Čia egzistuoja dviejų klasių atskyrimo problema (angl. *binary classification*). Pirma klasė vadinama susijusių elementų klase, arba šiuo atveju pirkėjo įsigytos prekės. Antrai klasei priklauso prekės (elementai), kurių pirkėjas neįsigijo. Prieš apmokant klasifikatorių turima duomenų aibė padalijama į mokymo ir testavimo aibes. Tuomet klasifikatorius apmokomas turima mokymo aibe, siekiant teisingai nustatyti kiekvienos prekės (elemento) klasę atskiram vartotojui. Paskui jis testuojamas naudojant testavimo aibę. Dažnai klasifikatoriai neteisingai priskiria pirmai klasei antros klasės elementus, ir atvirkščiai, o šias klasifikavimo klaidas stengiamasi įvertinti įvairiais RS efektyvumo įverčiais.

Klasifikatoriaus vertinimo principas pateikiamas 1 lentelėje. Atlikus klasifikatoriaus testavimą, gauti rezultatai surašomi į šią lentelę ir pagal ją apskaičiuojami įvairūs įverčiai. Toliau pateikiami dažniausiai taikomi įverčiai.

1 lentelė. ***Klasifikatoriaus vertinimo principai***

		Iš tiesų priklauso klasei	
		I klasė (susijusių)	II klasė
Klasifikatoriaus priskirta klasė	I klasė	TP (angl. *true positive*) (teisingai priskirta)	FP (angl. *false positive*) (I tipo klaida, kai I klasė buvo priskirta II)
	II klasė	FN (angl. *false negative*) (II tipo klaida, kai I klasei priskiriami antros klasės elementai)	TN (angl. *true negative*) (teisingai priskirta II klasei)

Preciziškumas

Preciziškumas (angl. *precision*) yra klasifikatoriaus, atskiriančio dvi klases, klasifikavimo įvertis. Šis įvertis parodo santykį, kiek iš pirmai klasei priskirtų elementų iš tiesų priklauso pirmai klasei. Kuo preciziškumas didesnis, tuo daugiau pirmai klasei priskirtų elementų iš tikrųjų yra iš primos klasės, vadinasi, tuo tiksliau veikia RS.

$$Preciziškumas = \frac{Teisingai\ priskirta\ pirmai\ klasei}{Viso\ priskirta\ pirmai\ klasei} = \frac{TP}{TP + FP}$$

Jautrumas

Klasifikatoriaus jautrumas (angl. *sensitivity, hit rate recall*) rodo teisingai priskirtų pirmai klasei pirmos klasės elementų santykį su bendru pirmos klasės elementų skaičiumi. Kuo jautrumas didesnis, tuo daugiau pirmos klasės elementų teisingai priskirta pirmai klasei, tuo tiksliau veikia RS:

$$Jautrumas = \frac{Teisingai\ priskirta\ pirmai\ klasei}{Pirmos\ klasės\ elementų} = \frac{TP}{TP + FN}$$

Plotas po ROC kreive (AUC)

AUC (angl. *area under the ROC curve*) įvertis nusako eilės (eiliškumo) nustatymo (rangavimo) kokybę įvertinant plotą po *ROC* kreive. Didžiausia *AUC* parametro reikšmė yra 1, o jeigu eilė nustatoma neatsitiktinai, tuomet *AUC* > 0,5. *AUC* reikšmė, arba plotas po *ROC* kreive, yra lygus tikimybei, kad klasifikatorius atsitiktinai parinktą pirkėjo nupirktą produktą įvertins aukštesniu įverčiu negu atsitiktinai parinktą produktą, kurio jis neįsigijo.

ROC kreivė – tai grafikas, iliustruojantis klasifikatoriaus, atskiriančio dvi klases, jautrumo ir specifiškumo santykį. Specifiškumas rodo antros klasės elementų priskyrimą antrai klasei.

Vidutinių preciziškumų vidurkis (MAP)

Tuo atveju, kai turime pirmos klasės elementų aibę $q_j \in Q, q_j = \{b_1, b_2, ..., b_{mj}\}$ ir eilę pirmųjų RS sugrąžintų elementų R_{jk}, tuomet vidutinis preciziškumas (angl. *average precision*) apskaičiuojamas pagal formulę:

$$ap_j = \frac{1}{m_j}\sum_{k=1}^{m_j} Preciziškumas(R_{jk}).$$

Vidutinis preciziškumas rodo, kiek tiksliai rekomendacinė sistema nuspėja, kurių produktų pirkėjas įsigis. Vidutinis preciziškumas priklauso nuo elemento vietos R_{jk} eilėje. Vidutinis preciziškumas geometriškai yra interpretuojamas kaip plotas po preciziškumo ir jautrumo kreive.

Imant pirmus 5 arba 10 R_{jk} elementų apskaičiuojami preciziškumo ir jautrumo įverčiai parodo, kiek iš jų yra primos klasės lyginant su R_{jk} ilgiu ir visų pirmos klasės elementų skaičiumi (jie žymimi *prec@5*, *prec@10*, *recall@5*, *recall@10*).

Vidutinių preciziškumų vidurkis (angl. *mean average precision, MAP*) – tai klasifikatoriaus jautrumo lygių kokybinis įvertis (Manning, Raggavan, Schutze, 2009). Turint kiekvieno pirkėjo vidutinį preciziškumą, *MAP* galima apskaičiuoti pagal formulę:

$$MAP(Q) = \frac{1}{|Q|}\sum_{j=1}^{|Q|} ap_j =$$

$$= \frac{1}{|Q|}\sum_{j=1}^{|Q|}\frac{1}{m_j}\sum_{k=1}^{m_j} Preciziškumas(R_{jk}).$$

Iš kitų klasifikatoriaus vertinimo įverčių šis išsiskiria stabilumu ir geru mokėjimu atskirti rezultatus. Geometriškai *MAP* atitinka vidutinį plotą po preciziškumo kreive. Šis plotas priklauso intervalui [0;1]. Tačiau norit *MAP* įverčiu tiksliai įvertinti rekomendacinės sistemos efektyvumą, reikia parinkti didelę ir pakankamai diversifikuotą testavimo aibę.

Nors visi aprašyti įverčiai skaičiuojami skirtingai ir praktikoje jie yra beveik lygiaverčiai, tačiau įvairiose duomenų bazėse gali parodyti kardinaliai skirtingus rezultatus. Todėl siekiant rasti efektyviausiai konkrečioje duomenų bazėje veikiančius rekomendavimo algoritmus, jų veikimo rezultatus būtina vertinti ne pagal kurį nors atskirą įvertį, bet pagal šių įverčių visumą. Šiame straipsnyje apskaičiuoti kiekvieno klasifikatoriaus įverčiai ranguojami ir galutiniam klasifikatoriaus įvertinimui naudojamas gautų rangų vidurkis.

Nemokamos ir atvirojo kodo rekomendacinės sistemos (programinė įranga)

Šiame poskyryje pateikiama trumpa lyginamosios labiausiai paplitusių nemokamo arba atvirojo kodo RS programinės įrangos analizės suvestinė (2 lentelė).

Svarbiausi programinės įrangos vertinimo aspektai turimos duomenų bazės atžvilgiu – atnaujinimo data, realizuotų RS algoritmų skaičius bei sistemose esančios reitingų prognozavimo ir produktų rekomendavimo galimybės.

2 l e n t e l ė. ***Nemokamos ir atvirojo kodo RS programinės įrangos palyginimas***

RS	Atnaujinta	Progr. kalba	RS algoritmų	Realizuoti savo	Reitingų progn.	Produktų rekomend.
MyMediaLite[1]	2013 04	*C#*	>20	Taip	Taip	Taip
Apache Mahout[2]	2012 06	*Java*	3	Ne	Taip	Taip
GraphLab[3]	2012 05	*C++*	15	Taip	Taip	Taip
LensKit[4]	2012	*Java*	?	Taip	Taip	Taip
Waffles[5]	2013 04	*C++*	>5	Ne	Taip	Ne
easyrec[6]	2012 02	*Online*	1	Ne	Taip	Taip
RecLab[7]	2011 02	*Online*	1	Ne	Taip	Ne
Crab[8]	2011	*Python*	>5	Taip	Taip	Taip
recommenderlab[9]	2011 11	*C++*	4	Taip	Taip	Taip
Jellyfish[10]	2012 12	*Python*	1	Ne	Taip	Ne
OpenSlopeOne[11]	2010 06	*PHP*	1	Taip	Taip	Ne
AppRecommender[12]	2011	*Python*	>10	Ne	Taip	Taip

[1] *MyMediaLite* (http://mymedialite.net/)
[2] *Apache Mahout* (http://mahout.apache.org/)
[3] *GraphLab* collaborative filtering library (http://select.cs.cmu.edu/code/graphlab/pmf.html)
[4] *LensKit* (http://lenskit.grouplens.org/)
[5] *Waffles* (http://waffles.sourceforge.net/)
[6] *Easyrec* (http://easyrec.org/)
[7] *RecLab* (http://code.richrelevance.com/reclab-core/)
[8] *Crab* (http://muricoca.github.io/crab/)
[9] *Recommenderlab* (http://cran.r-project.org/web/packages/recommenderlab/index.html)
[10] *Jellyfish* (http://hazy.cs.wisc.edu/hazy/victor/download/)
[11] *OpenSlopeOne* (http://code.google.com/p/openslopeone/)
[12] *AppRecommender* (https://github.com/tassia/AppRecommender)

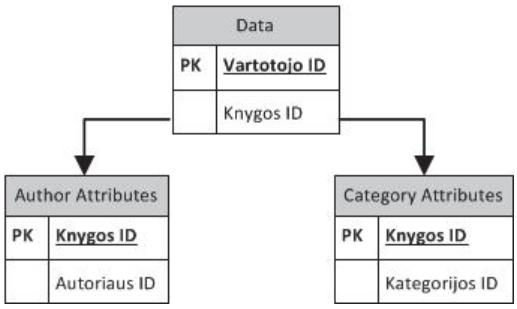
Iš 2 lentelės matyti, kad per paskutinį pusės metų laikotarpį vis dar buvo tobulinamos tik *MyMediaLite*, *Waffles* ir *Jellyfish* sistemos. Daugiausia sistemų sukurta naudojant *C++* ir *Python* programavimo kalbas. Daugiausia algoritmų realizuota *MyMediaLite* sistemoje, nuo jos nedaug atsilieka *GraphLab*. Pažymėtina, kad reitingų prognozavimas galimas visose nagrinėtose sistemose, o produktų rekomendacijos – tik didesnėje pusėje. Tiek algoritmų skaičiumi, tiek atnaujinimais ir kitomis galimybėmis *MyMediaLite* sistema lenkia kitas analizuotas. Todėl ji naudojama algoritmų efektyvumui įvertinti analizuojamoje duomenų aibėje.

Eksperimentinis tyrimas

Siekiant nustatyti, kurie rekomendavimo algoritmai veikia efektyviausiai, atlikti eksperimentai su realiais duomenimis. Paprastai rekomendavimo algoritmo efektyvumui įvertinti naudojami skirtingi įverčiai, nes visų algoritmo savybių ir jo priklausomybės nuo duomenų negalima įvertinti vienu matu. Norit nustatyti efektyviausią algoritmą pagal pasirinktus įverčius, buvo skaičiuojami įverčių rangai bei išvedamas kiekvienam algoritmui vidutinis įverčių rangas, ir kuo jis mažesnis, tuo algoritmo efektyvumas didesnis. Eksperimentiniams tyrimams buvo panaudoti Lietuvoje veikiančios el. knygų parduotuvės duomenys. Nuo el. parduotuvės atidarymo joje buvo siūloma įsigyti 19329 skirtingas knygas, iš kurių 12564 buvo bent kartą nupirktos. Pirkimo istorijoje užfiksuoti 14197 vartotojai, iš kurių 9930 įsigijo bent po vieną produktą. Taip pat buvo pateikta informacija apie 41177 įvykusių pardavimų (be pasikartojimų) ir jų datą. Dėl per mažo įvykusių pardavimų skaičiaus, palyginti su produktų ir vartotojų skaičiumi, atliktuose eksperimentuose pardavimo data nebuvo įvertinama. Tyrimams atlikti konstruojamas duomenų rinkinys, kurį sudaro poros $\{(Vartotojo\ ID, Knygos\ ID)\}$. Šis duomenų rinkinys žymimas *Data*.

Papildomai pateikti duomenų rinkiniai apie kiekvienai knygai priskirtas kategorijas ir autorius žymimi atitinkamai *Category_Attributes* ir *Author_Attributes*.

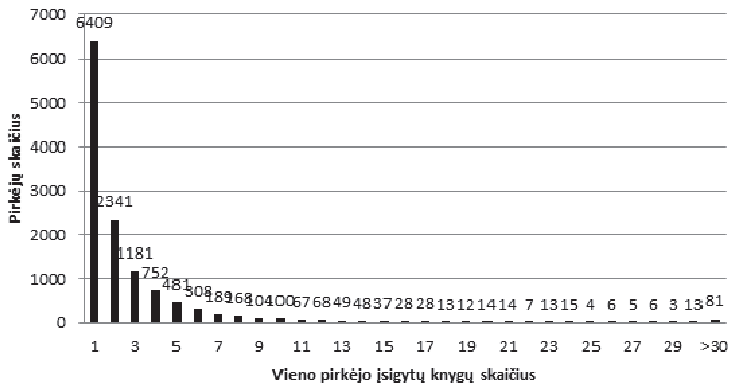
Duomenų rinkinių *Data*, *Category_Attributes* ir *Author_Attributes* struktūra bei tarpusavio ryšiai pateikiami 1 paveiksle.



1 p a v. ***Tyrimui naudotų duomenų rinkinių struktūra ir jų tarpusavio ryšiai***

Tolesniuose skaičiavimuose *Data* duomenų rinkinys transformuojamas į matricą, turinčias 9930 eilučių ir 12564 stulpelius (t. y. tiek, kiek vartotojų nupirko bent vieną knygą ir kiek buvo skirtingų knygų), matrica užpildoma vienetais ir nuliais. Vienetas reiškia, kad vartotojas *i* nupirko knygą *j*. Šios *Data* matricos užpildymas nenuliniais elementais (tankis) lygus $\frac{41177}{14197 \times 19329} = 0{,}015$, tačiau atmetus nė karto nenupirktas knygas bei nė vienos knygos nenupirkusius vartotojus, pasiekiamas $\frac{41177}{9930 \times 12564} = 0{,}033$ užpildymas.

Pakankamai svarbus duomenų rinkinio *Data* analizės aspektas yra pirkėjų pasiskirstymo pagal nupirktų knygų skaičių nustatymas. Toks pasiskirstymas iliustruojamas 2 pav. grafiku.



2 p a v. ***Pirkėjų pasiskirstymas pagal nupirktų knygų skaičių***

Kaip matoma iš pateiktojo grafiko, 50,68 pirkėjų įsigijo tik vieną knygą, o 88,29 sistemos vartotojų yra nupirkę mažiau nei 5 knygas. Esant tokiam vartotojų pasiskirstymui, prognozės nebūna labai tikslios dėl per mažo rekomendacinės sistemos turimo informacijos apie individualų vartotoją kiekio.

Duomenų rinkinyje pastebimos ir anomalijos – čia yra keli vartotojai, nupirkę daugiau nei 500 knygų. Šie vartotojai taip pat gali iškreipti rekomendacijų tikslumą.

Siekiant nustatyti geriausiai su turimais duomenimis veikiančius algoritmus, atlikta 17 rekomendacinėse sistemose dažniausiai taikomų prekių rekomendavimo algoritmų lyginamoji analizė. Visiems šiems algoritmams buvo atlikta po 10 eksperimentų, kai testavimo aibei atsitiktinai parenkami 10 proc. vartotojų, o mokymo aibėje paliekami likusieji. Eksperimentų rezultatų vidurkiai pateikiami 3 lentelėje ir 3 pav. Eksperimentams atlikti buvo naudojamas *DELL Optiplex* 790 kompiuteris.

Trečioje lentelėje *AUC*, *prec@5*, *prec@10*, *MAP*, *recall@5*, *recall@10* ir *NDCG* – rekomendacinių sistemų efektyvumo įverčiai. Mokymo laikas – laikas, kuris buvo reikalingas RS mokymui konkrečiame kompiuteryje, testavimo laikas – laikas, kurio reikėjo RS rezultatų testavimui ir patikrinimui atlikti.

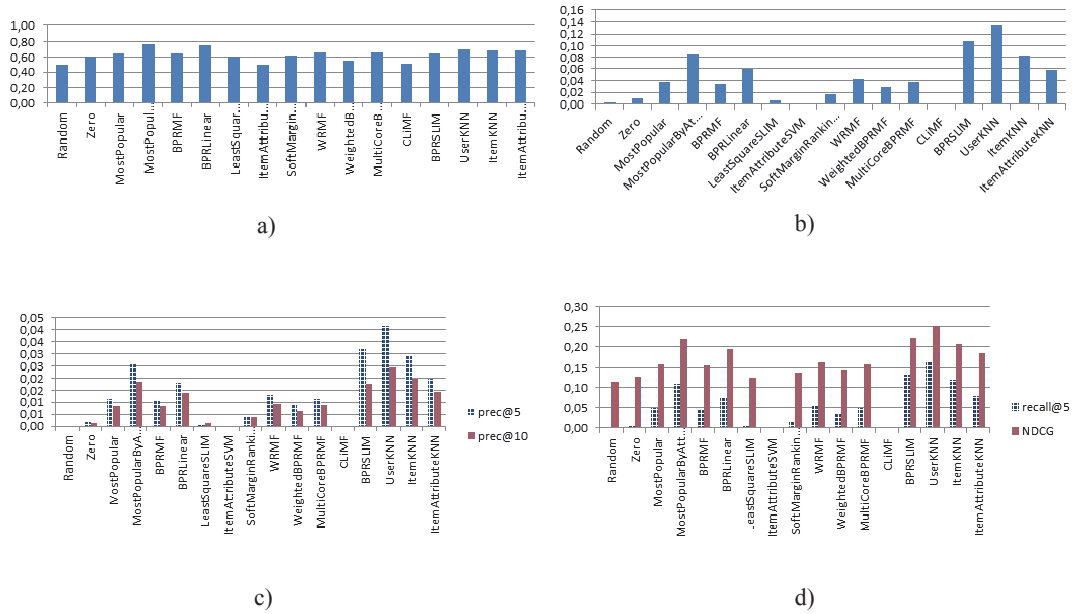
Iš 3 lentelėje ir 3 pav. pateiktų rezultatų galime išskirti turimame duomenų rinkinyje efektyviai ir neefektyviai veikiančius algoritmus. Geriausius rezultatus rodo *UserKNN*, *MostPopularByAttributes*, *BPRSLIM* ir *ItemKNN* algoritmai, tačiau pažymėtina, kad *UserKNN* ir *ItemKNN* algoritmai reikalauja didelių skaičiavimo išteklių.

Savo ruožtu neefektyviausių algoritmų grupei priklauso *LeastSquareSLIM*, *ItemAttributeSVM*, *CliMF*, *WeightedBPRMF* ir *SoftMarginRankingMF* algoritmai, kurie, veikdami šiame duomenų rinkinyje, pateikia prastus rezultatus.

3 l e n t e l ė. ***Rekomendacinės sistemos algoritmų efektyvumo palyginimas***

	RS algoritmų efektyvumo įverčiai						**Mokymo laikas, ms**	**Testavimo laikas, ms**
	AUC	***prec@5***	***prec@10***	***MAP***	***recall@5***	***NDCG***		
Random	0,50	0,00	0,00	0,00	0,00	0,11	0,00	176,91
Zero	0,61	0,00	0,00	0,01	0,01	0,13	0,00	152,21
MostPopular	0,65	0,01	0,01	0,04	0,05	0,16	0,10	173,21
MostPopularByAtt.	**0,77**	0,03	0,02	0,09	0,11	0,22	1,60	238,01
BPRMF	0,65	0,01	0,01	0,03	0,05	0,16	118,91	168,41
BPRLinear	0,76	0,02	0,01	0,06	0,08	0,19	1639,99	201,21
LeastSquareSLIM	0,61	0,00	0,00	0,01	0,00	0,12	244112,00	671,54
ItemAttributeSVM	0,50	0,00	0,00	0,00	0,00	0,00	1032,26	75,00
SoftMarginRankingMF	0,62	0,00	0,00	0,02	0,02	0,14	114,51	184,21
WRMF	0,66	0,01	0,01	0,04	0,05	0,16	524,73	187,81
WeightedBPRMF	0,55	0,01	0,01	0,03	0,04	0,15	108,21	167,91
MultiCoreBPRMF	0,65	0,01	0,01	0,04	0,05	0,16	29,90	209,41
CLiMF	0,50	0,00	0,00	0,00	0,00	0,00	7808,05	29,50
BPRSLIM	0,64	0,03	0,02	0,11	0,13	0,22	4333,45	278,52
UserKNN	0,70	**0,04**	**0,02**	**0,13**	**0,16**	**0,25**	49674,44	2233,73
ItemKNN	0,68	0,03	0,02	0,08	0,12	0,21	125422,70	3787,32
ItemAttributeKNN	0,69	0,02	0,01	0,06	0,08	0,19	129198,60	5252,30

Žymėjimai:	*Random* atskaitos taškas	*Zero* atskaitos taškas	Geras rezultatas	Blogas rezultatas



3 p a v. ***Rekomendacinės sistemos algoritmų efektyvumo palyginimas: a) pagal AUC įvertį; b) pagal MAP įvertį; c) pagal prec@5 ir prec@10 įverčius; d) pagal recall@5 ir NDCG įverčius***

4 l e n t e l ė. ***Kryžminio patvirtinimo eksperimentų rezultatas (k=5)***

Algoritmas	*AUC*	Vieta	*prec@5*	Vieta	*MAP*	Vieta	<…>**	Vidurk.	Galut. vieta
Random	0,499	18	0,001	17	0,004	17		13,18	**17**
Zero	0,552	15	0,001	16	0,008	16		12,27	**16**
MostPopular	0,719	3	0,017	10	0,051	10		7,18	**9**
*MostPopByAtt(C)**	0,781	1	0,032	5	0,100	5		5,27	**3**
BPRMF	0,711	5	0,016	13	0,050	12		9,82	**12**
BPRLinear	0,716	4	0,017	11	0,054	8		9,36	**8**
SoftMarginRankMF	0,650	9	0,005	15	0,021	15		12,18	**14**
WRMF	0,654	8	0,018	8	0,054	9		9,09	**10**
WeightedBPRMF	0,506	16	0,009	14	0,026	14		12,45	**15**
MultiCoreBPRMF	0,722	2	0,017	12	0,051	11		8,73	**11**
CLiMF	0,501	17	0,000	18	0,001	18		15,27	**18**
BPRSLIM	0,635	11	0,038	4	0,120	3		7,64	**4**
UserKNN	0,677	7	0,048	1	0,149	1		5,27	**1**
ItemKNN	0,639	10	0,030	7	0,084	7		9,45	**7**
ItemAttributeKNN	0,633	12	0,017	9	0,047	13		13,00	**13**
*MostPopByAtt(A)**	0,700	6	0,030	6	0,085	6		6,91	**6**
BPRLinear	0,623	14	0,041	3	0,120	4		6,91	**5**
ItemAttributeKNN	0,624	13	0,043	2	0,123	2		7,00	**2**

* Požymiams naudojama kategorija (C) arba autorius (A).

** Taip pat įvertinami ir ranguojami *prec@10*, *recall@5*, *recall@10*, *NDCG* bei *MRR* įverčiai, mokymo ir testavimo laikas bei atminties poreikis. Apskaičiuojamas rangų vidurkis ir pagal jį išrenkami geriausi algoritmai.

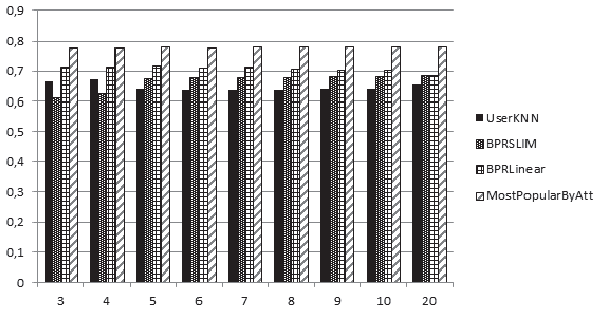
Likusieji algoritmai pateikia vidutiniškus, tačiau geresnius rezultatus nei *Random* ir *Zero* algoritmai, kurie laikomi visų algoritmų vertinimo atskaitos tašku.

Siekiant įvertinti algoritmų efektyvumą kintant vartotojo įsigytų knygų skaičiui, eksperimentai atlikti ir naudojant kryžminio patvirtinimo (angl. *cross_validation*) modelį algoritmui įvertinti. Kryžminis patikrinimo modelis sudaromas turimą duomenų rinkinį atsitiktinai sumaišius ir jį padalijus į k lygių dalių. Tuomet algoritmo mokymui nuosekliai imamos $k-1$, o likusi dalis naudojama testavimui. Toliau vertinimas tęsiamas k kartų imant kitas $k-1$ dalis mokymui ir vieną dalį testavimui.

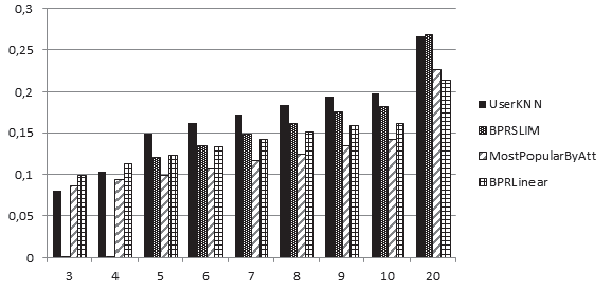
Rekomendacinių sistemų algoritmo rezultatų vertinimo eksperimentuose buvo taikomas šiek tiek modifikuotas kryžminio patvirtinimo modelis, t. y. iš viso duomenų rinkinio į mokymo aibę buvo perkeliami tų vartotojų duomenys, kurie pirko mažiau knygų nei k reikšmė. Likęs duomenų rinkinys sudalijamas į k lygių dalių. Tuomet viena suskirstyto rinkinio dalis atskiriama kaip testavimo aibė, o likusios priskiriamos mokymo aibei. Eksperimentai atliekami su visomis duomenų rinkinio kopijomis, nuosekliai keičiant dalį, laikomą testavimo aibe.

Mūsų atveju eksperimentai atlikti esant $k=3;4;5;6;7;8;9;10;20$ ir skaičiavimus atliekant su 18 potencialiai tinkamų algoritmų. Gauti rezultatai ranguojami ir tokiu būdu kiekvienam atvejui išrenkami geriausi metodai. Kryžminio patvirtinimo eksperimentų rezultatų dalis, kai $k=5$, pateikiama 4 lentelėje.

UserKNN, *BPRSLIM*, *BPRLinear* ir *MostPopularBy-Attributes* algoritmų *MAP* įverčio reikšmės augimas, priklausantis nuo k reikšmės, pateikiamas 4 pav. b grafike. Natūralu, kad didėjant vartotojo įsigytų knygų skaičiui algoritmai prognozuoja tiksliau.



a)



b)

4 p a v. ***AUC (a) ir MAP (b) įverčių kaita didėjant k reikšmei***

UserKNN, *BPRSLIM*, *BPRLinear* ir *MostPopularByAttributes* algoritmų *AUC* įverčio reikšmės kaita, priklausanti nuo k reikšmės, pateikiamas 4 pav. a grafike. Pažymėtina, kad šis RS įvertis dviem algoritmams kinta mažai (*MostPopularByAttributes* – paklaidų ribose, *BPRLinear* – fiksuojamas šioks toks sumažėjimas $k \in [10; 20]$ intervale), *BPRSLIM* – fiksuojamas didesnis reikšmės ūgtelėjimas $k \in [3; 5]$ intervale, o *UserKNN* įvairuoja $k \in [3; 5]$ intervale, vėliau nusistovi ties fiksuota reikšme ir vėl paauga $k \in [10; 20]$ intervale.

Tai, kad *MAP* ir *AUC* įverčiai tiems patiems algoritmams rodo skirtingus rezultatus, reiškia būtinumą ranguoti įverčius ir apskaičiuoti vidutinį algoritmą – tik apskaičiavus visus galimus įverčius, nustačius algoritmų rangus ir apskaičiavus vidurkį, galima optimaliai nustatyti algoritmo efektyvumą.

Atlikus visų įverčių tipų, skaičiavimų trukmės ir naudojamų atminties išteklių rangavimą, 5 lentelėje pateikiami geriausiai veikiantys algoritmai penkiais skirtingais kryžminės patikros atvejais.

Iš 5 lentelės matyti, kad *UserKNN* ir *MostPopularByAttributes (category)* algoritmai pirmauja nepriklausomai nuo vartotojo įsigytų knygų skaičiaus. Esant didesniam įsigytų knygų skaičiui, pakankamai efektyvūs yra *BPRSLIM* ir *MostPopularByAttributes*, o naujiems vartotojams, neturintiems ilgos pirkimų istorijos, efektyvūs yra ir *MostPopular* bei *BPRLinear (author)* algoritmai.

Iš analizės rezultatų pastebėtas gana įdomus dalykas: vienas algoritmas (*CLiMF*) šiame duomenų rinkinyje veikia prasčiau nei atsitiktinių pasiūlymų algoritmas (*Random*). Tikėtina, kad šis algoritmas efektyvus tik labai specifiniuose duomenų rinkiniuose. Be to, du algoritmai (*ItemAttributeKNN* ir *WeightedBPRM*) intervale $k \in [3; 8]$ gauna prastesnius rezultatus nei dar vienas vertinimo atskaitos taškas – *Zero* algoritmas. Kai $k = 10$, prastesnius nei *Zero* algoritmas rezultatus demonstruoja ir *SoftMarginRankingMF.* Esant dideliam pirktų knygų skaičiui ($k \approx 20$), visi tirti algoritmai (išskyrus *CLiMF*) demonstruoja geresnius rezultatus už *Random* ir *Zero* atskaitos taškus.

5 l e n t e l ė. ***Geriausiai veikiantys algoritmai skirtingų k reikšmių atveju***

k	3	5	8	10	20
1	*MPopByAtt (c)*	*UserKNN*	*UserKNN*	*UserKNN*	*MPopByAtt (c)*
2	*UserKNN*	*MPopByAtt (c)*	*MPopByAtt (c)*	*MPopByAtt (c)*	*UserKNN*
3	*MostPopular*	*MPopByAtt (a)*	*BPRSLIM*	*BPRSLIM*	*BPRSLIM*
4	*BRPLinear (a)*	*BRPLinear (a)*	*MPopByAtt (a)*	*MPopByAtt (a)*	*MPopByAtt (a)*
5	*MPopByAtt (a)*	*ItemAttributeKNN*	*BRPLinear (a)*	*ItemKNN*	*ItemKNN*

Išvados

Šiame straipsnyje atlikta elektroninės parduotuvės pirkimo įrašų duomenų bazės analizė, pagal iš anksto parinktus kriterijus atlikta atvirojo kodo rekomendacinių sistemų lyginamoji analizė ir eksperimentiškai įvertintas esamų rekomendavimo algoritmų efektyvumas konkrečios duomenų bazės atveju.

Atkreipiamas dėmesys, kad turimo duomenų rinkinio užpildymais nenuliniais elementais yra labai mažas (50,68 pirkėjų įsigijo tik vieną knygą, o 88,29 sistemos vartotojų yra nupirkę mažiau nei 5 knygas), todėl siūlomų algoritmų tikslumas nėra didelis. Kaip rodo eksperimentų suvestinė, algoritmų efektyvumas tiesiogiai priklauso nuo vartotojų įsigytų knygų skaičiaus, vadinasi, šiam skaičiui didėjant RS rekomenduoja tiksliau. Praktikoje šios metodikos veikimą reikėtų įvertinti pasiekus bent 1–3 proc. duomenų rinkinio užpildymą.

Naujiems arba neaktyviems vartotojams (pirkimų istorijoje turintiems tik 1 ar 2 nupirktų knygų įrašus) geriausius rezultatus parodė *MostPopular* ir *MostPopularByAttributes* algoritmai, kai atsižvelgiama ne vien į knygų, bet ir į knygų kategorijų bei autorių populiarumą visoje sistemoje.

Vartotojams, turintiems daugiau nei 3 knygas pirkimų istorijoje, geriausias rekomendacijas generavo *UserKNN* algoritmas, analizuojantis konkretaus vartotojo pirkimų istoriją ir ieškantis panašumų kitų vartotojų pirkimų istorijose. Tačiau reikėtų atkreipti dėmesį, kad šis metodas reikalauja daugiau kompiuterio operatyviosios atminties ir laiko išteklių skaičiavimams, todėl jis negali veikti realiu laiku.

Nustatyta, kad esant didesniam vartotojo pirktų knygų skaičiui (daugiau kaip 8), pakankamai gerus rezultatus demonstravo *BPRSLIM* algoritmas, kurio skaičiavimai vyksta daug greičiau. Šis algoritmas galėtų būti naudojamas kaip alternatyva didelių išteklių reikalaujančiam *UserKNN* algoritmui.

Atlikus lyginamąją rekomendacinių sistemų programinės įrangos analizę pastebėta, kad šiuo metu eksperimentams yra tinkamiausia *MyMediaLite* sistema.

LITERATŪRA

Netflix (2009). Oficialus projekto tinklalapis [interaktyvus] [žiūrėta 2013 m. gegužės 15 d.]. Prieiga per internetą: <http://www.netflixprize.com//leaderboard?showtest=f&limit=1000>.

GANTNER, Z.; DRUMOND, L.; FREUDENTHALER, C.; & SCHMIDT-THIEME, L. (2009). *Bayesian Personalized Ranking for Non-Uniformly Sampled Items*. JMLR: Workshop and Conference Proceedings, no. 18, p. 231–247.

GANTNER, Z.; DRUMOND, L.; FREUDENTHALER, C.; RENDLE, S.; & SCHMIDT-THIEME, L. (2011). Learning Attribute-to-Feature Mappings for Cold-Start Recommendations. In: *ICDM'10 proceedings of the 2010 IEEE International Conference on Data Mining*, p. 176–185.

GONG, S. (2010). A Collaborative Filtering Recommendation Algorithm Based on User Clustering and Item Clustering. *Journal of Software*, no. 7, p. 745–752.

HUANG, Z.; ZENG, D.; & CHEN, H. (2007). *A Comparative Study of Recommendation Algorithms in ECommerce* [interaktyvus] [žiūrėta 2013 m. gegužės 1 d.]. Prieiga per internetą: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.87.8138&rep=rep1&type=pdf>.

MANNING, C. D.; RAGGAVAN, P.; & SCHUTZE, H. (2009). *An Introduction to Information Retrieval.* Cambridge: Cambridge University Press. 482 p.

McFEE, B.; BERTIN-MAHIEUX, T.; ELLIS, D. P.; & LANCRIET, G. R. (2009). *The million song dataset challenge* [interaktyvus] [žiūrėta 2013 m. gegužės 15 d.]. Prieiga per internetą: <http://www.columbia.edu/~tb2332/Papers/admire12.pdf>.

MIN, S.-H.; & HAN, I. (2005). *Recommender Systems Using Support Vector Machines.* Web Engineering. Springer, p. 387–393.

NING, X., & KARYPIS, G. (2011). *Slim: Sparse linear methods for top-n recommender systems*. ICDM.

ONUMA, K.; TONG, H.; & FALOUSTOS, C. (2009). TANGENT: A Novel, „Surprise-me", Recommendation Algorithm. In: *KDD'09 proceedings*, p. 105–115.

PAN, R.; ZHOU, Y.; CAO, B.; LIU, N. N.; LUKOSE; R. M.; SCHOLZ, M. et al. (2008). *One-class collaborative filtering*. ICDM [interaktyvus] [žiūrėta 2013 m. gegužės 10 d.]. Prieiga per internetą: <http://www.rongpan.net/publications/pan-oneclasscf.pdf>.

RAJARAMAN, A.; LESKOVEC, J.; & ULLMAN, J. D. (2013). *Mining of Massive Datasets.* Cambridge University Press. 326 p.

RASHID, A. M. (2006). ClustKNN: *A highly scalable hybrid model & memory based algorithm* [interaktyvus] [žiūrėta 2013 m. gegužės 10 d.]. Prieiga per internetą: <http://glaros.dtc.umn.edu/gkhome/fetch/papers/clustKNN.pdf>.

RENDLE, S. (2011). Context-Aware Ranking with Factorization Models. New York: Springer. 180 p.

RENDLE, S.; FREUDENTHALER, C.; GANTNER, Z.; & SCHMIDT-THIEME, L. (2009). BPR: Bayesian Personalized Ranking from Implicit Feedback. In: *UAI'09 Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence,* p. 452–461.

RICCI, F.; ROKACH, L. SHAPIRA, B.; & KANTOR, P. B. (2010). *Recommender Systems Handbook.* New York: Springer. 842 p.

SHI, Y.; KARATZOGLOU, A.; BALTRUNAS, L.; LARSON, M.; OLIVIER, N.; & HANJALIC, A. (2012). CLiMF: Learning to Maximize Reciprocal Rank with Collaborative Less-is-More Filtering. In: *RecSys'12 Proceedings of the sixth ACM conference on Recommender systems*, p. 139–146.

TOSCHER, A.; JAHRER, M.; & BELL, R. M. (2009). *The BigChaos Solution to the NetFlix Grand Prize* [interaktyvus] [žiūrėta 2013 m. gegužės 1 d.]. Prieiga per internetą: <http://www.netflixprize.com/assets/GrandPrize2009_BPC_BigChaos.pdf>.

VOZALISs, E.; & MARGARITIS, K. G. (2003). *Analysis of recommender systems algorithms* [interaktyvus] [žiūrėta 2013 m. gegužės 1 d.]. Prieiga per internetą: <http://lsa-svd-application-for-analysis.googlecode.com/svn-history/r72/trunk/LSA/Other/LsaToRead/hercma2003.pdf>.

ZIEGLER, C. N. (2013). *Social Web Artifacts for Boosting Recommenders.* New York: Springer. 187 p.

ANALYSIS OF THE EFFICIENCY OF RECOMMENDATORY SYSTEMS ALGORITHMS IN AN E-BOOKSHOP

Aurimas Rapečka, Virginijus Marcinkevičius, Gintautas Dzemyda

S u m m a r y

In the paper, the efficiency of various recommendatory systems algorithms in a data set of the local e-bookshop is analysed. The key goal of analysis is to determine effective and not effective algorithms in the data set used for analysis. An analytical review of free or open source software of ecommendatory systems is presented. Some comparison criteria are selected. According to the criteria, a comparative analysis of the popular software of ecommendatory systems is made and some experiments with the best evaluated software are done. We have determined here which algorithms are effective in the data set, used for the experiments.

PROGRAMAVIMAS IR PROGRAMINĖ ĮRANGA

OpenGL ir DirectX technologijų spartos palyginimas programuojant trimačius vaizdus

Ernestas Filatovas

Vilniaus Gedimino technikos universiteto
Fundamentinių mokslų fakulteto
lektorius, daktaras
Vilnius Gediminas Technical University,
Faculty of Fundamental Sciences,
Lecturer, Doctor
Saulėtekio al. 11–615, LT-10223, Vilnius
El. paštas: Ernest.Filatov@gmail.com

Grigorijus Melničenko

Lietuvos edukologijos universiteto
Gamtos, matematikos ir technologijų fakulteto
docentas, daktaras
Lithuanian University of Educational Sciences,
Faculty of Science and Technology,
Assoc. Professor, Doctor
Studentų g. 39, LT-08106, Vilnius
El. paštas: gmelnicenko@gmail.com

Vytautas Jakštys

Lietuvos edukologijos universiteto Gamtos, matematikos ir technologijų fakulteto magistras
Lithuanian University of Educational Sciences, Faculty of Science and Technology, Master
Studentų g. 39, LT-08106, Vilnius
El. paštas: vitaakas@gmail.com

Pagrindiniai trimačių vaizdų apdorojimo konkurentai yra DirectX ir OpenGL technologijos. Straipsnyje pateikiama šių technologijų lyginamoji analizė. Eksperimentiškai tiriama DirectX ir OpenGL technologijų veikimo sparta. Eksperimentams skirta testinė programinė įranga sukurta populiariausiomis programavimo kalbomis C++ ir C#. Darbe pateikiami ir analizuojami tyrimų rezultatai, gauti atlikus kelis skirtingus spartos testus naudojant sukurtą programinę įrangą.

Įvadas

Grafinis duomenų apdorojimas ir vaizdavimas plačiai naudojamas daugelyje sričių – moksle, versle, inžinerijoje, mene, reklamoje, laisvalaikio poreikiams ir kt. Modernus pasaulis jau neįsivaizduojamas be trimatės kompiuterinės grafikos. Su trimate grafika susiduriama tokiose srityse, kaip realaus laiko modeliavimo sistemos, kompiuteriniai žaidimai, filmai. Čia trimatės grafinės informacijos kokybė ir atnaujinimo sparta yra vienas svarbiausių parametrų, lemiančių jos naudojimo efektyvumą, patogumą, patrauklumą ir populiarumą.

Daugelį metų programuotojų bendruomenėje diskutuojama apie praktikoje naudojamų trimačių vaizdų apdorojimo technologijų privalumus ir trukumus. Pagrindiniai konkurentai trimačių vaizdų programavime srityje yra bibliotekų rinkiniai *DirectX* ir *OpenGL*. Vienareikšmiškai įvardinti, kuri iš šių technologijų yra geresnė, labai sudėtinga.

Minėtos technologijos dažniausiai naudojamos kuriant programinę įrangą C++ ir C# progra-

mavimo kalbomis. Mokslinėje literatūroje šiuolaikinių *OpenGL* ir *DirectX* versijų greitaveikos eksperimentinio lyginimo rezultatų neskelbiama. Interneto tinklapiuose ir forumuose nemažai rašoma apie konkrečios technologijos pranašumus, arba trūkumus, tačiau dauguma iš pasisakymų yra subjektyvus ir neparemti eksperimentiniais tyrimais. O keli internete skelbiami *DirectX* ir *OpenGL* technologijų veikimo spartos eksperimentinių tyrimų rezultatai (*OpenGL*, 2013; Sebastian, 2012) nėra patikimi, nes eksperimentai yra neišsamus (atlikti tik vienu kompiuteriu, tik viena operacine sistema, per mažai eksperimentų). Todėl šiuo darbu siekiama atlikti *DirectX* ir *OpenGL* technologijų lyginamąją analizę, o taipogi eksperimentiškai ištirti jų greitaveiką. *DirectX* ir *OpenGL* technologijos nagrinėjamos C++ ir C# programavimo kalbų pagrindu.

Trimačių vaizdų programavimo technologijos

Konkuruoja dvi trimačių vaizdų (trimatės grafikos) programavimo technologijos – *DirectX* ir *OpenGL*.

OpenGL (angl. *Open Graphics Library*) – tai grafinės kompiuterio techninės įrangos programinė sąsaja (bibliotekų rinkinys). Bibliotekos susideda iš funkcijų, leidžiančių programuotojui atvaizduoti objektus vaizduoklyje, atlikti įvairias operacijas su jais, kurti aukštos kokybės grafinius vaizdus ir tekstūras, naudojamas objektams spalvinti (*OpenGL*, 2013; Shreiner et al., 2013).

OpenGL bibliotekas būtų galima apibūdinti kaip komandas, leidžiančias aprašyti dvimačius ar trimačius objektus, bei komandų rinkinį, kuris nustato, kaip objektas vaizduojamas ekrane, t. y. įkeliamas į ekrano buferį (angl. *Framebuffer*), saugomą darbinėje atmintyje. *OpenGL* gali veikti dviem režimais: tiesioginiu (angl. *Immediate mode*) ir netiesioginiu (angl. *Retained mode*). Pirmu atveju grafinius objektus formuojantys primityvai (primityvios figūros, pvz., trikampiai) vizualizuojami ekrane tiesiogiai ir nenaudojama papildoma vaizdo plokštės atmintis primityvams saugoti (informacija apie primityvus prarandama juos vizualizavus). Antru atveju, programos vykdymo metu, primityvios figūros išsaugomos specialiame sąraše, kuris saugomas vaizdo plokštės atmintinėje. Tokiu būdu primityvai gali būti atvaizduoti ekrane kelis kartus ir panaudoti keliems skirtingiems objektams piešti.

Dalis *OpenGL* funkcijų yra skirtos elementariausiems geometriniams objektams (taškams, atkarpoms, daugiakampiams) piešti. Kita dalis – objektų atvaizdavimui ekrane, įskaitant apšvietimą, spalvinimą, išsidėstymą, atvaizdavimą trimatėje koordinačių sistemoje ir kt. Taip pat yra numatytos funkcijos, leidžiančios tiesiogiai kreiptis į ekrano buferį, t. y. galima skaityti ir rašyti į atskirus ekrano pikselius. Tiek tekstas, tiek vaizdai vaizduoklio ekrane konstruojami iš taškų (pikselių) matricos.

OpenGL atlieka tik skaitymą iš ekrano buferio ir rašymą į jį. *OpenGL* nepalaiko ir aukšto lygio komandų, aprašančių trimačius objektus. Tokios komandos leistų atvaizduoti sudėtingus geometrinius objektus, pavyzdžiui, lėktuvus ar automobilius. *OpenGL* tik sudaro galimybes programuotojui visa tai sukurti pačiam iš nedidelio skaičiaus geometrinių primityvų: taškų, linijų ir daugiakampių.

OpenGL sudaro kelios plačiai naudojamos bibliotekos: *GL*, *GLU*, *WGL*, *GLX*, *GLUT*, *GLFW*, *GLM*, *WebGL* ir kt., kurių detalesnis aprašymas pateiktas šaltiniuose (*OpenGL*, 2013; Shreiner et al., 2013).

DirectX – tai bendras aplikacijų programavimo sąsajų (API) rinkinys, skirtas daugialypės terpės užduotims tvarkyti, ypač kurti tuos žaidimus, kurie vėliau bus žaidžiami naudojant *Microsoft* produktus, tokius kaip *Windows, Xbox ir Xbox 360* (*DirectX*, 2013; *DirectX* Graphics, 2013; Luna, 2008).

Nors *DirectX* bibliotekų rinkinys iš pradžių buvo skirtas žaidimų programavimui *Windows* operacinei sistemai, šiandien jis yra naudojamas ne tik žaidimams, bet vaizdą bei garsą apdorojančioms ir išvedančioms programoms, bei matematiniams skaičiavimams. Naujesnėse *Windows* operacinėse sistemose, tokiose kaip *Windows 7* ir *Windows 8*, *DirectX* tapo ir pagrindiniu operacinės sistemos grafinės informacijos atvaizdavimo varikliu.

Šiuolaikinės *DirectX* versijos objektus vizualizuoja tik tiesioginiu režimu.

DirectX yra platinamas dviem programų paketais:

- *Runtime* (skirtu naudotojams);
- SDK (skirtu programuotojams).

DirectX technologija susideda iš kelių skirtingų su COM (angl. *Component Object Model*) suderinamų komponentų (bibliotekų): *DirectX Graphics*, *DirectInput*, *DirectPlay*, *DirectSound*, *DirectSound3D*, *DirectMusic*, *DirectX Media Object*, *DirectX Media Objects*, kurių detalesnis aprašymas pateiktas šaltiniuose (Luna, 2008; Sherrod, Jones, 2011).

COM trumpai gali būti apibrėžtas kaip technologinis programinės įrangos kūrimo standartas, kuris remiasi tarpusavyje sąveikaujančiais komponentais. Programos, sukurtos remiantis COM standartu, iš tiesų nėra savarankiškos, o sudaro tarpusavyje sąveikaujančių COM komponentų rinkinį. Šie komponentai gali būti naudojami vienu metu skirtingomis aplikacijomis (COM, 2013; What is COM, 2013). „Microsoft“ šią technologiją pristatė 1993 m. COM technologija yra kitų *Microsoft* sukurtų ir plačiai naudojamų technologijų pagrindas, pavyzdžiui, *OLE, ActiveX, COM+, DCOM, DirectX, Windows Runtime* (Sherrod, Jones, 2011).

Kaip minėta, yra dvi pagrindinės technologijos trimatei grafikai programuoti: *DirectX* ir *OpenGL*. Rinkoje dažniau naudojama *Microsoft* sukurta technologija *DirectX*, ypač žaidimuose ir grafinėse sistemose, kuriose būtinas aparatinis spartinimas. *OpenGL* technologija plačiau naudojama CAD sistemose, vizualizacijai moksliniuose tyrimuose, skrydžių simuliacijai (Mažutis, 2009). Tačiau, renkantis trimačių vaizdų programavimo priemonę, reikia išskirti ir gerai išanalizuoti pagrindinius *DirectX* ir *OpenGL* technologijų pranašumus ir trūkumus.

Išanalizavus literatūros šaltinius (Comparison, 2013; Hill, Kelley, 2007; Mažutis, 2009; *OpenGL*..., 2013; *OpenGL*, 2013; Shreiner et al., 2013; Wright et al., 2010), pateikiamos pagrindinės *OpenGL* technologijos savybės.

Teigiamos savybės:

- Galima naudoti įvairiose operacinėse sistemose;
- Gerai išnaudoja pritaikytos aparatūros spartinimo galimybes;
- Turtinga funkcijų biblioteka;
- Palaiko daugiau nei vieną grafikos sąsają (angl. *Multiple Graphics Adapters*);
- Palaiko išplėtimą (angl. *Extentions*);
- Nemokama;
- Stabili.

Neigiamos savybės:

- Ne visos vaizdo plokštės palaiko aparatinį *OpenGL* spartinimą;
- Naudojimui būtinos papildomos bibliotekos;
- Versijos pasirinkimo problema (dėl produkto atvirumo yra išleistos kelios skirtingų kūrėjų *OpenGL* versijos);
- Kai kurios bibliotekos yra mokamos.

Išanalizavus literatūros šaltinius (Cai et al., 2011; Comparison, 2013; *DirectX*, 2013; *DirectX*..., 2013; Guo et al., 2012; Luna, 2006 ir 2008; Mulholland, Hakala, 2004; Sherrod, Jones, 2011) straipsnyje yra pateikiamos pagrindinės *DirectX* technologijos savybės.

Teigiamos savybės:

- Paprasta panaudoti;
- Labai gerai išnaudoja pritaikytos aparatūros spartinimo galimybes;
- Didžioji dauguma vaizdo plokščių palaiko aparatinį *DirectX* spartinimą;
- Integruota į *Microsoft* operacines sistemas;
- Yra daug patogių programavimo įrankių, nesudėtingai palaikančių *DirectX* bibliotekas;
- Labai daug dokumentacijos ir pavyzdžių;
- Platus naudojimas, kas verčia nuolat tobulinti *DirectX*;
- Palaiko išplėtimą;
- Palaiko daugiau nei vieną grafikos sąsają;
- Yra CAD (angl. *Computer-aided design*) sistemų, kurios turi failų formatą, tiesiogiai pritaikytą *DirectX* aplinkai;
- Nemokama;
- Stabili.

Neigiamos savybės:

- Palaiko tik vieną operacinių sistemų šeimą *Windows*;
- Kad būtų galima naudoti, reikalingos papildomos bibliotekos.

Svarbu pažymėti, kad *OpenGL* ir *DirectX* architektūros yra skirtingos: *OpenGL* technologija duomenims perduoti naudoja procedūrinį modelį, o *DirectX* technologija remiasi COM modeliu, kurio naudojimo sudėtingumą šiandien supaprastina *Microsoft* .NET Framework (Comparison, 2013; Guo et al., 2013; Mulholland, Hakala, 2004).

C++ ir C# programavimo kalbos programuojant trimačius vaizdus

Kaip minėta ankstesniame skyriuje, pagrindinės trimačių vaizdų programavimo technologijos yra *DirectX* ir *OpenGL*. Kad būtų galima naudoti šias technologijas, programuotojui reikia parinkti programavimo kalbą. Yra sukurta gausybė programavimo kalbų, tačiau populiariausios yra kelios. Kompanija TIOBE kas mėnesį skelbia populiariausių pasaulyje programavimo kalbų sąrašą. Pirmoje lentelėje pateikiama informacija pagal 2013 m. balandžio mėnesio duomenis (Tiobe..., 2013).

Taigi tarp populiariausių modernių programavimo kalbų, tinkamiausių trimačiams vaizdams programuoti, yra trys: C++, Objective-C ir C#. Objective-C yra pagrindinė programavimo kalba, skirta *OS X* ir *iOS* operacinėms sistemoms ir naudojama *Apple* kompanijos produktų programavimui (Kochan, 2011). Šiame straipsnyje apsistota ties C++ ir C# programavimo kalbomis, kurios gerai pritaikomos Windows operacinių sistemų šeimai ir jos grafikos programavimui. Abejose kalbose galima naudoti tiek *DirectX*, tiek *OpenGL* technologijas.

Norint programiškai valdyti (programuoti) trimatę vizualizaciją reikia prijungti atitinkamas *OpenGL* ir *DirectX* bibliotekas prie kuriamos programos kodo. Pastebėtina, kad, vizualizuojant tiek *OpenGL*, tiek *DirectX* technologijomis, C++ ir C# programavimo kalboms sukurtos ir naudojamos skirtingos bibliotekos. Kaip buvo

1 lentelė. ***Populiariausios programavimo kalbos 2013 m. balandžio mėn. duomenimis***

Pozicija	Programavimo kalba	Naudojimas
1	C	17,862 proc.
2	Java	17,681 proc.
3	C++	9,714 proc.
4	Objective-C	9,598 proc.
5	C#	6,150 proc.
6	PHP	5,428 proc.
7	(Visual) Basic	4,699 proc.
8	Python	4,442 proc.
9	Perl	2,335 proc.
10	Ruby	1,972 proc.

minėta ankstesniame skyrelyje, dėl kodo atvirumo yra išleistos kelios skirtingų kūrėjų *OpenGL* bibliotekos. Tyrimuose buvo naudojamos tokios populiariausios bibliotekos:

- C++ kalboje darbui su *OpenGL* technologija naudojama nemokama biblioteka glut32.dll (Freeglut, 2013), o su *DirectX* technologija – biblioteka d3dx11_43.dll (Download, 2013).
- C# kalboje darbui su *OpenGL* technologija naudojamos nemokamos bibliotekos Tao.*OpenGL*.dll, Tao.Paltform.Windows.dll, Tao.FreeGlut.dll (Mono, 2013), o su *DirectX* – Microsoft.*DirectX*.dll, Microsoft.*DirectX*.Direct3D.dll ir Microsoft.*DirectX*.Direct3DX.dll (Download, 2013).

Galimi ir kiti deriniai (taip pat mokamos *OpenGL* bibliotekos), tačiau eksperimentiniams tyrimams buvo pasirinktos populiariausių nemokamų bibliotekų naujausios versijos.

Eksperimentiniai tyrimai

Naudodamas *DirectX* ir *OpenGL* technologijas, programuotojas negali valdyti, kada vizualizaciją (angl. *Rendering*) atliks centrinis procesorius (CPU), o kada vaizdo plokštės procesorius (GPU). Kompiuterio išteklius bibliotekos paskirsto automatiškai.

Šiuolaikiniuose vaizduokliuose vaizdas atnaujinamas apie 60–120 kartų per sekundę (vaizdo atnaujinimo dažnis yra 60–120 Hz),

o priklausomai nuo CPU ir GPU galimybių ir trimačių vaizdų sudėtingumo, jis gali būti atnaujintas (perskaičiuotas) kelis tūkstančius kartų per sekundę. Taigi standartiškai *DirectX* ir *OpenGL* bibliotekose uždėti ribotuvai tam, kad trimačiai vaizdai būtų atnaujinami ne dažniau, nei vaizduoklis gali tai padaryti. Kuriant programinę įrangą eksperimentiniams tyrimams į tai yra atsižvelgta ir toliau darbe pateikiami tie rezultatai, kurie gauti (apskaičiuoti) kompiuterių procesoriais, o ne tie, kurie rodomi vaizduokliuose.

Trimačių vaizdų vizualizacijoje dažniausiai objektai sudaromi iš elementarių figūrų – trikampių (Zhang et al., 2012). Kuo sudėtingesnė figūra, tuo daugiau elementarių trikampių yra panaudota jai suformuoti, tuo daugiau skaičiavimo išteklių reikia vaizdų formavimui vaizduoklyje. Be to, reikia papildomų skaičiavimų išteklių reikia, kai, pavyzdžiui, žaidimuose naudojamas didelis vaizdo gilumas, kuo iš toliau matomas objektas žaidime, tuo didesnis vaizdo gilumas.

Eksperimentiškai buvo siekiama nustatyti *DirectX* ir *OpenGL* technologijų vizualizacijos spartą tokiais atvejais:

- Elementarios (trikampės) figūros judėjimas;
- Sudėtinės (sudarytos iš daugelio elementarių trikampių) figūros judėjimas;
- Figūrų, esančių skirtinguose gilumuose, judėjimas (vizualizuojamos figūros atitolinamos).

Programinė įranga, skirta eksperimentams, buvo sukurta naudojant *Microsoft Visual Studio* aplinką. Visomis sukurtomis programomis, skirtomis eksperimentiniams tyrimams, vizualizuojamos figūros buvo formuojamos standartiniame *Windows Form* lange. Jos buvo vaizduojamos vienodo dydžio, rodymo kampo, perspektyvos, tačiau dėl skirtingose bibliotekose naudojamų skirtingų funkcijų minėtų programų realizacijos skyrėsi.

Darbe eksperimentiniu būdu buvo siekiama nustatyti, kiek skiriasi *DirectX* ir *OpenGL* technologijų sparta, jomis vizualizuojant judančias elementarias (trikampes) ir sudėtines (sudarytas iš daugelio elementarių) figūras.

Buvo atlikti eksperimentai, kai vienu metu buvo vizualizuojamas skirtingas trikampių skaičius:

- 1 trikampis – eksperimentuose vadinama T1;
- 3 trikampiai – eksperimentuose vadinama T3;
- 7 trikampiai – eksperimentuose vadinama T7.

Buvo atlikti ir sudėtinės figūros – sferos vizualizacijos eksperimentai. Sferos buvo sudaromos iš skirtingo elementarių figūrų skaičiaus:

- iš 100 figūrų – eksperimentuose vadinama S100;
- iš 2500 figūrų – eksperimentuose vadinama S2500;
- iš 10000 figūrų – eksperimentuose vadinama S10000.

Kiekvienas eksperimentas (vizualizacija) buvo atliktas nustatant figūroms skirtingus gilumus (atitolinimus). Eksperimentuose buvo naudojami tokie gilumai:

- 0 – figūros nėra atitolinamos;
- 2 – figūros atitolinamos tam tikru atstumu, vaizduoklyje matyti visas figūras;
- 2,5 – figūros atitolinamos ir vaizduoklyje matyti tik trys ketvirtadaliai figūrų;
- 3 – figūros atitolinamos ir vaizduoklyje matyti tik pusę figūrų;
- 3,5 – figūros atitolinamos ir vaizduoklyje matyti tik ketvirtadalis figūrų;
- 4 ir 8 – figūros atitolinamos ir vaizduoklyje figūrų nematyti.

Pavyzdžiui, jei gilumas yra 2, objektas atrodys 2 kartus mažesnis už tą, kurio gilumas yra 0.

Eksperimentai buvo atliekami įvairiuose kompiuteriuose su skirtingomis *Windows* operacinės sistemos versijomis ir skirtingomis kompiuterių komplektacijomis (CPU, GPU, RAM). *DirectX* ir *OpenGL* greitaveika matuojama atnaujinimo kadrais per sekundę (atnaujinimo dažnis) (angl. *Frames Per Second* – FPS).

EEksperimentams sukurtomis programomis gauti vaizdai parodyti 1 pav. ir 2 pav. Juose matyti trikampių ir sferų vizualizacijos, kai gilumas nesiskyrė (jis šiais atvejais lygus 0). Sferų vizualizacijos, kai nustatyti skirtingi gilumai sferai, sudarytai iš 2500 elementarių figūrų, matyti 3 pav.

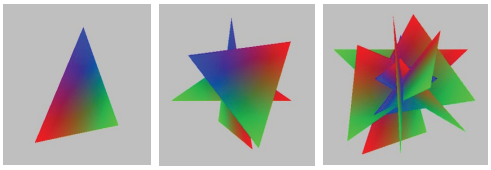
Eksperimentuose naudoti kompiuteriai ir jų komplektacijos patekiamos 2 lentelėje.

Vizualizacijos metu atnaujinimo dažnis šiek tiek kinta, todėl kiekvienas pirmiau aprašytas eksperimentas truko 1 minutę ir buvo kartojamas po 10 kartų kiekviename kompiuteryje. Taigi bendra eksperimentų trukmė buvo mažiausiai 168 valandos. Buvo apskaičiuoti 10-ies eksperimentų rezultatų vidurkiai.

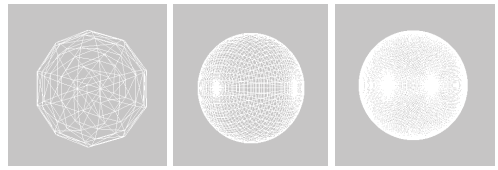
Skirtinguose kompiuteriuose atliktų vizualizacijų rezultatai pateikiami 4 pav. Siekiant palyginti *DirectX* ir *OpenGL* technologijų trimačių vaizdų vizualizavimo spartą atliekant skirtingų figūrų vizualizacijas, buvo apskaičiuotos kiekvieno iš šešių eksperimentų, atliktų su visais gilumais, rezultatų vidutinės reikšmės (kiekvieno eksperimento atskirai).

Taikant *DirectX* technologiją visuose eksperimentams naudotuose kompiuteriuose pasireiškia C# kalbos pranašumas (4 pav.). Kitaip veikė *OpenGL* technologija: pastebėta, kad *kompiuteryje 1, kompiuteryje 3* ir *kompiuteryje 4* C++ kalba pranašesnė nei C#. Be to, beveik visuose eksperimentuose nustatyta, kad *DirectX* pranašumas už *OpenGL* technologiją.

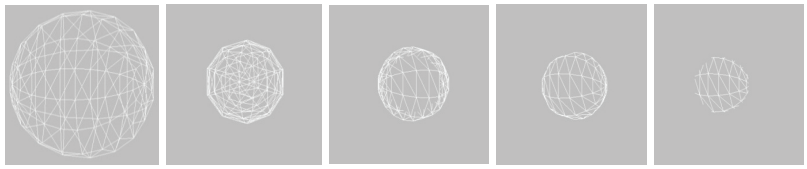
5 pav. yra pateiktos visų kompiuterių kiekvieno eksperimento, atlikto su visais gilumais, rezultatų vidutinės reikšmės. Akivaizdu,



1 p a v. ***Trikampių vizualizacijos***



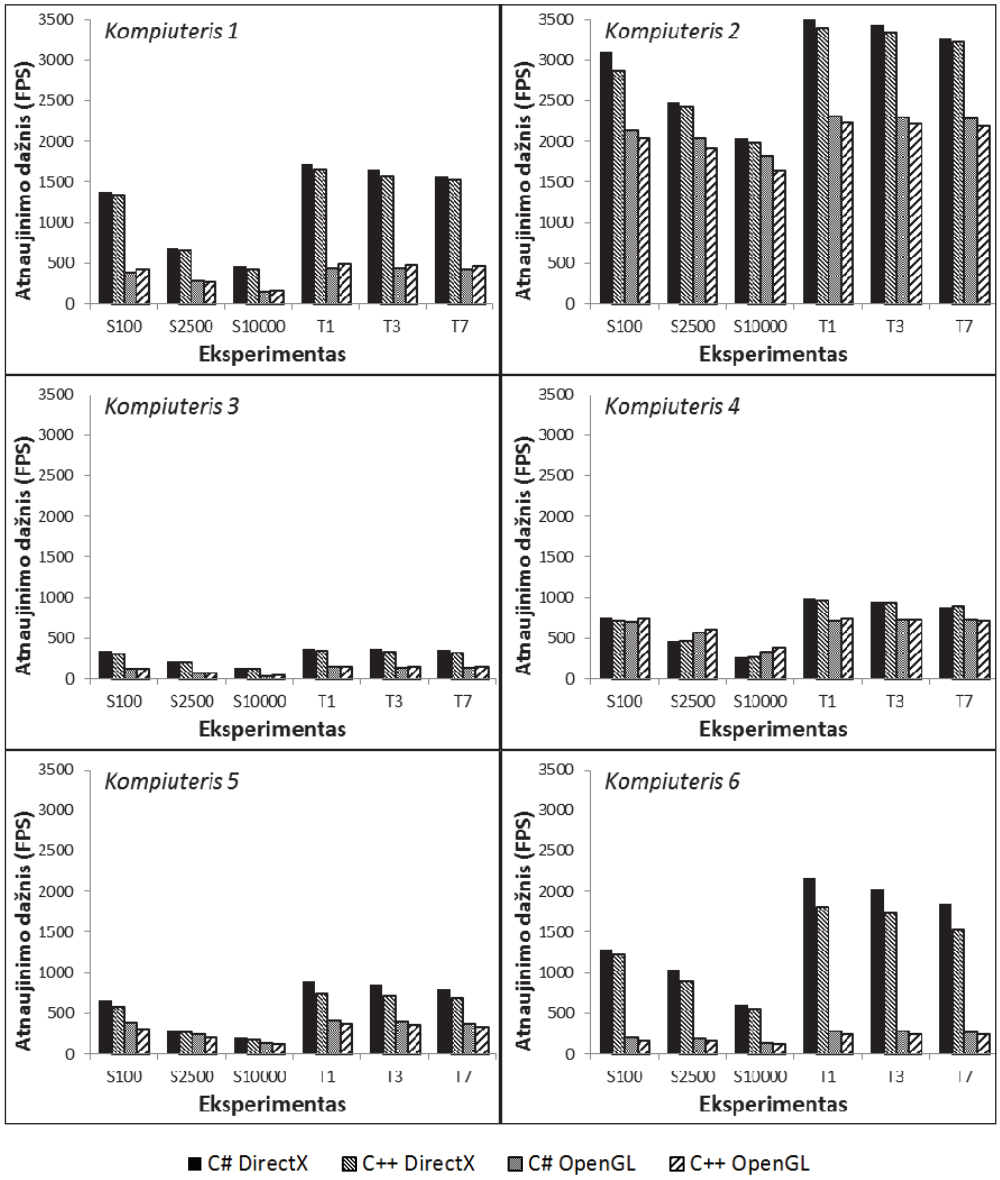
2 p a v. ***Sferų vizualizacijos***



3 p a v. ***Sferos vizualizacijos su skirtingu gilumu (0; 2; 2,5; 3; 3,5)***

2 l e n t e l ė. ***Eksperimentams naudoti kompiuteriai***

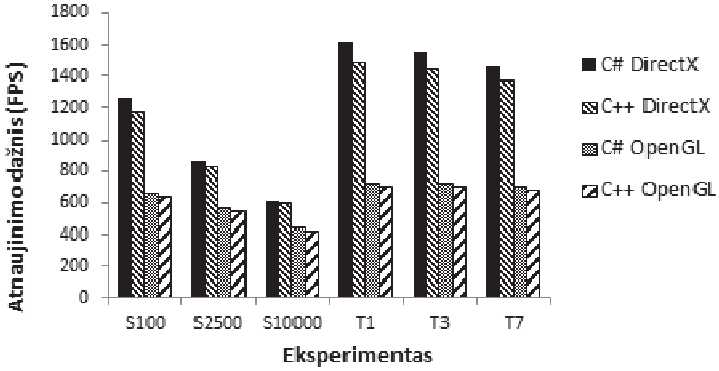
Kompiuterio pavadinimas	**Kompiuterio modelis**	**Operacinė sistema**	**CPU (centrinis procesorius)**	**GPU (vaizdo plokštės procesorius)**	**RAM (darbinė atmintinė)**
Kompiuteris 1	Lenovo X201	Windows 7 x64	Intel Core i7 M 620 2.67GHz	Intel Graphic Media Accelerator HD	8 GB
Kompiuteris 2	Asmeninis kompiuteris 1	Windows 7 x64	Intel Core Quad Q9300 2.5GHz	2xAMD Radeon HD 6900 Series	16 GB
Kompiuteris 3	Asus EeePC	Windows XP x32	Intel(R) Atom(TM) 1.6GHz	Mobile Intel(R) 945 Express Chipset Family	2 GB
Kompiuteris 4	Samsung NP350V5C	Windows 8 x64	Intel Core i3-3110M CPU 2.40GHz	Intel HD Graphics 4000	4 GB
Kompiuteris 5	ASUS KJ501J	Windows 7 x64	Pentium Dual Core T4200 2.00GHz	Mobile Intel 45 Express Chipset Family	4 GB
Kompiuteris 6	Asmeninis kompiuteris 2	Windows 7 x64	AMD Athlon 64 X2 Dual Core Processor 5000+ 2.6 GHz	ATI Radeon Xpress 1150	4 GB



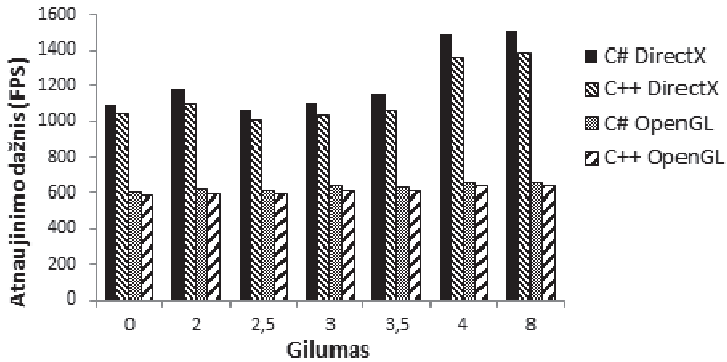
4 p a v. ***Skirtingų figūrų vizualizavimo eksperimentų rezultatai***

kad *DirectX* technologija yra pranašesnė už *OpenGL*. Esant gana nesudėtingam vizualizacijos uždaviniui (eksperimentas S100) *DirectX* yra 1,94 karto spartesnė už *OpenGL*, kai programuojama C# kalba, ir 1,86 karto, kai programuojama C++. Šis pranašumas sumažėja, kai vizualizacijos uždavinys tampa sudėtingesnis (eksperimentas S10000) – *DirectX* spartesnė 1,41 karto, kai programuojama C#, ir 1,43 karto, kai programuojama C++.

Kaip ir galima buvo tikėtis, visuose kompiuteriuose, didinant vizualizuojamų elementarių figūrų skaičių (eksperimentai T1, T3, T7), *DirectX* ir *OpenGL* technologijų vizualizacijos sparta mažėja. Lyginant elementarių figūrų vizualizacijas (eksperimentai T1 ir T7), *DirectX*



5 p a v. ***Visų kompiuterių skirtingų figūrų judėjimo eksperimentų rezultatų vidurkiai***



6 p a v. ***Visų kompiuterių skirtingų gilumų rezultatų vidurkiai***

veikimo sparta sumažėja 10 proc, kai programuojama C# kalba, ir 8 proc – kai C++; veikiant *OpenGL* – atitinkamai 2 proc. ir 3 proc.

Vizualizacijos sparta mažėja, kai didinamas sudėtinės figūros elementų skaičius (eksperimentai S100, S2500, S10000). Lyginant sudėtinių figūrų vizualizacijas (eksperimentai S100 ir S10000), *DirectX* atveju sparta sumažėja 51 proc., kai programuojama C# kalba ir 50 proc. – kai C++; naudojantis *OpenGL* – atitinkamai 33 proc. ir 34 proc.

Taigi *DirectX* technologija uždavinio sudėtingumui yra jautresnė nei *OpenGL* – sudėtingėjant uždaviniui *DirectX* sparta krinta greičiau.

Apskaičiuoti visų šešių kiekvieno gilumo eksperimentų rezultatų vidurkiai pateikiami 6 pav. Teoriškai, didinat gilumą, atnaujinimo dažnis turėtų didėti, nes vizualizuojant mažesniu gilumu, ekrane piešiama didesnė figūra, o tai reikalauja daugiau skaičiavimo resursų. Tuo įsitikinama lyginant atvejus, kai gilumas yra 0 ir 2, – atnaujinimo sparta antru atveju yra vidutiniškai didesnė. Tačiau tais atvejais, kai piešiama tik figūros dalis (gilumai 2,5; 3; 3,5), procesoriams reikia atlikti papildomų skaičiavimų. Todėl, kai gilumas yra 2,5, atnaujinimo sparta sumažėja lyginant su rezultatu, kai gilumas yra 2. Tais atvejais, kai vizualizuojant rodoma tik figūros dalis, kuo mažesnę dalį reikia vizualizuoti, tuo atnaujinimo sparta yra didesnė. Kai gilumas yra 4 ir 8, vizualizacijos skaičiavimai atliekami, tačiau ekrane vaizdo nematyti, todėl atnaujinimo dažnis padidėja, palyginant su visais gilumo atvejais.

Tinklalapiuose *http://www.cpubenchmark.net* ir *http://www.videocardbenchmark.net* skelbiami nepriklausomos bendrovės *PassMark® Software Pty Ltd* kompiuterio centrinio procesoriaus ir vaizdo plokštės našumo testų rezultatai. Testai atliekami skirtinguose savanoriškai dalyvaujančių *PassMark* programoje vartotojų kompiuteriuose. *CPU Benchmarks* indeksas gaunamas apskaičiuojant vidurkį iš skirtingų 8 centrinio procesoriaus našumo testų (CPU, 2013). 2013 m. gegužės 17 d. duomenimis, sparčiausio procesoriaus indeksas buvo 14969. *Videocard Benchmarks* indeksas gaunamas apskaičiuojant vidurkį iš skirtingų trijų vaizdo plokštės našumo testų (PassMark, 2013). Tais pačiais duomenimis, sparčiausios vaizdo plokštės indeksas buvo 8404. Minėti testai detaliau aprašyti *PassMark® Software Pty Ltd* bendrovės tinklapyje *http://www.passmark.com/*.

Trečioje lentelėje pateikti eksperimentuose naudotų kompiuterių *MarkPass CPU* ir *MarkPass G3D* indeksai (reitingai) bei skirtingų figūrų vizualizavimo eksperimentų rezultatai (*DirectX* ir *OpenGL* technologijų spartos santykis), apskaičiuojant kiekviename eksperimente gautų rezultatų vidurkį. *MarkPass CPU* – centrinio procesoriaus indeksas pagal *CPU Benchmarks – http://www.cpubenchmark.net* (žiūrėta 2013-05-17); *MarkPass G3D* – vaizdo plokštės indeksas pagal *Videocard Benchmarks – http://www.videocardbenchmark.net* (žiūrėta 2013-05-17).

3 lentelė. ***DirectX ir OpenGL technologijų spartos skirtinguose kompiuteriuose santykis***

Kompiuteris	MarkPass CPU	MarkPass G3D	DirectX ir OpenGL spartos santykis	
			C#	C++
Kompiuteris 1	2709	113	3,50	3,16
Kompiuteris 2	3191	2x1983	1,38	1,41
Kompiuteris 3	311	4	2,78	2,41
Kompiuteris 4	3036	479	1,15	1,08
Kompiuteris 5	1163	40	1,90	1,87
Kompiuteris 6	1311	29	6,49	6,54

Iš 3 lentelėje pateiktų duomenų matyti, jog kai kompiuterio procesorius „galingas“, o vaizdo plokštė „silpna“ (*kompiuteris 1, kompiuteris 3, kompiuteris 5, kompiuteris 6*), tai *DirectX* tecnnologija yra daug pranašesnė už *OpenGL* nepriklausomai nuo programavimo kalbos. Kitais atvejais, kai vaizdo plokštės yra galingesnės, *DirectX* technologijos pranašumas sumažėja. Tačiau, norint tiksliau ištirti šias tendencijas, reikėtų atlikti daugiau eksperimentų skirtinguose kompiuteriuose.

Išvados

Straipsnyje palygintos ir eksperimentiškai ištirtos dvi populiariausios trimačių vaizdų vizualizavimo technologijos – *DirectX* ir *OpenGL* naudojant C# ir C++ programavimo kalbas. Atlikti minėtų technologijų spartos eksperimentiniai tyrimai bei lyginamoji analizė leido padaryti tokias išvadas:

- *DirectX* technologija trimačius vaizdus vizualizuoja sparčiau nei *OpenGL*, vidutiniškai 1,9 karto.
- Sudėtingėjant vizualizavimo uždaviniui, *DirectX* technologijos sparta krinta greičiau nei *OpenGL* technologijos ir skirtumas tarp nagrinėtų technologijų spartų sumažėja.
- *DirectX* technologija geriau išnaudoja kompiuterio centrinio procesoriaus ir vaizdo plokštės išteklius, ypač atliekant nesudėtingas vizualizacijas.
- Vizualizuojant trimačius vaizdus *DirectX* technologija, C# programavimo kalba yra pranašesnė už C++ kalbą. *OpenGL* technologijos taikyme programavimo kalbos pranašumo negalima nustatyti.

LITERATŪRA

CAI, J.; LIU, Q.; YU, X. (2011). Research and realization of shadowgraph performance technology based on *DirectX*. In: *Proceedengs of International Conference on Graphic and Image Processing* (ICGIP 2011).

Comparison (2013). Comparison of *OpenGL* and Direct3D. In: *Wikipedia*, the free encyclopedia [interaktyvus] [žiūrėta 2013 m. gegužės 28 d.]. Prieiga per internetą: <http://en.wikipedia.org/wiki/Comparison_of_*OpenGL*_and_Direct3D>.

COM (2013). Component Object Model. In: *Wikipedia*, the free encyclopedia [interaktyvus] [žiūrėta 2013 m. gegužės 28 d.]. Prieiga per internetą: <http://en.wikipedia.org/wiki/Component_Object_Model>.

CPU (2013). CPU Benchmarks – *CPU Test Information* [žiūrėta 2013 m. gegužės 17 d.]. Prieiga per internetą: <http://www.cpubenchmark.net/cpu_test_info.html>.

DirectX Graphics (2013). *DirectX Graphics and Gaming*. Windows Dev Center [interaktyvus] [žiūrėta 2013 m. gegužės 28 d.]. Prieiga per internetą: <http://msdn.microsoft.com/en-us/library/windows/desktop/ee663274(v=vs.85).aspx>.

DirectX (2013). In: *Wikipedia*, the free encyclopedia [interaktyvus] [žiūrėta 2013 m. gegužės 28 d.]. Prieiga per internetą: <http://lt.wikipedia.org/wiki/*DirectX*>.

Download (2013). *Download Center – Download DirectX Software Development Kit from Official Microsoft Download Center* [žiūrėta 2013 m. liepos 10 d.]. Prieiga per internetą: <http://www.microsoft.com/en-us/download/details.aspx?id=6812>.

Freeglut (2013). *The freeglut Project* : About [žiūrėta 2013 m. liepos 10 d.]. Prieiga per internetą: <http://freeglut.sourceforge.net/>.

GUO, P. ; CHENG, Y.; LIU, Z. J. (2012). Design and Implementation of 3D Graphics Rendering Engine with *DirectX*. In: *Communications and Information Processing*. Springer.

HILL, F. S.; KELLEY, S. M. (2007). *Computer Graphics Using OpenGL*, 3/E. Prentice Hall.

KOCHAN, S. G. (2011). *Programming in Objective-C*. Addison-Wesley Professional

LUNA, F. D. (2006). *Introduction to 3D Game Programming with DirectX 9.0c: A Shader Approach*. Plano, Tex.: Wordware Pub.

LUNA, F. D. (2008). *Introduction to 3D Game Programming With DirectX 10.* Plano, Tex.: Wordware Pub.

MAŽUTIS, T. (2009). *Trimatės grafikos elementų panaudojimas vizualizacijos sistemose*: Magistro darbas.

Microsoft Direct3D (2013). In: *Wikipedia*, the free encyclopedia [interaktyvus] [žiūrėta 2013 m. liepos 10 d.]. Prieiga per internetą:<http://en.wikipedia.org/wiki/Microsoft_Direct3D>.

Mono (2013). *Download Mono* [žiūrėta 2013 m. liepos 10 d.]. Prieiga per internetą: <http://www.go-mono.com/mono-downloads/download.html>.

MULHOLLAND, A.; HAKALA, T. (2004*). Programming Multiplayer Games*. Wordware Publishing, Inc.

OpenGL – The Industry Standard for High Performance Graphics. The Industry's Foundation for High Performance Graphics [žiūrėta 2013 m. gegužės 28 d.]. Prieiga per internetą: <http://www.*OpenGL*.org/>.

OpenGL (2013). In: *Wikipedia*, the free encyclopedia [interaktyvus] [žiūrėta 2013 m. gegužės 28 d.]. Prieiga per internetą: <http://en.wikipedia.org/wiki/*OpenGL*>.

OpenGL v *DirectX* [žiūrėta 2013 m. liepos 10 d.]. Prieiga per internetą: <http://www.blitzbasic.com/Community/posts.php?topic=53219>.

PassMark (2013). *PassMark Software – Video Card Benchmarks – Video Card Test Information* [žiūrėta 2013 m. gegužės 17 d.]. Prieiga per internetą: <http://www.videocardbenchmark.net/gpu_test_info.html>.

SEBASIAN, A. (2012). *Valve: OpenGL is faster than DirectX – even on Windows* [žiūrėta 2013 m. liepos 10 d.]. Prieiga per internetą: <http://www.extremetech.com/gaming/133824-valve-*OpenGL*-is-faster-than-*DirectX*-even-on-windows>.

SHERROD, A.; JONES, W. (2011). *Beginning DirectX 11 Game Programming*. Course Technology Press, Boston, MA, United States.

SHEINER, D.; SELLERS, G.; KESSENICH, J. M.; LINCEA-KANE, B. M. (2013). *OpenGL programming guide: the official guide to learning OpenGL, versions 4.1*. Upper Saddle River, NJ: Addison-Wesley.

TIOBE Software – Coding Standard Company. In: *TIOBE Programming Community Index for April 2013* [žiūrėta 2013 m. gegužės 17 d.]. Prieiga per internetą: <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>.

What is COM? – COM: Component Object Model technologies [žiūrėta 2013 m. liepos 10 d.]. Prieiga per internetą: <http://www.microsoft.com/com/default.mspx>.

WRIGHT, Jr. R. S.; HAEMEL, N. S.; SELLERS, G.; LIPCHAK, B. (2010). *OpenGL SuperBible: comprehensive tutorial and reference*. Addison-Wesley.

ZHANG, P.; WANG, H. Y.; LI, Y. G.; MAO, S. F.; DING, Z. J. (2012). Monte Carlo simulation of secondary electron images for real sample structures in scanning electron microscopy. *Scanning*, vol. 34(3), p. 145–150.

COMPARISON OF OPENGL AND DIRECTX TECHNOLOGIES IN 3D PROGRAMMING

Ernestas Filatovas, Grigorijus Melničenko, Vytautas Jakštys

S u m m a r y

In this study, three-dimensional rendering technologies *DirectX* and *OpenGL* were reviewed. A comprehensive survey, evaluation of advantages and disadvantages, and a comparative analysis of these technologies have been made.

By applying C# and C++ programming languages, the software has been developed for testing the speed of *DirectX* and *OpenGL* technologies. A relatively large number of experiments were conducted in order to determine the speed of these technologies. The experiments included visualization of basic and complex shapes movements, and different depth of shapes.

The experimental investigations were performed in Windows and they have revealed that the speed of the *DirectX* technology is higher. This factor is the most significant one in three-dimensional rendering, especially in the development of computer games. The *DirectX* technology was proved to be fastest while programming in C#. Moreover, the experimental investigations have indicated that the *DirectX* technology uses CPU resources more efficiently.

DUOMENŲ ANALIZĖ IR VAIZDAVIMAS

Duomenų tyrybos sistemos, pagrįstos saityno paslaugomis

Olga Kurasova

Vilniaus universiteto Matematikos ir informatikos instituto daktarė
Vilnius University, Institute of Mathematics and Informatics, PhD, Senior researcher
Akademijos g. 4, LT-08663 Vilnius
El. paštas: Olga.Kurasova@mii.vu.lt

Virginijus Marcinkevičius

Vilniaus universiteto Matematikos ir informatikos instituto daktaras
Vilnius University, Institute of Mathematics and Informatics, PhD, Senior researcher
Akademijos g. 4, LT-08663 Vilnius
El. paštas: Virginijus.Marcinkevicius@mii.vu.lt

Viktor Medvedev

Vilniaus universiteto Matematikos ir informatikos instituto daktaras
Vilnius University, Institute of Mathematics and Informatics, PhD, Researcher
Akademijos g. 4, LT-08663 Vilnius
El. paštas: Viktor.Medvedev@mii.vu.lt

Aurimas Rapečka

Vilniaus universiteto Matematikos ir informatikos instituto doktorantas
Vilnius University, Institute of Mathematics and Informatics, PhD student
Akademijos g. 4, LT-08663 Vilnius
El. paštas: Aurimas.Rapecka@mii.vu.lt

Straipsnis skirtas duomenų tyrybos, pagrįstos saityno paslaugomis, analizei. Apibrėžiamos pagrindinės su saityno paslaugomis susijusios sąvokos. Pristatomos paskirstytosios duomenų tyrybos galimybės bei jų įgyvendinimo priemonės – Grid, Hadoop. Atliekama duomenų tyrybos sistemų, pagrįstų saityno paslaugomis, analitinė apžvalga. Parenkami sistemų palyginimo kriterijai. Pagal šiuos kriterijus atliekama populiariausių duomenų tyrybos sistemų, pagrįstų saityno paslaugomis, lyginamoji analizė. Nustatoma, kurios sistemos įvertinamos geriausiai, o kurios neatitinka daugumos kriterijų.

Įvadas

Duomenų tyryba (angl. *data mining*) yra svarbi žinių radimo duomenų bazėse proceso dalis. Kurį laiką tyrimai buvo nukreipti į duomenų tyrybos metodų kūrimą ir jų taikymą. Buvo intensyviai kuriamos įvairios duomenų tyrybos sistemos: *Weka* (Hall et al., 2009), *Orange* (Podpecan et al., 2012), *Knime* (Berthold et al., 2007), *RapidMiner* (Mierswa et al., 2006), R (Knell, 2013) ir kt. Pastaruoju metu sparčiai vystomi saityno paslaugų (angl. *web services*) tyrimai. Duomenų tyrybos sistemos kuriamos naudojant paslaugų architektūrą (angl. *Service-Oriented Architecture*, SOA). Be to, yra bandymų sukurti duomenų tyrybos algoritmus kaip saityno paslaugas, kurios gali būti panaudotos kitose sistemose, praplečiant jų galimybes be papildomo programavimo (FAEHIM, 2005). Dar viena aktuali duomenų tyrybos problema – didelių apimčių duomenų analizė, reikalaujanti ne tik specialių algoritmų, bet ir naujų technologijų.

Šio straipsnio tyrimo objektas – duomenų tyrybos sistemos, pagrįstos saityno paslaugomis, tyrimo metodika – analizuojamų sistemų lyginamoji analizė. Straipsnio tikslas – apžvelgti esamas duomenų tyrybos sistemas, pagrįstas saityno paslaugomis; parinkti kriterijus, pagal kuriuos atlikti šių sistemų lyginamąją analizę, ir išryškinti lyginamų sistemų pranašumus ir trūkumus. Tokia lyginamoji analizė bus naudinga kuriant naują duomenų tyrybos sistemą.

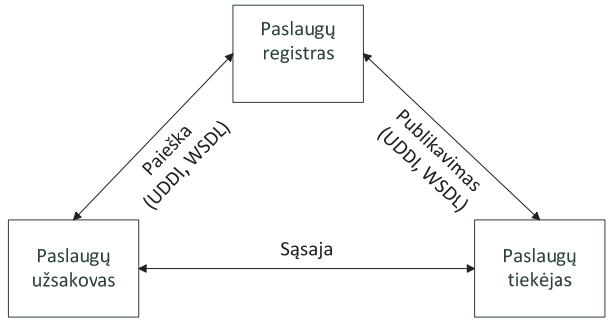
Paskirstytosios duomenų tyrybos įgyvendinimo priemonės

Pastaraisiais metais sparčiai plinta saityno paslaugos dėl jų paprasto panaudojimo ir plataus paslaugų stiliaus architektūros paradigmos taikymo įvairiose srityse. Paslaugų stiliaus architektūra – tai naujas programinės įrangos architektūros modelis, kuris užtikrina verslo procesų transformavimą į aibę susietų paslaugų (ar aibę pasikartojančių verslo užduočių), kurios gali būti pasiektos per saityną (IBM Developers Works, 2013). Transformavimas gali būti atliktas tiek lokaliame tinkle, tiek saityne ar serveryje. Saityno paslaugos jungiamos siekiant išspręsti specifines verslo užduotis ar vykdant sudėtingą verslo procesą. Verslo procesų sudėtingumą lemia tai, kad jiems vykdyti paprastai turi būti pasitelkta keletas skirtingų programinės įrangos paketų, kurie yra skirtingose kompiuterinėse platformose. SOA architektūra leidžia spręsti minėtas problemas naudojant programinės įrangos ir infrastruktūros perorganizavimo į aibę tarpusavyje bendraujančių paslaugų būdą (Papazoglou, 2003). Būtent tai užtikrina, jog saityno paslaugos gali būti greitai ir efektyviai pritaikytos prie kintančių verslo sąlygų.

Paslauga (angl. *service*) vadinamas programos blokas, gebantis atlikti specifines užduotis ar funkcijas, naudojant griežtai apibrėžtas sąsajas (IBM developers Works, 2013). Paslaugos veikia sistemos viduje, tačiau kiti sistemos komponentai nežino, kaip jos užtikrina savo funkcijas – svarbus tik paslaugos grąžinamas rezultatas. Saityno paslauga – tai viename komponente esantis funkcijų, pasiekiamų pasinaudojus standartiniais interneto protokolais, rinkinys. Paslaugos prieinamumas internetu suteikia galimybę integruoti skirtingose platformose veikiančias programas. Jos lengvai įdiegiamos dėl sukurtų saityno paslaugų standartų ir jų suderinamumo su visuotinai paplitusiomis ir naudojamomis technologijomis: XML (angl. *Extensible Markup Language*), SOAP (angl. *Simple Object Access Protocol*), REST (angl. *Representational State Transfer*), WSDL (angl. *Web Services Description Language*) ir UDDI (angl. *Universal Description Discovery and Integration*).

Kuriant saityno paslaugas ir naudojant paslaugų stiliaus architektūrą dalyvauja trys veikėjai: paslaugų registras, paslaugų tiekėjas ir paslaugų užsakovas (1 pav.). Paslaugų registre saugoma informacija apie siūlomas ar pasiekiamas saityno paslaugas, tam naudojant WSDL ir UDDI formatus. Paslaugų tiekėjai publikuoja savo siūlomas saityno paslaugas ir užtikrina jų funkcionalumą. Paslaugų užsakovas ieško tinkamų paslaugų, realizuojančių norimą funkcionalumą, ir integruoja su kitomis saityno paslaugomis, siekdamas įvykdyti tam tikrą procesą. Atliekant saityno paslaugos paiešką kreipiamasi į paslaugų registrą ir gaunamas bendras paslaugos aprašymas UDDI formatu arba surastos saityno paslaugos WSDL, kuriuo remiantis galima tiesiogiai kreiptis į paslaugos tiekėją.

Dažnai tiriami duomenys yra didelių apimčių. Nustatyta, kad 90 proc. visų pasaulio duomenų yra sukurta tik per pastaruosius dvejus metus. Todėl nuolat kyla poreikis analizuoti didelius duomenų kiekius. Tokių duomenų tyryba reikalauja



1 p a v. ***Saityno paslaugų architektūra (Kreger, 2001)***

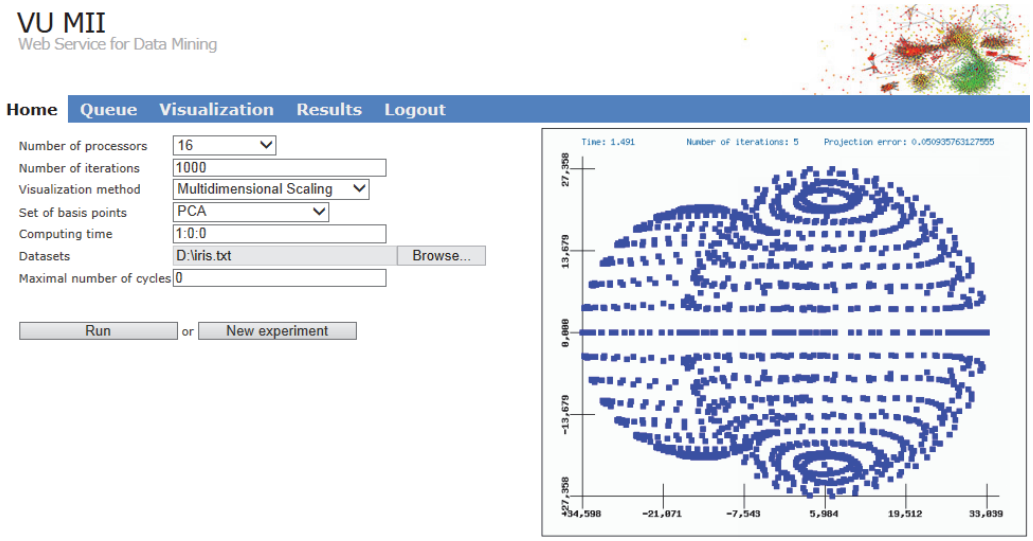
didelių tiek skaičiavimo išteklių, tiek saugojimo galimybių. Todėl pavienių kompiuterių skaičiavimo išteklių neužtenka. Didelių skaičiavimo išteklių reikalaujantiems uždaviniams spręsti naudojama vadinamoji paskirstytoji duomenų tyryba (angl. *distributed data mining*), kai duomenų tyrybos uždaviniai lygiagrečiai sprendžiami keliuose kompiuteriuose (Talia, Trunfio, 2012). Yra sukurtos kelios platformos, įgyvendinančios paskirstytąją duomenų tyrybą: *DataMiningGrid* (Stankovski et al., 2008), *Knowledge Grid* (Congiusta et al., 2008), *Discovery Net* (Čurčin et al., 2002).

Dažnai didelių skaičiavimo išteklių reikalaujantiems uždaviniams spręsti pasitelkiami gridai (angl. *Grid*). Gridas – tai laisvai prieinama, suderinta ir standartizuota infrastruktūra, kuri užtikrina lankstų, saugų, koordinuotą skaičiavimų ir informacijos saugojimo išteklių paskirstymą (Foster, Kesselman, 2004). Pagrindinis aspektas, skiriantis gridą nuo kompiuterių klasterio, yra tas, kad klasterius sudaro homogeniniai kompiuteriai, o gridą – heterogeniniai (Jadhav, 2009). Bendruoju atveju gridas – plačiai geografiškai paskirstyta infrastruktūra, kuri jungia daug skirtingo tipo išteklių. Prieigą prie šių išteklių naudotojas gali gauti iš bet kurios fizinės vietos, nepriklausomai nuo jų išdėstymo.

Gridų galimybes iliustruojančiu pavyzdžiu galima laikyti BOINC (angl. *Berkeley Open Infrastructure for Network Computing*) gridus, sukurtus naudojant programinę įrangą (BOINC, 2013). Ši programinė įranga, kurta specialiai SETI@Home (SETI@Home, 2013) projektui, kurio pagrindiniam tikslui, nežemiškos gyvybės signalų paieškai kosmoso signalų sraute, reikėjo labai didelių skaičiavimo išteklių, ilgainiui tapo atvira ir prieinama visiems norintiems kurti savo gridus. Šiuo metu BOINC programine įranga naudojasi apie 80 atvirų gridų. Pasauliniu mastu patys populiariausi – Einstein@Home (Einstein@Home, 2013) ir jau minėtas SETI@Home, gride turintys atitinkamai apie 4,2 ir 3,4 mln. kompiuterių (BOINCstats, 2013). Nors BOINC programinė įranga labai paplitusi, tačiau ji nėra vienintelė. Gridams kurti naudojamos ir kitos atvirojo kodo programos, pavyzdžiui, *Globus Toolkit* (Globus Toolkit, 2013), *GridWay* (GridWay, 2013) ir kt.

Pastaruoju metu paskirstytiems skaičiavimas vis plačiau taikoma *Hadoop* projekto programinė įranga, apimanti keletą atvirojo kodo programinės įrangos projektų, tokių kaip *MapReduce*, HDFS, *Hbase*, *Hive*, *Pig* ir kt. (Hadoop, 2012). *Hadoop* sukūrė Dougas Cuttingas, vystydamas *Apache Nutch* projektą, kuris buvo plačiai naudojamas kaip biblioteka teksto paieškai. *Apache Nutch* projektas pradėtas 2002 metais, o nuo 2006 metų tai jau *Hadoop* projektas, turintis pagrindines komponentes *MapReduce* ir HDFS. *MapReduce* – paskirstytų duomenų apdorojimo modelis ir kartu vykdymo aplinka, veikianti dideliuose kompiuterių klasteriuose (Whie, 2012). Kita svarbi komponentė – paskirstyta failų sistema, arba HDFS (angl. *Hadoop Distributed File System*). Naudojantis šia sistema, dideli failai sudalijami į 64 MB ar didesnes vienodas dalis, kurios tolygiai paskirstomos po kompiuterio klasterio kompiuterius. Palyginti su gridu, tai leidžia taupyti laiką, reikalingą duomenims persiųsti iš vieno kompiuterio į kitą. *MapReduce* – tai *Google* kompanijos inžinierių darbas, skirtas paieškos rezultatams indeksuoti, tačiau dabar ji taikoma įvairiausiose srityse: vaizdų analizei, kompiuteriniam mokymui (angl. *machine learning*). Įvairioms duomenų tyrybos sritims yra sukurta su *Hadoop* veikianti *Mahout* biblioteka (Mahout, 2011), kurioje įgyvendinti populiarūs klasifikavimo, klasterizavimo, regresijos, daugiamačių duomenų vizualizavimo (dimensijos mažinimo), evoliuciniai algoritmai. Tačiau šių algoritmų nėra daug bei dėl *MapReduce* specifikos ne visada lengvai ir efektyviai galima panaudoti esamus duomenų tyrybos algoritmus. Todėl kitame skyriuje nagrinėsime kelias populiarias duomenų tyrybos sistemas, kurios orientuojasi į saityno paslaugas. Dažnai tokios sistemos palaiko paskirstytųjų skaičiavimų galimybes.

Paskirstytosios duomenų tyrybos sistemos pradedamos kurti ir Lietuvoje. Yra sukurta interneto naršykle naudojama ir valdoma sistema, kurioje įgyvendinti daugiamačių duomenų vizualizavimo metodai (Dzemyda ir kt., 2011). Sistema leidžia vizualizuoti didelės apimties duomenų aibes (2 pav.), optimizuoti vizualizavimo algoritmų parametrus. Daugiamačių duomenų vizualizavimo metodai reikalauja daug skaičiuojamųjų sąnaudų, o rezultatus įprastai siekiama gauti per trumpą laiką, todėl visi skaičiavimai yra atliekami naudojant kompiuterių klasterį.



2 pav. ***Daugiamačių duomenų vizualizavimo sistema***

Duomenų tyrybos sistemos

Šiame skyriuje apžvelgiamos sistemos, pagrįstos saityno paslaugomis (*Weka4WS*, *Orange4WS*, *Knime*, *Taverna*, *ClowdFlows*, DAME), kuriose įgyvendinti įvairūs duomenų tyrybos metodai.

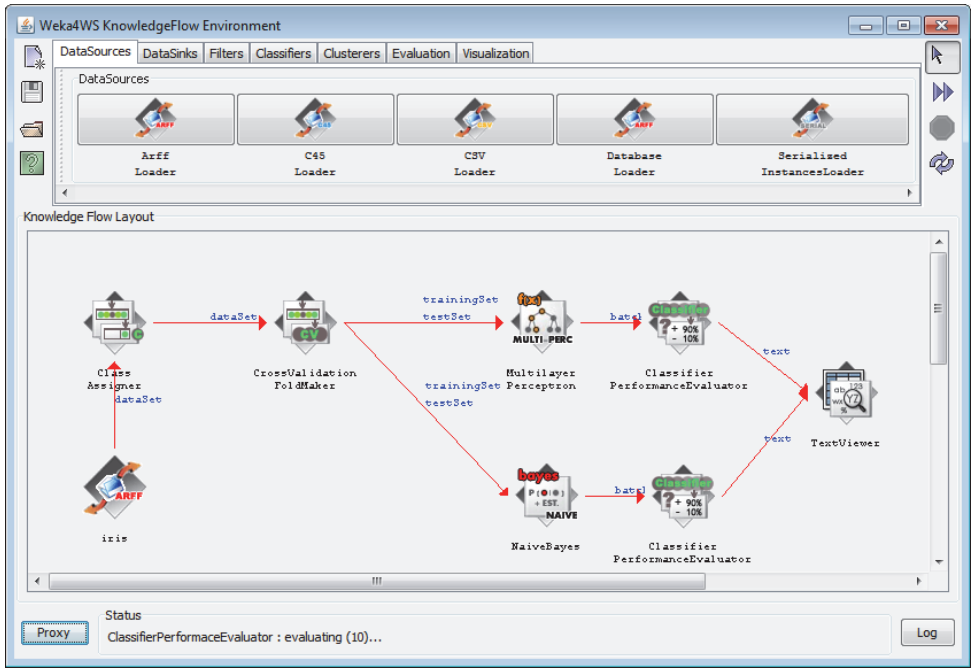
Weka4WS yra gerai žinomos populiarios *Weka* sistemos plėtinys, leidžiantis vykdyti paskirstytąją duomenų tyrybą griduose (Talia et al., 2008). Sistema turi būti įdiegta į naudotojo kompiuterį, tačiau joje įgyvendinta galimybė pasirinkti skaičiavimo išteklius ir skaičiavimus vykdyti lygiagrečiai. *Weka4WS* sukurta naudojant *Globus Aliance Java WSRF* (angl. *Web Service Resource Framework*) (Globus Toolkit, 2013) karkasą. Sistemoje yra įgyvendinta vadinamųjų darbų sekų (angl. *workflows*) sudarymo galimybė. Jos pavyzdys pateikiamas 3 paveiksle. Čia duomenys klasifikuojami dviem klasifikavimo algoritmais (*NaiveBayes* ir *Multilayer Perceptron*), kurie gali būti vykdomi skirtinguose grido mazguose.

Orange4WS yra kitos populiarios duomenų tyrybos sistemos *Orange* plėtinys (Podpecan, 2012). Lyginant su *Orange* sistema, joje įgyvendintos kelios naujos savybės: įrankiai, palengvinantys saityno paslaugų realizaciją naudojant *Python* programavimo kalbą; sistemoje įgyvendinta galimybė naudoti saityno paslaugas kaip darbų sekos komponentes.

Knime sistemoje taip pat galima naudoti saityno paslaugas (Berthold et al., 2007). Joje įgyvendintas saityno paslaugų klientas, leidžiantis prijungti jau sukurtas (išorines) saityno paslaugas prie *Knime* darbų sekų. Tam reikia pasirinkti grafinę komponentę *Generic Web Service Client*, įkelti ją į darbo langą bei nurodyti saityno paslaugos WSDL failo internetinį adresą, prijungti prie komponentės įvesties ir išvesties ar pasirinkti kitas reikalingas komponentes. Tokiu būdu bus sudaryta darbų seka, kuri esant būtinybei gali būti papildyta kitomis komponentėmis.

Programavimo ir skaičiavimo aplinkoje *Matlab* yra sukurtos funkcijos (*callSoapService*, *createClassFromWsdl*, *createSoapMessage*, *parseSoapResponse*), leidžiančios panaudoti saityno paslaugas.

Pastaruoju metu kuriamos duomenų tyrybos sistemos saityno paslaugų pagrindu. Tokie įrankiai įgyvendinti *MyGrid* projekte (myGrid, 2008), jungiančiame *myExperiment* (De Roure et al., 2008), *BioCatalogue* (Bhagat et al., 2010), *Taverna* (Missier et al., 2010) ir kitus įrankius bei aplinkas, skirtas tokioms mokslo kryptims kaip biologija, socialiniai mokslai,



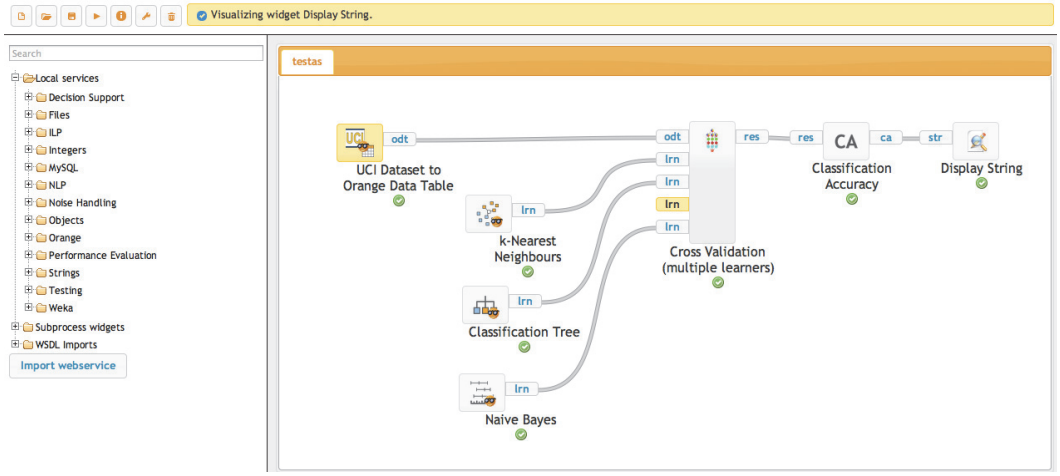
3 p a v. ***Darbų sekos pavyzdys „Weka4WS“ sistemoje***

muzika, astronomija, daugialypė terpė ir chemija. *BioCatalogue* yra saityno paslaugų gyvybės mokslams registras. *MyExperiment* bendradarbiavimo aplinkoje mokslininkai gali saugoti savo sukurtas darbų sekas, dalytis jomis su kitais, rasti norimų.

Taverna yra atvirojo kodo ir nuo srities nepriklausanti darbų sekų valdymo sistema (Missier et al., 2010). Sistemoje nėra saityno paslaugų, skirtų duomenų tyrybai, tačiau yra galimybė išorines saityno paslaugas lengvai prijungti ir iš jų sudaryti darbų sekas. Į sistemą integruotos *myExperiment* aplinkos paslaugos bei paslaugos iš *BioCatalogue*.

Visos iki šiol apžvelgtos sistemos turi būti įdiegtos į asmeninį kompiuterį. Nauja kryptis yra duomenų tyrybos saityno programos (angl. *web applications*), kurios naudojamos ir valdomos interneto naršykle. Dažniausia programų naudotojai turi galimybę susikurti savo aplinką tyrimams ir saugoti joje gautus rezultatus. Todėl šių programų pranašumas prieš sistemas, kurias būtina įdiegti į kompiuterį, yra tas, kad naudotojas savo aplinką gali pasiekti iš bet kur interneto ryšį turinčio kompiuterio be jokio papildomo diegimo. Viena tokių programų yra *ClowdFlows* sistema (Kranjc et al., 2012). Naudotojas turi užsiregistruoti sistemoje, sukurti norimas darbų sekas ir jas išsaugoti. Sistemos vaizdas pateikiamas 4 paveiksle. Čia sudaryta tokia darbų seka: paimami duomenys, esantys UCI duomenų saugykloje (Bache, Lichman, 2013), jie klasifikuojami trimis klasifikatoriais, įgyvendintais *Orange* sistemoje, rezultatai testuojami kryžmine patikra, gauti klasifikavimo tikslumai parodomi ekrane. Sistemoje yra įgyvendintos ir *Orange*, ir *Weka* sistemose esančios saityno paslaugos. Be to, yra galimybė prijungti ir kitas saityno paslaugas, nurodant WSDL failo internetinį adresą.

DAME (angl. *DAta Mining & Exploration*) yra paskirstytosios duomenų tyrybos infrastruktūra, skirta didelės apimties duomenims analizuoti, naudojant duomenų tyrybos metodus (DAME project, 2010; Brescia et al., 2012). Ši sistema taip pat naudojama ir valdoma interneto naršykle. Skaičiavimai atliekami ne vartotojo kompiuteryje, todėl didesnių apimčių duomenų analizė gali būti atliekama daug greičiau. Į DAME sistemą



4 pav. ***Darbų seka, sudaryta sistemoje „ClowdFlows“***

yra integruojama *Knime* sistemos funkcijos ir duomenų tyrybos metodai. DAME sistemoje neįgyvendintas darbų sekų sukūrimo mechanizmas, kuris yra daugumoje anksčiau minėtų sistemų.

Duomenų tyrybos sistemų lyginamoji analizė

Duomenų tyrybos sistemos yra lyginamos pagal įvairius kriterijus (Stankovski et al., 2008; Hmida, Slimani, 2010). Dažnai didžiausias dėmesys kreipiamas į duomenų tyrybos uždavinius, todėl trūksta lyginamosios saityno paslaugų analizės. Šiame straipsnyje atliekamos lyginamosios analizės tikslas – palyginti naujas ir vis dar tobulinamas duomenų tyrybos sistemas, pagrįstas saityno paslaugomis, bei išryškinti tų sistemų pranašumus ir trūkumus. Visos vertintos sistemos yra atvirojo kodo, laisvai prieinamos internete. Kriterijai, pagal kuriuos toliau bus lyginamos duomenų tyrybos sistemos, galimos reikšmės bei kriterijų svarbos pagrindimas, pateikiami 1 lentelėje.

Duomenų tyrybos sistemų, pagrįstų saityno paslaugomis, vertinimo rezultatai pateikti 2 lentelėje. Jei sistema atitinka nurodytą

1 lentelė. ***Sistemų lyginimo kriterijai***

Kriterijus	Galimos reikšmės	Pagrindimas
Informacijos perdavimo tarp saityno paslaugų būdas	SOAP, RESTful	Šiuo metu populiariausi būdai bendrauti su saityno paslaugomis, todėl svarbu, kad sistemose būtų įdiegiami abu būdai
Operacinės sistemos	MS Windows, Linux, Mac OS X	Svarbu, kad sistemos veiktų įvairiose operacinėse sistemose
Praplečiamumas	taip, ne	Svarbu, kad būtų įmanoma prijungti išorines saityno paslaugas be papildomo programavimo
Darbų sekos	taip, ne	Darbų sekos leidžia sukurti norimą aplinką eksperimentams, ją išsaugoti kitiems tyrimams
Saityno programa	taip, ne	Saityno programų pranašumas yra tas, kad jos naudojamos ir valdomos interneto naršykle be papildomo įdiegimo.
Duomenų tyrybos metodai	klasifikavimo, grupavimo, asociatyvių taisyklių, teksto tyryba	Sistemų universalumas yra labai svarbus duomenų tyrybos uždaviniams spręsti, nes labai dažnai tiems patiems duomenims būtina taikyti kelis skirtingus duomenų tyrybos metodus.

2 lentelė. ***Duomenų tyrybos sistemų palyginimas***

Kriterijai / Sistemos	Informacijos perdavimas		Operacinės sistemos			Praplečiamumas	Darbų sekos	Saityno programa	Duomenų tyryba				Iš viso
	SOAP	RESTful	MS Win	Linux	Mac OS X				klasifikavimo	grupavimo	asociat. taisyklių	teksto tyryba	
Weka4WS	+	–	+	+	–	–	+	–	+	+	+	–	7
Orange4WS	+	–	+	+	+	+	+	–	+	+	+	–	9
Knime	+	–	+	+	+	+	+	–	+	+	+	+	10
ClowdFlows	+	–	+	+	+	+	+	+	+	+	–	+	10
Taverna	+	+	+	+	+	+	+	–	–*	–*	–*	–*	7
DAME	–	+	+	+	+	–	–	+	+	+	–	–	7
Iš viso	5	2	6	6	5	4	5	2	5	5	3	2	

* sistemoje galima tik prijungti kitur sukurtas išorines duomenų tyrybos saityno paslaugas

kriterijų, žymima „+“, priešingu atveju – „–“. Paskutiniame stulpelyje „Iš viso“ nurodytas kiekvienos sistemos pliusų skaičius, o paskutinėje eilutėje „Iš viso“ nurodytas sistemų, atitinkančių nurodytą kriterijų, kiekis.

Weka4WS, *Orange4WS*, *Knime*, *ClowdFlows* sistemose saityno paslaugos sukurtos naudojant tik SOAP informacijos perdavimo būdą. DAME sistemoje naudojamas *RESTful*, o *Taverna* sistemoje galima įkelti ir SOAP, ir *RESTful* saityno paslaugas. Visos sistemos veikia *MS Windows* ir *Linux* operacinėse sistemose, o *Mac OS X* sistemoje neveikia tik *Weka4WS*. *Orange4WS, Knime*, *ClowdFlows* ir *Taverna* sistemose yra įgyvendinta galimybė įkelti kitur sukurtas (išorines) saityno paslaugas be papildomo programavimo. Darbų sekos įgyvendintos visose tirtose sistemose, išskyrus DAME. *ClowdFlows* ir DAME yra saityno programos. Visos keturios duomenų tyrybos metodų grupės (klasifikavimas, grupavimas, asociatyvios taisyklės bei teksto tyryba) yra įgyvendintos tik *Knime* sistemose. *ClowdFlows* sistemoje kol kas nėra įgyvendinti asociatyvių taisyklių sudarymo algoritmai. *Taverna* sistemoje nėra nė vieno duomenų tyrybos metodo, tačiau yra galimybė prijungti kitur sukurtas duomenų tyrybos paslaugas. DAME yra įgyvendinti keli klasifikavimo ir grupavimo metodai.

Pagal atliktus vertinimus didžiausiais balais įvertintos *Knime* ir *ClowdFlows* sistemos (10 iš 12 galimų). *ClowdFlows* sistemos pranašumas yra tas, kad ji yra naudojama ir valdoma interneto naršykle. Be to ji intensyviai tebevystoma, todėl ateityje tikėtinas dar didesnis funkcionalumas.

Išvados

Pastaruoju metu nemaža dalis programinės įrangos kuriama į paslaugas orientuotos architektūros pagrindu. Ne išimtis ir duomenų tyrybos sistemos. Kelių populiarių duomenų tyrybos sistemų – *Weka* ir *Orange* pagrindu yra sukurtos saityno paslaugomis pagrįstos sistemos – *Weka4WS* ir *Orange4WS*. Dar viena populiari sistema *Knime* yra papildyta galimybe prijungti kitų kūrėjų sukurtas (išorines) saityno paslaugas. *ClowdFlows* ir DAME sistemos iš kitų išsiskiria tuo, kad sistemų nereikia įdiegti į naudotojo kompiuterį, joms naudoti ir valdyti pakanka interneto naršyklės. *Taverna* sistemoje nėra įgyvendintų duomenų tyrybos metodų, tačiau yra galimybė nesudėtingai prijungti išorines tiek SOAP, tiek *RESTful* saityno paslaugas. Visos tirtos sistemos yra atvirojo kodo, visose, išskyrus DAME sistemą, yra įgyvendinta darbų sekų konstravimo galimybė.

Darbe parinkti kriterijai, pagal kuriuos palygintos kelios saityno paslaugomis pagrįstos sistemos. Lyginamoji analizė parodė, kad pagal vertinamus kriterijus geriausiai įvertintos *Knime* ir *ClowdFlows* sistemos. Atliktos lyginamosios analizės rezultatai bus panaudoti kuriant naują duomenų tyrybos sistemą, pagrįstą saityno paslaugomis.

Padėka. Šis tyrimas atliktas Europos socialinio fondo finansuojamo projekto „Paslaugų interneto technologijų kūrimo ir panaudojimo našių skaičiavimų platformose teoriniai ir inžineriniai aspektai“ (Nr. VP1-3.1-ŠMM-08-K-01-010) lėšomis.

LITERATŪRA

BACHE, K.; LICHMAN, M. (2013). *UCI Machine Learning Repository.* University of California, Irvine, School of Information and Computer Sciences.

BERTHOLD, M.; CEBRON, N.; DILL, F.; KOTTER, T.; MEINL, T. (2007). *KNIME: The Konstanz Information Miner.* Studies in Classification, Data Analysis, and Knowledge Organization (GfKL 2007). Springer.

BHAGAT, J.; TANOH, F.; NZUOBONTANE, E.; LAURENT, T.; ORLOWSKI, J.; ROOS, M. (2010). BioCatalogue: a universal catalogue of web services for the life sciences. *Nucleic Acids Research*, W689–W694.

BOINC (2013) [interaktyvus] [žiūrėta 2013 m. liepos 8 d.]. Prieiga per internetą: <http://boinc.berkeley.edu>.

BOINCstats (2013) [interaktyvus] [žiūrėta 2013 m. liepos 8 d.]. Prieiga per internetą: <http://boincstats.com/en/stats/projectStatsInfo>.

BRESCIA, M.; LONGO, G.; CASTELLANI, M.; CAVUOTI, S.; D‘ABRUSCO, R.; LAURINO, O. (2012). DAME: A Distributed Web Based Framework for Knowledge Discovery in Databases. *Memorie della Societa Astronomica Italiana Supplement,* vol. 19, p. 324–329.

CONGIUSTA, A.; TALIA, D.; TRUNFIO, P. (2008). Service-oriented middleware for distributed data mining on the grid. *Journal of Parallel and Distributed Computing*, vol. 68(1), p. 3–15.

ČURČIN, V.; GHANEM, M.; GUO, Y.; KÖHLER, M.; ROWE, A.; SYED, J. (2002). Discovery Net: towards a grid of knowledge discovery. In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM Press, p. 658–663.

CURK, T.; DEMŠAR, J.; XU, Q.; LEBAN, G.; PETROVIČ, U.; BRATKO, I. (2005). Microarray data mining with visual programming. *Bioinformatics*, vol. 21(3), p. 396–398.

DAME project (2010) [interaktyvus] [žiūrėta 2013 m. liepos 8 d.]. Prieiga per internetą: <http://dame.dsf.unina.it>.

DE ROURE, D.; GOBLE, C.; STEVENS, R. (2008). The Design and Realisation of the myExperiment Virtual Research Environment for Social Sharing of Workflows. *Future Generation Computer Systems*, vol. 25, p. 561–567.

DZEMYDA, G.; MARCINKEVIČIUS, V.; MEDVEDEV, V. (2011). Large-Scale Multidimensional Data Visualization: A Web Service for Data Mining. ServiceWave 2011. *Lecture Notes in Computer Science*. Springer, p. 14–25.

Einstein@Home (2013). [interaktyvus] [žiūrėta 2013 m. liepos 8 d.]. Prieiga per internetą: <http://einstein.phys.uwm.edu/>.

FAEHIM (2005). *Federated Analysis Environment for Heterogeneous Intelligent Mining* [interaktyvus] [žiūrėta 2013 m. liepos 8 d.]. Prieiga per internetą: <http://users.cs.cf.ac.uk/Ali.Shaikhali/faehim/index.htm>.

FOSTER, I.; KESSELMAN, C. (2004). *The Grid 2*. University of Southern California.

Globus Toolkit (2013) [interaktyvus] [žiūrėta 2013 m. liepos 8 d.]. Prieiga per internetą: <http://www.globus.org/toolkit>.

GridWay (2013) [interaktyvus] [žiūrėta 2013 m. liepos 8 d.]. Prieiga per internetą: <http://www.gridway.org/doku.php>.

Hadoop (2012). Apache Hadoop [interaktyvus] [žiūrėta 2013 m. liepos 8 d.]. Prieiga per internetą: <http://hadoop.apache.org/>.

HALL, M.; FRANK, E.; HOLMES, G.; PFAHRINGER, B.; REUTEMANN, P.; WITTEN, I. H. (2009). The WEKA Data Mining Software: An Update. *SIGKDD Explorations,* vol. 11(1).

HMIDA, M.; SLIMANI, Y. (2010). Meta-learning in grid-based data mining systems. *International Journal of Communication Networks and Distributed Systems*, vol. 5(3), p. 214–228.

IBM Developers Works (2013) [interaktyvus] [žiūrėta 2013 m. liepos 8 d.]. Prieiga per internetą: <http://www.ibm.com/developerworks/webservices/newto/>.

JADHAV, S. (2009). *Advanced Computer Architecture and Computing*. Technical Publications Pune.

KNELL, R. J. (2013). *Introductory R: A Beginner's Guide to Data Visualisation and Analysis using R.*

KRANJC, K.; PODPECAN, V.; LAVRAC, N. (2012). ClowdFlows: A Cloud Based Scientific Workflow Platform. *Proceedings of European Conference on Machine Learning and Knowledge Discovery in Databases*, ECML PKDD 2012. Springer, p. 816–819.

KREGER, H. (2001). *Web Services Conceptual Architecture (WSCA 1.0)*. IBM.

Mahout (2011). Apache Mahout [interaktyvus] [žiūrėta 2013 m. liepos 8 d.]. Prieiga per internetą: <http://mahout.apache.org/>.

MIERSWA, I.; WURST, M.; KLINKENBERG, R.; SCHOLZ, M.; EULER, T. (2006). YALE: Rapid Prototyping for Complex Data Mining Tasks. In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, p. 935–940.

MISSIER, P.; SOILAND-REYES, S.; OWEN, S.; TAN, W.; NENADIC, A.; DUNLOP, I. (2010). Taverna, reloaded. In *Proceedings of 22nd International Conference on Scientific and Statistical Database Management*, SSDBM 2010. Heidelberg, Germany.

myGrid (2008) [interaktyvus] [žiūrėta 2013 m. liepos 8 d.]. Prieiga per internetą: <http://www.mygrid.org.uk/>.

PAPAZOGLOU, M. (2003). Service-oriented computing: concepts, characteristics and directions. Web Information Systems Engineering. WISE 2003. In: *Proceedings of the Fourth International Conference on*, p. 3–12.

PODPECAN, V.; ZEMENOVA, M.; LAVRAC, N. (2012). Orange4WS Environment for Service-Oriented Data Mining. *The Computer Journal*, vol. 55(1), p. 82–98.

SETI@Home. (2013) [interaktyvus] [žiūrėta 2013 m. liepos 8 d.]. Prieiga per internetą: <http://setiathome.berkeley.edu/>.

STANKOVSKI, V.; SWAIN, M.; KRAVTSOV, V.; NIESSEN, T.; WEGENER, D., KINDERMANN, J. (2008). Grid-enabling data mining applications with DataMiningGrid: An architectural perspective. *Elsevier, Future Generation Computer Systems*, vol. 4(4), p. 259–279.

TALIA, D.; TRUNFIO, P. (2012). *Service-oriented distributed knowledge discovery*. Chapman and Hall/CRC.

TALIA, D.; TRUNFIO, P.; VERTA, O. (2008). The Weka4WS framework for distributed data mining in service-oriented Grids. *Concurrency and Computation: Practice and Experience*, vol. 20(16), p. 1933–1951.

WHIE, T. (2012). *Hadoop: The Definitive Guide.* Third Edition. Sebastopol: O'Reilly Media, Inc.

DATA MINING SYSTEMS, BASED ON WEB SERVICES

Olga Kurasova, Virginijus Marcinkevičius, Viktor Medvedev, Aurimas Rapečka

S u m m a r y

In the paper, data mining systems, based on web services, are analysed. The main notation related with web services is described. The possibilities of distributed data mining and their implementation tools – Grid, Hadoop are introduced. An analytical review of the data mining systems, based on web services, is provided. Some comparison criteria are selected. According to the criteria, a comparative analysis of the popular data mining systems, based on web services, is made. The paper illustrates, which systems are best for evaluating and which do not satisfy most of the criteria.

Duomenų apdorojimas ir vizualizavimas mobiliuosiuose įrenginiuose

Kristina Lapin

Vilniaus universiteto Matematikos ir informatikos fakulteto docentė, daktarė
Vilnius University, Institute of Mathematics and Informatics, Assoc. Professor, Doctor
Didlaukio g. 47, LT-08303 Vilnius
El. paštas: Kristina.Lapin@mif.vu.lt

Sigitas Dapkūnas

Vilniaus universiteto Matematikos ir informatikos fakulteto docentas, daktaras
Vilnius University, Institute of Mathematics and Informatics, Assoc. Professor, Doctor
Didlaukio g. 47, LT-08303 Vilnius
El. paštas: Sigitas.Dapkunas@mif.vu.lt

Straipsnyje nagrinėjami duomenų apdorojimo mobiliuosiuose įrenginiuose bei klasikiniai ir neseniai sukurti vizualizavimo metodai, tinkantys vaizduoti duomenis mažame ekrane. Analizuojami mobiliųjų įrenginių apdorojimo ir duomenų pateikimo ribojimai, į kuriuos atsižvelgiama kuriant programas visur esančių skaičiavimų aplinkose. Apžvalginio tyrimo rezultatus planuojama pritaikyti kuriant saityno paslaugos vizualizavimo modelį paslaugų architektūros stiliaus sistemoje.

Įvadas

Žmonės nuo seno naudoja vizualizavimo priemones – žemėlapius, diagramas ar schemas, siekdami geriau suvokti problemą ir greičiau ją išspręsti. Nuolatinis kompiuterių galimybių augimas sudaro prielaidas kurti patogesnius vizualizavimo būdus. Populiarėjant mobiliesiems įrenginiams ir mažėjant jų kainai, plečiasi ir jų panaudojimo sritis. Vizualizavimo sprendimų adaptavimas mobiliesiems įrenginiams leidžia naudotojui analizuoti duomenis patogioje vietoje ir patogiu laiku. Į mobiliuosius telefonus diegiama vis daugiau ir įvairesnės techninės ir programinės įrangos (pavyzdžiui, skaitmeninės kameros, GPS, *Bluetooth*). Mobiliųjų telefonų paslaugos šiuo metu neapsiriboja vien balso pokalbiais ar trumpųjų žinučių siuntimu. Naujų programų ir paslaugų mobiliesiems įrenginiams atsiradimas – viena iš sparčiausiai besivystančių sričių. Aktyviai kuriamos paslaugos medicinos srityje.

Mobilieji įrenginiai visur esančių skaičiavimų aplinkoje apdoroja ir pateikia naudotojui aplinkoje esančių jutiklių duomenis. Šiose aplinkose sąveikauja lokalios sistemos ir konkrečiu momentu prie šios aplinkos prisijungę įrenginiai. Veikiančios programos turi dinamiškai prisitaikyti prie nuolat kintančios prisijungusių įrenginių konfigūracijos (Hurdin, Tigli, Lavirotte et al., 2008).

Visur esančių skaičiavimų aplinkos yra kuriamos įvairių architektūrų pagrindu. Tradicinėse architektūrose adaptavimo galimybės yra numatomos projektavimo metu. Pagrindinis iššūkis yra sistemos lankstumas ir operabilumas, gebėjimas veikimo metu keisti savo komponentų realizacijas. Šį reikalavimą tenkina paslaugų stiliaus architektūros.

Straipsnyje nagrinėjamos problemos, kylančios kuriant visur esančių skaičiavimų aplinkos sąveiką su mobiliuosiuose įrenginiuose veikiančiomis programomis bei vizualizuojant lokalių jutiklių duomenis mažuose mobiliųjų įrenginių ekranuose*. Straipsnio tikslas yra apžvelgti klasikinius ir naujai kuriamus duomenų apdorojimo ir vizualizavimo mobiliuosiuose įrenginiuose metodus, siekiant ištirti šios srities problemas. Prieigai prie interneto teikiamų paslaugų

* Šis tyrimas atliktas Europos socialinio fondo finansuojamo projekto „Paslaugų interneto technologijų kūrimo ir panaudojimo našių skaičiavimų platformose teoriniai ir inžineriniai aspektai“ (Nr. VP1-3.1-ŠMM-08-K-01-010) lėšomis.

per mobiliuosius įrenginius realizuoti reikalingi specifiniai metodai, atsižvelgiant į šių įrenginių ribojimus. Jie apžvelgiami pirmame skyriuje. Antrame skyriuje yra nagrinėjami duomenų apdorojimo ypatumai mobiliuosiuose įrenginiuose, trečiame – pasirinkti vizualizavimo būdai, gebantys kompaktiškai pateikti didelius duomenų kiekius mažame ekrane.

1. Duomenų apdorojimas mobiliuose įrenginiuose

Dabartinių mobiliųjų įrenginių galimybės vis dar išlieka ribotos ir rimtesniam duomenų apdorojimui reikalingi papildomi ištekliai – daugiau atminties, spartesnis procesorius. Gamintojų įvairovė, skirtingos programinės įrangos platformos, skirtingi ekrano dydžiai, prieinami tinklai ir daugybė kitų veiksnių sukelia sunkumų realizuojant paslaugas mobiliesiems telefonams. Tokių programų kūrėjai dažnai kuria programas, kurios bus tinkamos vieniems įrenginiams, bet netiks kitiems. Kūrėjai susiduria su tokiomis problemomis (van Gurp, 2006):

- programinė įranga tinka daug mažesniam įrenginių kiekiui, negu jų yra naudojama apskritai; kuriant programinę įrangą specifinėms įrenginių funkcijoms, galimybė pritaikyti ją kitiems įrenginiams dar labiau sumažėja;
- paprastai įrenginiai rinkoje yra pakankamai trumpai ir po keleto mėnesių arba geresniu atveju – metų juos pakeičia nauji įrenginiai;
- kuriant programinę įrangą panašių įrenginių grupei, ją būtina testuoti kiekvienam grupės įrenginiui;
- programinę įrangą sunku perkelti į kitus įrenginius.

Siekiant išspręsti šias problemas pastaruoju metu aktyviai plėtojamos paslaugų architektūros stiliaus sistemos. SearchSOA.com tinklalapio vykdytos apklausos duomenimis, 2011–2012 metais tik 21 proc. apklaustųjų teigė, kad programose mobiliesiems įrenginiams naudoja paslaugų architektūrą, bet 51,6 proc. sakė, kad tai ketina daryti artimiausioje ateityje. 44 proc. teigė, kad per kitus dvejus metus bus labiau finansuojamos programos mobiliesiems įrenginiams, kurios bus integruotos į jų organizacijoje naudojamą programinę įrangą (Mann, 2013). Tokios firmos kaip IBM, *Nokia*, *Microsoft* jau kuria saityno paslaugų programinę įrangą mobiliesiems įrenginiams (Netchetoi, 2008). Yra sukurta nemažai tam skirtų programų kūrimo priemonių.

Paslaugų, teikiamų mobiliaisiais telefonais, realizavimo galimybės yra įvairios. *Nokia* tyrimo centro mokslininkai išskiria tokias alternatyvas (van Gurp, 2006):

- paslaugos programa kuriama naudojant SIM-ATK ir diegiama telefone kaip papildoma telefono programinė įranga; šiuo atveju paslaugos programinė įranga pateikiama kartu su SIM kortele, įsigyjant telefoną;
- paslauga pasiekiama naudojant telefone veikiančią interneto naršyklę;
- galimybė kurti paslaugos programas naudojant telefono programinės įrangos platformą ir jas įdiegti telefone.

Dėl gana riboto procesoriaus, ribotos mobiliųjų įrenginių atminties, duomenų perdavimo greičio, maitinimo apribojimų paslaugų architektūros stiliaus sistemos mobiliesiems įrenginiams skiriasi nuo paslaugų architektūros stiliaus sistemų, naudojamų kompiuteriuose. Šiuo metu paslaugų architektūros stiliai mobiliesiems įrenginiams yra aktyviai kuriami. Yra pateikiama nemažai įvairių pasiūlymų, kaip tai turėtų būti realizuota. Aktyviai dirbama ir standartizuojant įvairius sprendimus. Standartizavimui yra svarbus Atviras mobilusis aljansas (angl. *Open Mobile Alliance* – OMA) – organizacija, kurianti atvirus standartus. OMA priklauso nemažai žinomų firmų – mobiliųjų įrenginių gamintojų, mobiliųjų operatorių, programinės įrangos gamintojų ir platintojų. Naudojant paslaugų architektūrą mobiliųjų įrenginių programinei įrangai svarbu (Netchetoi, 2008):

- minimizuoti perduodamų duomenų ir duomenų, saugomų mobiliajame įrenginyje, kiekius, išnaudojant žinias apie verslo procesus ir duomenų naudojimo statistiką;

- duomenų siuntimui naudoti kaip galima labiau suspaustą XML duomenų formatą;
- kuo labiau sumažinti informacijos kiekį SOAP pranešimuose;
- proaktyviai gauti duomenis iš serverio, atsižvelgiant į kliento užimtumo laiką;
- bendrauti asinchroniškai, tuo užtikrinant galimybę programoms veikti ir esant neprisijungus prie serverio.

Apžvelgus literatūrą buvo nustatyti trys paslaugų architektūros mobiliesiems įrenginiams problemų sprendimo būdai.

Gana dažnai pripažįstama, kad SOAP XML yra daugžodiškas. Mobiliesiems įrenginiams, kurių ištekliai ypač riboti, siūloma XML failus glaudinti (pakuoti), taip sutrumpinant perduodamus pranešimus. Siūloma naudoti duomenų glaudinimą su praradimais (Natchetoi, 2008). Toks būdas leidžia labiau suglaudinti duomenis, perduodant ne visą originalų pranešimą, kai kurią informaciją praleidžiant. Taip suglaudinti duomenys ne visi bus atkurti. Todėl šį būdą galima naudoti, kai kuri nors informacija gavėjui nėra reikalinga arba šios informacijos panaudojimo tikimybė yra maža. Naudojant šį metodą svarbu įsitikinti, kad informacijos neperdavimas nesugriaus programų funkcionalumo.

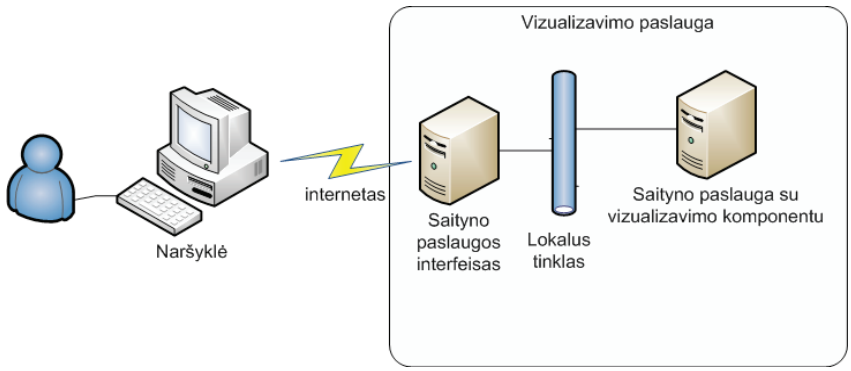
Paprastai į serverį verslo objektai įkeliami kiek galima universalesni, atliekantys kuo daugiau funkcijų. Tai padidina perduodamos informacijos kiekį, o tai mobiliesiems įrenginiams yra nepageidaujama. Siūlomos priemonės, leidžiančios automatiškai sumažinti verslo objektus aprašančių duomenų kiekį, paliekant tik duomenis, reikalingus klientams. Šiuo atveju perduodama informacija glaudinama, naudojant glaudinimą be praradimų (Natchetoi, 2008).

Siūlomi nauji pasaulinio saityno protokolai, kurie paslaugų architektūroje mobiliesiems įrenginiams pakeistų SOAP, dažniausiai naudojamą kompiuterių paslaugų architektūroje (Hamad, 2010). Tokių protokolų ypatumas – trumpesni pranešimai ir atsako laikas, dėl to jie labiau tinka mobiliesiems įrenginiams.

Mobiliųjų įrenginių procesoriaus galingumas riboja duomenų apdorojimo galimybes pačiame įrenginyje. Šiai problemai spręsti siūloma duomenis apdoroti nuotoliniame serveryje, tai yra pasinaudoti *vizualizacija per internetą* (angl. *visualization over the Internet*). Vizualizavimo per internetą sistemos architektūra iliustruojama 1 pav. Ją sudaro naršyklė, veikianti naudotojo įrenginyje, ir vizualizavimo paslaugos komponentai.

Naudojantis šia paslauga, siunčiama užklausa, nurodanti, kokio apdorojimo pageidaujama, ir patys duomenys, pavyzdžiui, paciento kardiograma. Patvirtinus duomenų priėmimą, jie apdorojami ir vizualizavimo komponentas suformuoja rodinį naudotojo įrenginio naršyklėje.

Centralizuota vizualizavimo saityno paslauga gali sumažinti daugelio medicinos programų kainą, nes beveik kiekviena jų turi savo vizualizavimo komponentą. Blazona ir Michailovic (2007) aprašytu atveju sistema veikia paslaugų architektūros stiliaus pagrindu. Saityno paslaugos sąsaja gauna naudotojo užklausą su pageidaujama duomenų apdorojimo procedūra, specifikuota XSD schemoje, ir pačius duomenis. Vizualizavimo komponentas įvykdo nustatytą procedūrą ir suformuoja vizualizuojamus objektus. Šis metodas



1 pav. ***Vizualizavimo per internetą sistemos komponentai. Adaptuota pagal (Blazona, Michailovic, 2007).***

leidžia tiems patiems duomenims taikyti kelias apdorojimo procedūras. Pavyzdžiui, gydytojas, pamatęs vienos užklausos rezultatus, gali paprašyti kito vizualizavimo būdo tiems patiems duomenims be papildomo sintimo.

Naudojant mobilųjį įrenginį kaip įvesties / išvesties įrenginį reikia išspręsti ribotą belaidžių tinklų pralaidumo problemą. Sistemoje *MobiMine* ši problema sprendžiama naudojant Fourier algoritmą sprendimo medžiams vaizduoti (Talia, Trunfio, 2010). Algoritmas, agreguodamas sprendimo medžius, sumažina siunčiamų duomenų apimtį.

Taigi, duomenų apdorojimą mobiliajame įrenginyje galima minimizuoti realizuojant vizualizavimo per internetą mechanizmą. Yra antra esminė vizualizavimo mobiliuosiuose prietaisuose problema – duomenų vaizdavimas mažame ekrane.

2. Vizualizavimo mobiliuosiuose įrenginiuose ypatumai

Vizualizavimo būdai, taikomi mobiliesiems įrenginiams, palyginti su naudojamais stacionariems kompiuteriams, turi esminių skirtumų (Chittaro, 2006), nes:

- ekranai yra nedideli, mažesnės raiškos ir turi mažiau spalvų;
- pločio ir aukščio santykis skiriasi nuo tradicinio 4 : 3;
- procesorius, atmintis, magistralės ir grafinės plokštės yra mažesnio galingumo;
- įvesties įranga nepatogi sudėtingiems uždaviniams;
- nauji įvesties būdai: pirštų gestai, nykščio įvestis, pieštuko įvestis;
- vizualizavimo priemonės, kaip antai grafinės bibliotekos, yra žemo lygmens ir ribotų funkcinių galimybių.

Dėl šių priežasčių stacionariųjų kompiuterių vizualizavimo būdai netinka mobiliesiems įrenginiams. Kai kurie šių ribojimų neišnyks ir ateityje, nes mobilieji įrenginiai turi išlikti nedideli.

Naudotojo judėjimas komplikuoja gebėjimą sutelkti dėmesį sąveikai su mobiliuoju įrenginiu. Sąveika su stacionariuoju kompiuteriu iš esmės skiriasi nuo sąveikos su mobiliaisiais įrenginiais, nes sąveika su stacionariuoju kompiuteriu yra pagrindinė naudotojo veikla, į kurią sutelktas visas dėmesys. Mobilieji įrenginiai paprastai nėra dėmesio centre: naudotojas kažką veikia, o įrenginys tik padeda jam atlikti pagrindinę veiklą. Pavyzdžiui, asmuo eina į teatrą svetimame mieste, retsykiais stebėdamas navigavimo programos nurodymus. Daugiausia mobilieji įrenginiai veikia pagrindinių veiklų fone ir yra naudojami trumpoms užduotims.

3. Klasikiniai vizualizavimo būdai, tinkantys mažiems ekranams

Populiarėjant paslaugoms, teikiamoms nešiojamųjų įrenginių naudotojams, duomenų vizualizavimas tampa aktyviu tyrimų objektu. Lietimui jautrus mobiliųjų įrenginių ekranas tarnauja tiek duomenims vaizduoti, tiek įvesti. Vaizdu gali būti manipuliuojama keliais pirštais vienu metu. Klasikiniai interaktyvieji vaizdavimo būdai buvo kuriami turint omenyje sąveiką su pele ar klaviatūra, kai konkrečiu momentu manipuliuojama vienu vaizdo fragmentu. Kelių pirštų sąveikai realizuoti reikalingi iš esmės kitokie metodai.

Dideliame ekrane talpinami tiek vizualizuojami duomenys, tiek sąveikos mygtukai. Lietimui jautriame ekrane mygtukai turi būti pakankamai dideli, kad būtų patogu juos nurodyti pirštu. Tačiau tuomet prarandama vieta patiems duomenims. Taigi, duomenų vizualizavimo būdai, sukurti dideliems stacionarių kompiuterių ekranams, turi būti pritaikomi kitam sąveikos tipui arba kuriami nauji.

Svarbu ne tik vaizduoti duomenis taip, kad jie būtų visi matomi, bet ir vizualiai juos susieti, palengvinant naudotojui suvokimą. Naudotojas pirštu liesdamas, spausdamas ar tempdamas vaizduojamus duomenis tikisi, kad jie sąveikaus pagal jo judesius bent jau panašiai, kaip sąveikauja realus objektas, kai jis yra čiupinėjamas.

3.1. Informacijos slinkties metodas

Informacijos slinkties metodas yra plačiai taikomas būdas pateikti daugiau informacijos,

nei jos telpa ekrane. Naudotojui pateikiama dalis informacijos ir pažymima, kad už ekrano ribų yra tęsinys. Tradiciškai tęsinį signalizuoja slinkties juosta, o pavyzdžiui, Metro stiliaus rodiniuose tai parodo už ekrano ribos užrašo arba paveikslo paslėptas kraštas.

Slinkimas gali būti horizontalus ir vertikalus. Slinkties metodas yra paprastas ir intuityvus. Jis tinka ir lietimui jautriam ekranui. Tačiau čia parodomas tik duomenų fragmentas, o ieškant dominančių duomenų reikia slankioti visą dokumentą. Duomenų kontekstą reikia įsiminti, kas didina naudotojo atminties apkrovimą.

3.2. Konteksto žemėlapis

Konteksto žemėlapyje (angl. *context map*) slinkimo metodas yra patobulintas skirtingo abstrakcijos lygmens rodiniais. Naudojant šį metodą, ekranas dalijamas į dvi dalis: duomenų erdvės apžvalgai ir jos fragmento detaliajam vaizdui. Apžvalgos lange pateikiamas visos ar beveik visos duomenų erdvės sumažintas vaizdas. Pasirinktas apžvalgos rodinyje duomenų fragmentas yra išdidinamas iki interpretuojamo detalaus vaizdo (2 pav.).

Apžvalgos rodinys gali būti kuriamas ne tik sumažinant rodomus objektus, bet ir grupuojant duomenis. Duomenų erdvė kategorizuojama, apžvalgoje rodomos kategorijos. Pasirinkta kategorija yra išskleidžiama. Jei ir apžvalga ar detalus vaizdas netelpa ekrane, navigacija vykdoma įprastu slinkties metodu.

2 pav. ***Apžvalgos rodinys dešiniajame apatiniame kampe pateikia visą Romos žemėlapį, jame pasirinkta erdvė rodoma detaliajame vaizde (vaizdas sukurtas naudojantis http://www.rome.info/map/)***

Konteksto žemėlapio pranašumas yra visos duomenų erdvės matomumas apžvalgos rodinyje. Naudotojui nereikia įsiminti detaliajame vaizde rodomų duomenų konteksto, nes jis yra matomas tame pačiame ekrane. Šio metodo trūkumas yra tas, kad skirtingi rodiniai yra vizualiai mažai susiję. Norėdamas suprasti, kuris apžvalgos erdvės fragmentas yra rodomas detaliajame vaizde, naudotojas turi šokinėti nuo vieno vaizdo prie kito. Atsiranda papildomas pažintinis apkrovimas. Nors lyginant su slinkties rodinio konteksto atsiminimu tai mažesnė problema, vis dėlto suvokimas nėra visiškai intuityvus. Šią problemą gali sušvelninti didinamojo stiklo technika (Ware, Lewis, 1995). Tuomet detalusis vaizdas pateikiamas apžvalgos rodinyje jo fragmentą didinant lyg pro didinamąjį stiklą. Tada aiškiai matyti, kuris duomenų erdvės fragmentas rodomas detaliajame vaizde. Detaliojo vaizdo fragmentas gali būti judinamas išryškinant vis kitų duomenų apžvalgos fragmentų detales.

Nors abiejų vaizdų sąsaja yra aiškiai parodyta, tačiau vis tiek šį rodinį sudaro du vaizdai. Žiūrėdamas į juos naudotojas šiek tiek dėmesio turi skirti jų interpretavimui ir tarpusavio susiejimui.

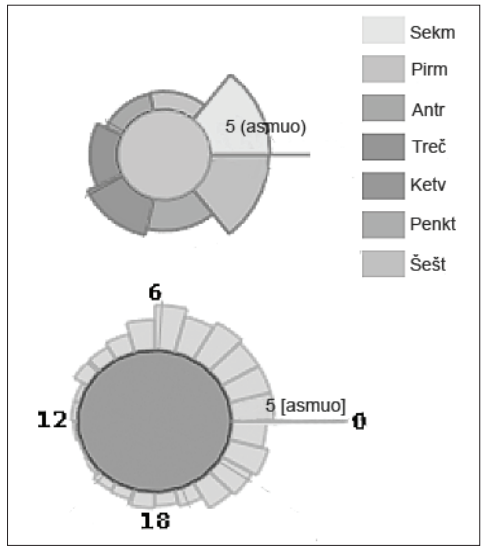
Konteksto žemėlapis tinka mažam ekranui, jei ekrane pakanka vietos šalia apžvalgos rodinio parodyti prasmingą detaliosios informacijos dalį. Tačiau gali būti sudėtinga sumažintame apžvalgos rodinyje pirštu nurodyti, kuri vieta turi būti parodyta detaliajame rodinyje.

3.3. Išdidinantis rodinys

Išdidinančiame rodinyje (angl. *focus-contex view*) visi duomenys rodomi viename ekrane su dviem židiniais (3 pav.). Pirmame židinyje detaliai rodomas išdidintas naudotojo dėmesio centre esantis objektas, o antrame židinyje apibendrintai pateikiami likę duomenys laipsniškai juos sumažinant. Išdidinančio meniu naudojimą sprendimų paramos sistemoje iliustruoja 3 pav. (Lapin, Rakovskaja, 2008).

Detalų vaizdą galima slankioti išdidinant pageidaujamą fragmentą. Šis metodas sujungia

3 p a v. ***Išdidinantis rodinys vaizduoja kintančio dydžio meniu sprendimų paramos sistemoje***



4 p a v. ***Elgsenos žiedai: viršuje parodytas savaitės žiedas, kuriame spinduliai pavaizduoja savaitės dienas, o spindulių ilgis atvaizduoja skambučių skaičių. Apačioje vaizduojamas skambučių skaičius paros metu. Rodiniai sukurti remiantis (Shen, Ma, 2008)***

slinkties ir konteksto žemėlapio metodų pranašumus. Nereikia susieti apžvalgos ir detaliojo vaizdo, nes aiškiai parodoma, kuri apžvalgos vieta išdidinta. Tačiau ir šis metodas turi trūkumų. Mažiausias judesys paveikia vaizdą, todėl judesiai turi būti labai tikslūs, o naudotojo ranka išlavinta.

Mažam ekranui šis rodinys taip pat tinka, jei detaliojo vaizdo erdvės užtenka parodyti prasmingą informacijos dalį. Sąveikauti pirštu šiuo būdu yra patogiau nei konteksto žemėlapyje, nes manipuliuojama didesniais objektais detaliajame rodinyje.

4. Šiuolaikinės vizualizacijos mažiems ekranams

4.1. Elgsenos žiedai

Elgsenos žiedai (angl. *behavior rings*) vaizduoja reikšmių laikinį kitimą. Pirmą kartą šį vaizdavimą panaudojo Florence Nightingale*, pademonstravusi, kad Krymo kare daugiau kareivių mirė nuo ligų ir blogų gyvenimo sąlygų nei nuo žaizdų.

Elgsenos žiedai taip pat gali vaizduoti socialinio tinklo mazgus (Shen, Ma, 2008). Savaitės žiedai turi 7 spindulius, vaizduojančius savaitės dienas, dienos žiedas turi atitinkamai 24 spindulius (4 pav.). Spindulio ilgis atitinka vizualizuojamo parametro reikšmę. Savaitės elgsenos žiedas paveiksle rodo, kad stebimas asmuo daugiausia skambina šeštadieniais ir sekmadieniais. Dienos elgsenos žiede matyti, kad daugiausia skambučių būna vakarais ir naktimis.

Elgsenos žiedai leidžia pastebėti pasikartojantį dėsningumą. Šis vizualizavimo metodas buvo sukurtas XIX amžiuje, tačiau išpopuliarėjo pastaruoju metu, vizualizuojant socialinių tinklų naudotojų elgseną.

Šis vaizdavimo būdas yra kompaktiškas, tinka rodyti duomenis ir sąveikauti lietimui jautriuose ekranuose.

* Florence Nightingale and the Crimean War. *Understanding uncertainty.* Kembridžo universiteto Statistikos laboratorijos tinklaraštis [žiūrėta 2013 m. gegužės 26 d.]. Prieiga per internetą: <http://understandinguncertainty.org/node/204>.

4.2. Duomenų kitimo ir tarpusavio ryšių vizualizavimas

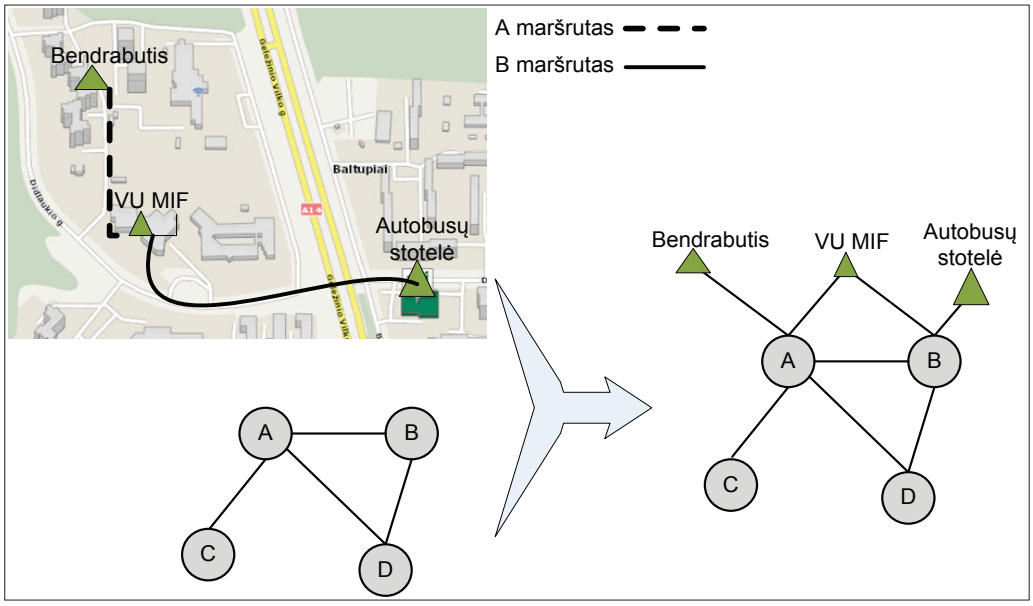
Mobilieji įrenginiai gali fiksuoti erdvinius ir socialinius duomenis. Erdvines koordinates fiksuoja globali pozicionavimo sistema. Socialiniai duomenys apima kaimynystėje esančius įrenginius, sąveikaujančius per *Bluetooth* ryšį, bei skambučių ir žinučių adresatų duomenis.

Klasikinės vizualizavimo priemonės rodo erdvinius duomenis vietovės žemėlapyje. Socialinių tinklų informacija paprastai vizualizuojama grafo mazgais ir briaunomis. Norint analizuoti šiuos duomenis, naudotojui tenka perjunginėti rodinius, ir tai nėra patogu. Projekte *MobiVis* integruojami erdvinis ir socialinis rodiniai viename ekrane (Shen, Ma, 2008):

1) Socialiniai duomenys – skambučiai, žinutės ir kaimyniniai įrenginiai – vaizduojami neorientuotu grafu, kuriame mazgas pažymi asmenį, o briauna – socialinį ryšį: skambutį, žinutę ar kaimyninių įrenginių sąveiką, reiškiančią buvimą toje pačioje vietoje. Briaunų duomenys kinta laike.
2) Erdviniai duomenys apibrėžiami trejetu: $ED = \{(a_i, v_j, t): a_i \in A, v_j \in V\}$, čia a_i žymi asmenį, esantį vietovėje v_j laiku t. Erdviniai duomenys taip pat kinta laike ir yra vaizduojami neorientuoto grafo pavidalu.

Siekiant palengvinti erdvinių ir socialinių ryšių suvokimą, socialinis ir erdvinis grafai yra integruojami viename heterogeniniame grafe (5 pav.). Erdvinis rodinys viršutiniame kairiajame kampe rodo, kad asmuo A judėjo nuo bendrabučio iki Matematikos ir informatikos fakulteto (VU MIF), o asmuo B – nuo autobusų stotelės iki fakulteto. Socialinio tinklo grafas vaizduoja, kad asmuo A skambino ar rašė žinutes asmenims B, C ir D, o B bendravo tik su A ir D. Integruotas vaizdas parodo socialinius ryšius ir buvimo vietą tame pačiame rodinyje.

Nurodžius dominančius objektus ir asmenis, stebėti jų erdvinius ir socialinius ryšius yra paprasta. Šis vizualizavimas leidžia patogiai sąveikauti lietimui jautriame ekrane, nes asmenis ir erdvės objektus vaizduojantys simboliai yra pakankamai dideli, o abstrahuotas vaizdas įskaitomas.



5 p a v. ***Erdvinis rodinys kairiajame viršutiniame kampe, integruotas su socialiniu tinklu apačioje. Rezultatas: heterogeninis grafas dešinėje, kuriame trikampiai pažymi buvimo vietą, o skrituliai – asmenis (adaptuota pagal (Shen, Ma, 2008))***

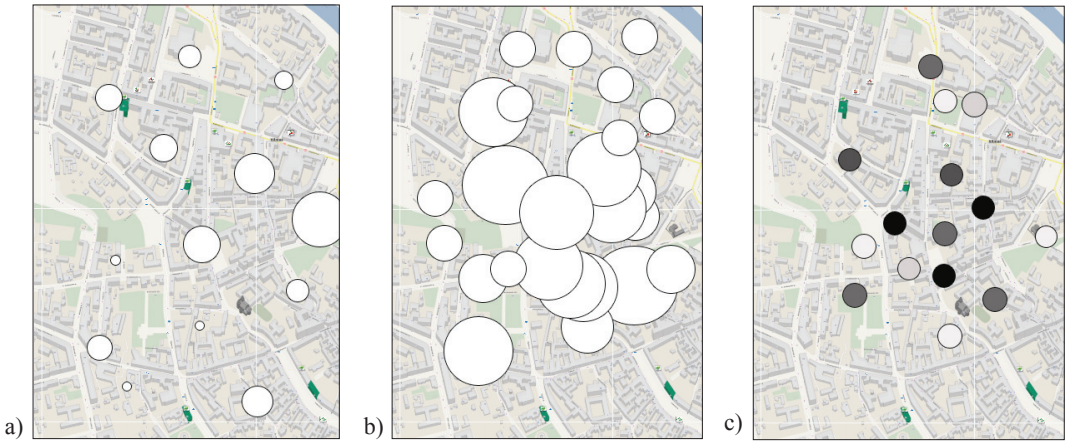
4.3. Ekrano užgriozdinimą mažinantis vizualizavimas

Visur esančių skaičiavimų aplinkoje veikiantys jutikliai siunčia duomenų srautus, kuriuos gali apdoroti ten esantys mobilieji įrenginiai. Tokia infrastruktūra leidžia naudotojui gauti aplinkos teikiamas paslaugas. Gaunamo duomenų srauto vizualizavimas, be mažo ekrano, kelia ir tokių iššūkių kaip nuolatiniai skaičiavimai ir greitai kintantis rodinys. Ekranas greitai užgriozdinamas ir tampa neįskaitomas. Siekiant palengvinti duomenų analizę sukurtas rodinių adaptavimo algoritmas, mažinantis ekrano užgriozdinimą (Gillick, AlTaiar, Krishnaswamy, Liono, Nicoloudis, Sinha, Zaslavsky, 2010). Šiame algoritme analizuojamas taksi automobilių, susitelkusių tam tikruose miesto rajonuose, skaičius (6 pav.).

Pradžioje konkrečiame rajone esančių taksi automobilių skaičių vaizduoja atitinkamo dydžio skrituliai (6 pav., a). Pasibaigus tame rajone vykstantiems renginiams klientai kviečia taksi ir jų skaičius greitai didėja (6 pav., b). Skrituliai uždengia vis didesnę miesto plano dalį ir susikloja. Vaizdas tampa sunkiai įžiūrimas. Užgriozdinimo matavimo algoritmas nuolat skaičiuoja uždengtos ekrano dalies dydį, vaizduojamų objektų skaičių bei susiklojančių ir susikertančių objektų skaičių. Kai stebimi parametrai viršija naudotojo nustatytus slenksčius, įvykdomas pirmo lygio adaptavimas, kuriame dinamiškai sumažinamas rodomų objektų mastelis. Tuomet visų objektų dydžiai proporcingai mažinami su sąlyga, kad mažiausias elementas negali būti mažesnis už naudotojo nustatytą dydį.

Jei po pirmo adaptavimo užgriozdinimas išlieka arba mažiausias elementas tapo mažesnis už nustatytą slenkstį, vizualizavimo algoritmas perjungia rodinį į antrą adaptavimo lygmenį – šešėliavimą. Šiame lygmenyje visi objektai rodomi vienodo dydžio skrituliais. Reikšmių skirtumą parodo atspalvio intensyvumas. Tamsiausias atitinka didžiausią reikšmę, šviesiausias – mažiausią (6 pav., c). Jei antro lygmens adaptavimo objektai ir toliau viršija nustatytus užgriozdinimo slenksčius, vizualizavimo programa persijungia į aukščiausią adaptavimo lygmenį – išrinkimą. Trečiame adaptavimo lygmenyje vaizdavimui parenkami tik konkrečiu momentu kintantys duomenų objektai.

Rodinio adaptavimas leidžia panaikinti suvokimo trukdžius ir interpretuoti vaizdą nepaisant nuolat kintamo duomenų kiekio. Tačiau vaizdo adaptavimas reiškia, kad ta pati duomenų savy-



6 pav. ***Ekrano užgriozdinimą mažinantis adaptavimas: a) neužgriozdintas vaizdas, padengta nedidelė vaizdo dalis, objektai nesusikloja; b) daugėjant vaizduojamų objektų, ekrano padengimas didėja, objektai susikloja; c) vaizdas įvykus šešėliavimui, užgriozdinimas sumažintas, einamoji būsena perteikiama atspalviais. Adaptuota pagal (Gillick, AlTaiar, Krishnaswamy, et al., 2010)***

bė yra rodoma skirtingai įvairiuose adaptavimo lygmenyse. Pavyzdžiui, tas pats taksi skaičius pirmame ir antrame adaptavimo lygmenyse yra vaizduojamas skirtingo dydžio skrituliais. Šis rodinys yra patogesnis suvokti duomenų santykį nei absoliučius skaičius. Tai yra intuityvus palyginimas, kuriame rajone taksi automobilių yra daugiau, tačiau norint suprasti, kiek, kad ir apytiksliai, jų ten yra, reikia mastelio. O tai reiškia papildomą pažintinį apkrovimą ir reikalauja papildomos ekrano vietos. Trečio lygio adaptavimas – naudojant atspalvius – taip pat patogesnis lyginti nei suvokti kiekius. Ketvirtame lygmenyje matomas tik duomenų kitimas, ir tai reiškia, kad interpretuojant vaizdą reikės dar turėti omenyje, kad čia nekintami duomenys nėra rodomi, bet jie egzistuoja. Juos interpretuojant reikės prisiminti ankstesnę būseną ir lyginti su masteliu. Tačiau nepaisant nurodytų trūkumų, be adaptavimo, vaizdas būtų visai neįskaitomas. Adaptavimas, nors ir reikalauja papildomų mintinių pastangų, tačiau vaizdo interpretavimą padaro įmanomą.

Išvados

Šio straipsnio tikslas – išanalizuoti duomenų apdorojimo ir vizualizavimo metodų mobiliuosiuose įrenginiuose, veikiančiuose visur esančių skaičiavimų kontekste, problemas. Visur esančių skaičiavimų aplinkos suteikia naudotojams galimybę naudotis aplinkoje esančiomis paslaugomis tiesiogiai arba per naudotojų mobiliuosius įrenginius.

Mobiliųjų įrenginių apdorojimo ir vizualizavimo ribojimai sukelia problemų, kurioms spręsti kuriami nauji duomenų glaudinimo ir vizualizavimo būdai. Išanalizavus glaudinimo metodus išskirti šie: visiškai atgaminami ir atgaminami su praradimais. Tačiau ir suglaudintų duomenų srautui vis tiek reikia pakankamai didelių skaičiavimų.

Ribotą mobiliųjų įrenginių skaičiavimų galingumą siūloma kompensuoti skaičiavimams naudojant saityno paslaugas. Naudotojo užklausos ir duomenys persiunčiami internetu galingesniam kompiuteriui, teikiančiam vizualizavimo paslaugą. Šios paslaugos rezultatas pateikiamas naudotojo įrenginyje. Tokiu būdu vizualizavimo per internetą metodas sumažina skaičiavimų apimtį mobiliajame įrenginyje, palikdamas jam tik užklausų formavimą ir duomenų vizualizavimą.

Duomenų srauto vizualizavimas turi pavaizduoti didelius duomenų kiekius mažame mobiliojo įrenginio ekrane. Klasikiniai metodai vizualizuoja duomenų fragmentą arba derina apžvalgos ir detaliojo vaizdo rodinius. Šie rodiniai gali būti pritaikomi ir mažiems ekranams. Tačiau sąveika lietimui jautriame ekrane gali būti nepatogi, kai duomenų objektu manipuliuojama apžvalgos rodinyje.

Mobiliuosiuose įrenginiuose yra naudojami gerai žinomi, bet retai naudojami vizualizavimo būdai, pavyzdžiui, elgsenos žiedai. Jie atskleidžia laike kintančių duomenų pasikartojantį dėsningumą kompaktiškame vaizde, tinkančiame tiek rodyti mažame ekrane, tiek manipuliuoti lietimui jautriame ekrane. Duomenų srautai, vaizduojami mažuose mobiliųjų įrenginių ekranuose, greitai juos užgriozdina ir sunkina suvokimą, todėl kuriami užgriozdinimą mažinantys adaptyvieji algoritmai.

LITERATŪRA

CHITTARO, Luka (2006). Visualizing information on mobile devices. *Computer*, vol. 39, no. 3, p. 40–45.

van GURP, Jilles; KARHINEN, Anssi; BOSCH, Jan (2006). Mobile Service Oriented Architectures (MOSOA). In: *Lecture Notes in Computer Science v. 4025* (6th IFIP WG 6.1 International Conference, DAIS 2006 Bologna, Italy, June 14–16, 2006 Proceedings), p. 1–15.

GILLICK, Brett; ALTAIAR, Hasnain; KRISHNASWAMY, Shonali; LIONO, Jonathan; NICOLOUDIS, Nicholas; SINHA, ABhijat; ZASLAVSKY, Arkady (2010). Clutter-Adaptive Visualization for Mobile Data Mining. In: *International Conference on Data Mining Workshops*, IEEE, p.1381–1384.

HAMAD, Hatem; SAAD, Motaz; ABED, Ramzi (2010). Performance Evaluation of RESTful Web

Services for Mobile Devices. *International Arab Journal of e-Technology*, vol. 1, no. 3, p. 72–78.

HURDIN, Vincent, TIGLI, Jean-Yves, LAVIROTTE, Stéphane, REY, Gaëtan, RIVEILL, Michel (2008). SLCA, Composite Services for Ubiquitous Computing. In: *Proceedings of the International Conference on Mobile Technology, Applications, and Systems*, *Mobility'08*, ACM New York [žiūrėta 2013 m. gegužės 13 d.]. Prieiga per internetą: <http://doi.acm.org/10.1145/1506270.1506284>.

LAPIN, Kristina; RAKOVSKAJA, Olga (2007) Combining the interaction styles to display complex data in decision-making system. In: *The Poster Proceedings of the 15th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, p. 9–12.

MANN, Stephanie (2013). Mobile app integration gets SO-ized: SOA goes mainstream 2012 [žiūrėta 2013 m. gegužės 9 d.]. Prieiga per internetą: <http://searchsoa.techtarget.com/feature/Mobile-application-integration-gets-SOA-ized-SOA-goes-mainstream-2012>.

NATCHETOI, Yuri; KAUFMAN, Viktor; SHAPIRO, Albina (2008). Service-Oriented Architecture for Mobile Applications. In: *Proceeding SAM '08 Proceedings of the 1st international workshop on Software architectures and mobility*, p. 27–32.

SHEN, Zeqian; MA, Kwan-Liu (2008). MobiVis: A Visualization System for Exploring Mobile Data. In: *IEEE Pacific Visualisation Symposium*, IEEE, Kyoto, Japan, p. 175–182.

SPENCE, Robert (2001). *Information Visualization*. Addison-Wesley.

TALIA, Domenico; TRUNFIO, Paolo (2010). Mobile Data Mining on Small Devices through Web Services. In: L. Yang, A. B. Waluyo, J. Ma, L. Tan, & B. Sri (Eds.). *Mobile Intelligence*. Hoboken, NJ, USA: John Wiley & Sons, Inc. Chapter 12.

WARE, Colin; LEWIS, Marlon (1995). The DragMag Image Magnifier. In: *Conference Companion on Human Factors in Computing Systems* (CHI ,95), I. Katz, R. Mack, and L. Marks (Eds.). ACM, New York, NY, USA, p. 407–408. DOI=10.1145/223355.223749 [žiūrėta 2013 m. gegužės 29 d.]. Prieiga per internetą: <http://doi.acm.org/10.1145/223355.223749>.

DATA PROCESSING AND VISUALIZATION ON MOBILE DEVICES

Kristina Lapin, Sigitas Dapkūnas

S u m m a r y

The paper surveys the data processing and visualization peculiarities in mobile devices. Developing the visualisation application to a mobile device a researcher has to consider their computational and size limitations. The mobile and traditional visualization methods for small screens are analysed. Researchers start to address the clutter problem showing the stream of data. The outcome of this research is planned to develop a model of visualisation of web service for a system based on the service-oriented architecture.

Duomenų tyrybos sistemų galimybių tyrimas įvairių apimčių duomenims analizuoti

Kotryna Paulauskienė

Vilniaus universiteto Matematikos ir informatikos instituto doktorantė
Vilnius University, Institute of Mathematics and Informatics, Doctoral student
Akademijos g. 4, LT-08663 Vilnius
El. paštas: kotryna.paulauskiene@mii.vu.lt

Olga Kurasova

Vilniaus universiteto Matematikos ir informatikos instituto vyresn. mokslo darbuotoja
Vilnius University, Institute of Mathematics and Informatics, Senior researcher
Akademijos g. 4, LT-08663 Vilnius
El. paštas: olga.kurasova@mii.vu.lt

Tobulėjant šiuolaikinėms informacinėms ir komunikacinėms technologijoms, sparčiai didėja apdorojamų ir saugomų duomenų kiekiai, todėl duomenų analizės uždavinys tampa vis sudėtingesnis, sunku daryti greitus, efektyvius ir teisingus sprendimus. Duomenų analizei dažnai pasitelkiama duomenų tyryba. Duomenų tyryba – tai procesas, kurio metu iš duomenų išgaunamos naudingos žinios. Duomenims apdoroti bei žinioms išgauti reikalingos duomenų tyrybos sistemos, leidžiančios apdoroti įvairios apimties duomenis. Tyrime siekiama nustatyti, kokios apimties duomenis per priimtiną laiką sugeba apdoroti populiariausios duomenų tyrybos sistemos. Nagrinėjamas ir lyginamas trijose atvirojo kodo duomenų tyrybos sistemose (WEKA, KNIME, ORANGE) įgyvendintų klasifikavimo ir klasterizavimo algoritmų skaičiavimo laikas, analizuojant skirtingos apimties duomenų aibes. Vertinant sistemas svarbus ne tik algoritmų skaičiavimo laikas, bet ir klasifikavimo bei klasterizavimo tikslumas, kurį pavyksta pasiekti per tą laiką, todėl straipsnyje pateikiamos ir eksperimentiniuose tyrimuose gauto tikslumo matų reikšmės.

Įvadas

Šiandieniame pasaulyje įvairiose srityse kaupiami dideli, nuolat augantys duomenų kiekiai. Šiuo metu net asmeniniai kompiuteriai leidžia saugoti tokius duomenų kiekius, kurių anksčiau nebuvo įmanoma saugoti dėl nepakankamos disko vietos. Duomenų vis daugėja, o santykinė dalis, kurią žmonės pajėgūs suprasti, grėsmingai mažėja (Witten, Frank, 2005). Analizuojant duomenis, dažnai pasitelkiama duomenų tyryba. Duomenų tyryba – tai procesas, kurio metu iš duomenų išgaunama informacija ir žinios, būtinos reikiamiems sprendimams priimti (Han, Kamber, 2006). Duomenų tyryba taikoma įvairiose srityse: versle ir komercijoje, inžinerijoje, telekomunikacijose, bankinėse ir draudimo sistemose, elektroninėje prekyboje, medicinoje ir kt. Duomenims apdoroti bei žinioms išgauti dažnai naudojamos duomenų tyrybos sistemos, leidžiančios apdoroti įvairios apimties duomenis. Kyla klausimas, kokie duomenys gali būti vadinami didelės apimties. Vienareikšmišką atsakymą į šį klausimą sunku rasti. Duomenys, kurie prieš kelerius metus buvo didelės apimties, atsiradus greitesniems duomenų apdorojimo įrenginiams ir metodams, tampa nedidelės apimties. Viena iš didelių apimčių duomenų apibrėžčių yra tokia: didelės apimties duomenimis galima laikyti tuos, su kuriais per priimtiną laiką nesusidoroja įprastos duomenų tyrybos sistemos, ir būtinos specialios, pritaikytos didelės apimties duomenims analizuoti, pavyzdžiui, pasitelkiant lygiagrečiuosius ir paskirstytuosius skaičiavimus bei debesų kompiuterijos technologijas. Priimtino laiko nustatymo uždavinys taip pat nėra elementarus. Tai priklauso nuo

sprendžiamo uždavinio specifikos ir norimo rezultatų tikslumo. Pavyzdžiui, analizuojant medicininius duomenis, labai svarbus tikslumas, todėl duomenų analizės rezultato tikslinga laukti kelias valandas ar net paras. Jei sprendžiamo uždavinio tikslumas nėra tiek svarbus, kiek rezultato radimo laikas, tik kelios sekundės gali būti laikomos priimtinu laiku.

Šio tyrimo objektas – įvairių apimčių duomenys ir duomenų tyrybos sistemos. Tyrimo tikslas – nustatyti, kokių apimčių duomenis per priimtiną laiką geba ištirti populiarios duomenų tyrybos sistemos, sprendžiant klasifikavimo ir klasterizavimo uždavinius. Tyrime taikoma informacijos paieškos, sisteminimo, analizės, lyginamosios analizės ir apibendrinimo metodika. Nagrinėjama klasifikavimo ir klasterizavimo algoritmų greitaveika naudojant skirtingos apimties duomenų aibes. Be sistemų skaičiavimo laiko, vertinami klasifikavimo ir klasterizavimo algoritmų tikslumo matai.

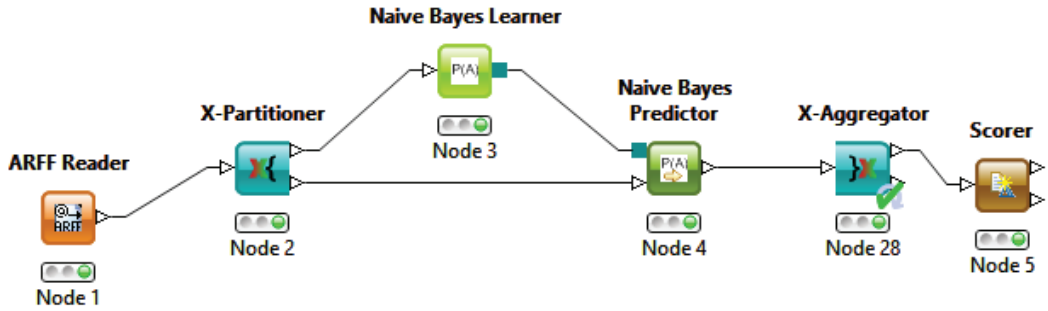
1. Duomenų tyrybos sistemos ir metodai

Straipsnyje nagrinėjamos ir lyginamos trys atvirojo kodo duomenų tyrybos sistemos:

- WEKA (Waikato Environment for Knowledge Analysis) (Hall et al.., 2009),
- KNIME (Konstanz Information Miner) (Berthold, 2007),
- ORANGE (Curk, 2005).

Tai vienos populiariausių duomenų tyrybos sistemų. Nors jos nėra pritaikytos didelės apimties duomenų apdorojimui ir analizei, jose įgyvendinti duomenų tyrybos metodai pajėgūs susidoroti su nemažomis duomenų aibėmis. Be to, šioms sistemoms naudoti nereikia specialių informatikos žinių. Greitai suvokiami jų naudojimo principai leido šioms sistemoms tapti populiariomis tarp įvairių sričių tyrėjų. Būtent dėl šių priežasčių minėtos sistemos pasirinktos tolesnei analizei. Šiose sistemose įgyvendintų klasifikavimo algoritmų gebėjimas teisingai klasifikuoti duomenis tiriamas darbe (Wahbeh et al., 2011), sistemų analizė atlikta darbe (Zupan, Demsar, 2008), tačiau ten nėra nustatyta, kokių apimčių duomenis sistemos pajėgios apdoroti ir analizuoti. Atvirojo kodo duomenų tyrybos sistemų taikymo sritys, vartotojų grupės, įgyvendinti algoritmai, vizualizavimo būdai ir kitos savybės vertinamos darbe (Madasamy, Tamilselvi, 2012), bet analizė naudojant įvairias duomenų aibes neatlikta. Autoriai (Chen, Williams, Xu, 2007) nurodo, kad WEKA, KNIME, ORANGE sistemos susidoroja su vidutinio dydžio duomenų aibėmis, tačiau ten nėra nurodoma, kokie duomenys vadinami vidutinio dydžio.

Tyrimuose naudotos šios sistemų versijos: WEKA 3.6.7, KNIME 2.7.3, ORANGE 2.6.1. Visose trijose sistemose yra įgyvendintas darbo eigos (angl. *workflows*) modulis, suteikiantis sistemoms patrauklumo. Vartotojas gali iš esamų mazgų (angl. *nodes*) sudaryti norimą schemą savo eksperimentams. Yra mazgų, skirtų įvairiems duomenų tyrybos algoritmams, pradiniam duomenų apdorojimui, rezultatų vizualizavimui ir kt. Sistemų intuityvios vartotojo sąsajos leidžia lengvai keisti darbo eigos modulius, įtraukiant ar šalinant mazgus, bei interaktyviai stebėti darbo eigos būseną ir analizės rezultatus. Vienos iš analizuojamų sistemų darbo eigos modulio pavyzdys pateikiamas 1 paveiksle.



1 pav. ***KNIME sistemos darbo eigos modulio pavyzdys***

WEKA – atvirojo kodo programa, realizuota Java programavimo kalba (Hall et al., 2009). Ši sistema paprasta naudoti pradedančiajam vartotojui. WEKA sistemoje realizuoti įrankiai: duomenų pradinis apdorojimas, klasterizavimas, klasifikavimas, loginės taisyklės, regresija, vizualizavimas. Sistemos pagrindinė vartotojo sąsaja yra *Explorer*, be jos, dar įgyvendinta darbo eigos moduliu paremta sąsaja *Knoweldge Flow* ir komandų eilutė. Vartotojo sąsaja *Experimenter* leidžia vartotojui palyginti tarpusavyje kelių eksperimentų rezultatus, kai analizuojamos skirtingos duomenų aibės (Bouckaert et al., 2012).

KNIME – vartotojui draugiška atvirojo kodo duomenų apdorojimo, analizės ir vizualizavimo sistema, kurios veikimas taip pat grindžiamas darbo eigos moduliu. Sistemą sudaro per 1000 mazgų, kuriuos jungiant sukuriamos darbo eigos schemos. Be to, sistemoje yra integruoti visi WEKA sistemos moduliai (Berthold et al., 2007). KNIME sistema naudojasi daugiau nei 3000 organizacijų daugiau nei 60 pasaulio šalių.

ORANGE – atvirojo kodo duomenų analizės sistema, skirta ir pradedantiesiems, ir ekspertams (Curk, 2005). Sistemoje duomenų tyryba vykdoma naudojant darbų eigos sudarymo įrankį *Orange Canvas* arba programuojant *Phyton* kalba. ORANGE sistemoje realizuotas duomenų pradinis apdorojimas bei populiarūs klasifikavimo, klasterizavimo, vizualizavimo, loginių taisyklių, mokymo be mokytojo, regresijos metodai.

Toliau trumpai aptariami klasifikavimo ir klasterizavimo algoritmai, naudojami eksperimentiniuose tyrimuose. Pasirinkti populiariausi klasifikavimo ir klasterizavimo algoritmai, kurie yra įgyvendinti visose arba bent dviejose sistemose.

Naudojami šie klasifikavimo metodai:

- Bajeso klasifikatorius (angl. *Bayes classification*),
- *k* artimiausių kaimynų (angl. *k- nearest neighbours*),
- sprendimų medis (angl. *decision tree*),
- daugiasluoksnis neuroninis tinklas (angl. *multilayer perceptron*),
- atraminių vektorių klasifikatorius (angl. *support vector machine*).

Naudojami šie klasterizavimo metodai:

- *k* vidurkių (angl. *k- means*),
- hierarchinis klasterizavimas (angl. *hierarchical clustering*).

Naïve Bajeso klasifikatorius remiasi Bajeso taisykle. Laikoma, kad visi duomenų požymiai yra nepriklausomi ir kiekvienas iš požymių daro įtaką klasifikavimo rezultatui. Klasifikatorius skaičiuoja aposteriorines tikimybes kiekvienai klasei. Objektas priskiriamas tai klasei, kuri įgyja didžiausią aposteriorinę tikimybę (Dunham, 2003).

Sprendimų medžio algoritmo rezultatą galima pavaizduoti struktūra, panašia į medį, kurio kiekvienas išsišakojimas reiškia vienos ar kitos sąlygos tenkinimą. Taip sudaromos taisyklės, kurios leidžia nagrinėjamą duomenų aibę suklasifikuoti atsižvelgiant į požymių savybes (Dunham, 2003).

k artimiausių kaimynų metodo idėja yra naujo objekto palyginimas su mokymo aibės objektais, kurie yra panašūs į jį (Han, Kamber, 2006). Norint naują objektą priskirti kuriai nors klasei, skaičiuojami atstumai nuo to objekto iki visų mokymo aibės objektų. Dažniausiai naudojamas Euklido atstumas. Naujas objektas priskiriamas tai klasei, kuriai priklauso dauguma iš artimiausių *k* jo kaimynų.

Dirbtinio neuroninio tinklo struktūra primena biologinius neuroninius tinklus. Daugiasluoksnis neuroninis tinklas sudarytas iš kelių sluoksnių: įvesties, išvesties ir vieno ar daugiau paslėptų neuronų. Be kitų uždavinių, neuroniniai tinklai naudojami ir klasifikavimo uždaviniui spręsti. Tuomet įvesties sluoksnyje pateikiama požymius aprašanti informacija, o išvesties sluoksnyje gaunamas rezultatas – priklausymas klasėms.

Atraminių vektorių klasifikatorius – algoritmas, kuris transformuoja pradinius duomenis į didesnę dimensiją, kur randama hiperplokštuma, skirianti dvi klases kiek galima didesniu atstumu tarp klasifikuojamų duomenų (Han, Kamber, 2006). Radus šią hipreplokštumą, duomenis galima suskirstyti į dvi atskiras klases.

Hierarchinių klasterizavimo metodų rezultatai nusako klasterių tarpusavio hierarchiją, t. y. visi objektai laikomi vienu dideliu klasteriu, kurį sudaro mažesni klasteriai, šiuos – dar mažesni ir t. t. Taikant šiuos metodus, nustatoma bendra visų klasterių tarpusavio priklausomybių struktūra ir tik paskui sprendžiama, koks klasterių skaičius optimalus. Hierarchinis jungimo metodas smulkius klasterius jungia vis į stambesnius, kol galų gale lieka vienas (Čekanavičius, Murauskas, 2002).

k vidurkių metodas yra vienas iš nehierarchinių klasterinės analizės metodų. Nehierarchiniai metodai paprastai taikomi tada, kai iš anksto žinomas (pasirenkamas) klasterių skaičius ir norima klasterizuoti tiriamus objektus. Klasterizavimo procedūrą sudaro tokie žingsniai: 1) objektai suskirstomi į *k* pradinių klasterių; 2) paeiliui apskaičiuojamas kiekvieno objekto atstumas iki klasterių centrų (atstumas paprastai skaičiuojamas naudojantis Euklido metrika arba jos kvadratu); objektas priskiriamas artimiausiam klasteriui; perskaičiuojami klasterių centrai; 3) algoritmas kartojamas tol, kol daugiau nėra perskirstymų (Čekanavičius, Murauskas, 2002).

Visi nurodyti duomenų tyrybos metodai turi tam tikrus valdymo parametrus. Pirmoje lentelėje pateikiamos tyrime naudojamos parametrų reikšmės. Tos pačios parametrų reikšmės naudojamos visose tiriamose duomenų tyrybos sistemose. Pakeitus parametrų reikšmes, klasifikavimo ir klasterizavimo rezultatų absoliutūs dydžiai pasikeistų, tačiau rezultatų, gautų skirtingomis sistemomis, santykiai išliktų tie patys.

Klasifikavimo algoritmų rezultatams įvertinti naudojamas *q* blokų kryžminio patikrinimo metodas (angl. *q-fold cross validation*). Duomenų aibė yra suskaidoma į *q* nesusikertančių blokų. Algoritmas yra apmokomas naudojant *q*–1 bloko duomenis, o likusi duomenų dalis yra naudojama algoritmui testuoti, fiksuojamos klasifikavimo matų reikšmės. Ši procedūra atliekama *q* kartų, mokymui imant vis kitus *q*–1 blokus, pabaigoje randamos klasifikavimo matų vidutinės reikšmės (Han, Kamber, 2006; Witten, Frank, 2005). Tyrime pasirinktas blokų skaičius *q* yra 10. Klasifikavimo tikslumui nustatyti vertinami šie matai: jautrumas (angl. *sensitivity*), bendras klasifikavimo tikslumas (angl. *accuracy*), bendra klasifikavimo klaida (angl. *error*).

Apibrėžkime pagrindines sąvokas:

- tikrai teigiamas (TT) (angl. *true positive – TP*) – objektas X_i priskirtas klasei C_j ir iš tiesų jis jai priklauso;
- tikrai neigiamas (TN) (angl. *true negative –TN*) – objektas X_i nepriskirtas klasei C_j ir iš tiesų jis jai nepriklauso;
- klaidingai neigiamas (KN) (angl. *false negative – FN*) – objektas X_i nepriskirtas klasei C_j, bet iš tiesų jis jai priklauso;
- klaidingai teigiamas (KT) (angl. *false true – FT*) – objektas X_i priskirtas klasei C_j, bet iš tiesų jis jai nepriklauso.

Tada klasifikavimo kokybė yra apskaičiuojama pagal šias formules:

$$\text{jautrumas} = \frac{\text{TT skaičius}}{\text{TT skaičius} + \text{KN skaičius}},$$

1 lentelė. ***Metodų parametrai***

Metodas	Parametrai
Bajeso klasifikatorius	Naïve Bajeso klasifikatorius
k artimiausių kaimynų	*k* = 3
Sprendimų medis	Medžio tipas – C4.5, be genėjimo (sumažinimo), minimalus stebėjimų skaičius lape – 2
Neuroninis tinklas	Vienas paslėptas sluoksnis su 10 neuronų, mokymosi epochų – 50
Atraminių vektorių klasifikatorius	Naudotas tiesinis branduolys
k vidurkių metodas	Klasterių skaičius – 2; mokymosi epochų – 50
Hierarchinis klasterizavimas	Klasterių skaičius – 2; atstumas tarp objektų – Euklido; atstumas tarp klasterių – vienetinė jungtis

$$\text{bendras klasifikavimo tikslumas} = \\ = \frac{\text{TT skaičius} + \text{TN skaičius}}{\text{visų objektų skaičius}},$$

$$\text{klasifikavimo klaida} = \\ = 1 - \text{bendras klasifikavimo tikslumas}.$$

Klasterizavimo kokybei įvertinti parinktas klasterizavimo rezultatų su stebimomis klasėmis (angl. *classes to clusters evaluation*) patikrinimo metodas. Rezultatuose pateikiama neteisingai klasterizuotų objektų dalis procentais.

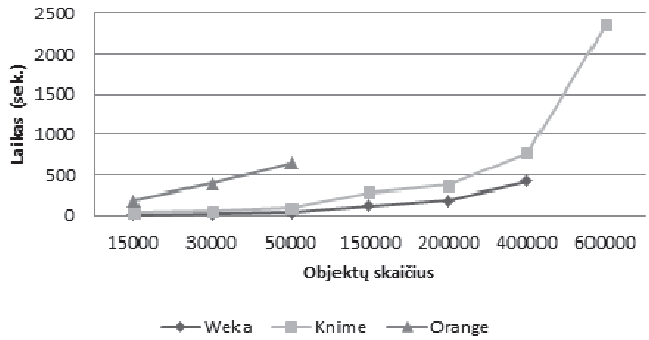
2. Eksperimentinių tyrimų rezultatai

Eksperimentams atlikti naudotas kompiuteris, kurio pagrindinės charakteristikos yra šios: operacinė sistema – Windows 8, operatyvioji atmintis (RAM) – 4 GB, procesorius – Intel i5-3317U, kurio taktinis dažnis – 1,7 GHz (Max Turbo dažnis 2,6 GHz). Atlikus eksperimentus naudojant kitų charakteristikų kompiuterį rezultatų skaitinių išraiškų absoliutūs dydžiai pasikeistų, tačiau išliktų toks pat santykis tarp skirtingomis sistemomis gautų rezultatų.

Eksperimentiniame tyrime siekiama išnagrinėti duomenų tyrybos sistemų galimybes analizuoti įvairaus dydžio duomenis ir nustatyti, kokių apimčių duomenų analizė negalima naudojant šias sistemas. Todėl buvo naudotos ne etaloninės duomenų aibės, skirtos duomenų tyrybos algoritmams vertinti, bet dirbtinai generuotos įvairių apimčių duomenų aibės, kurių požymių reikšmės tolygiai pasiskirsčiusios intervaluose (0; 1) ir (0,8; 2,2). Požymių skaičius fiksuotas – 100, objektų skaičius įvairus – 5000, 15 000, 30 000, 50 000, 150 000, 200 000, 400 000, 600 000. Objektai iš pirmo intervalo priskiriami pirmajai klasei, iš antro – antrajai. Duomenų intervalai parinkti tokie, kad skirtingų klasių duomenys iš dalies susiklotų, kaip dažniausiai yra realiose situacijose. Pasirinktas vienodas visų duomenų aibių požymių skaičius (lygus 100), nes toliau aprašytais eksperimentais norėta parodyti, kaip duomenų tyrybos sistemų pajėgumai priklauso nuo analizuojamų objektų skaičiaus, o ne nuo juos charakterizuojančių požymių skaičiaus. Pasirinkus kitą požymių skaičių, rezultatų absoliutūs dydžiai pasikeistų, tačiau skirtingomis sistemomis gautų rezultatų santykiai išliktų tie patys.

2.1. Klasifikavimo rezultatai

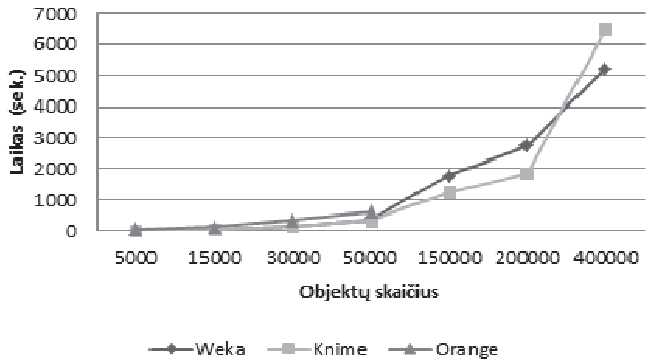
Dėl savo paprastumo Naïve Bajeso klasifikatorius visose lyginamose sistemose gana greitai gauna klasifikavimo rezultatą. Suklasifikuoti 50 000 objektų WEKA sistema užtrunka 31 sek., KNIME – 93 sek., tačiau ORANGE prireikia beveik 11 min. (2 pav.). Toliau didinant objektų skaičių iki 150 000, ORANGE sistema išsijungia dėl kompiuterio operatyviosios atminties trūkumo. Padidinus objektų skaičių iki 400 000, WEKA sistema suklasifikuoti duomenis užtrunka šiek tiek daugiau nei 7 min., o KNIME – kiek mažiau nei 13 min. WEKA sistema naudojant 600 000 objektų sudaro Naïve Bajeso modelį, tačiau pradėjus kryžminį patikrinimą sistema praneša apie klaidą, kad nepakanka kompiuterio operatyviosios atminties, o KNIME susidoroja su duomenimis per 39 min. Kaip matyti iš pateiktų rezultatų (2 pav.), WEKA ir KNIME sistemos gerai susidoroja su duomenimis, sudarytais iš maždaug 400 000 objektų, kai sprendžiamas klasifikavimo uždavinys naudojant Naïve Bajeso klasifikatorių. Šį faktą paaiškina tai, kad klasifikatorius nėra iteracinis, todėl rezultatas gaunamas gana greitai – skaičiavimai užtrunka ne daugiau kaip 13 minučių.



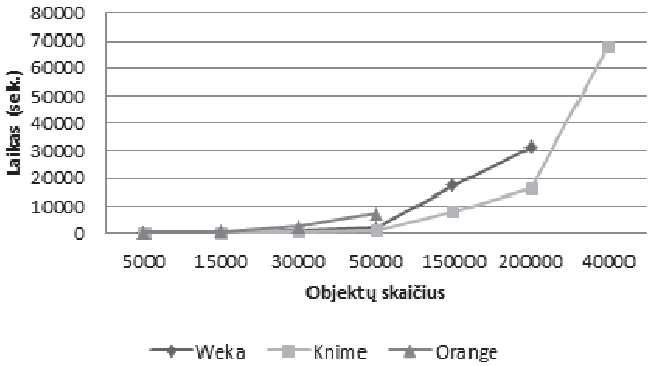
2 p a v. ***Naïve Bajeso klasifikatoriaus vykdymo laiko priklausomybė nuo klasifikuojamų objektų skaičiaus***

Analizuojant duomenų aibes iki 50 000 objektų visos nagrinėjamos sistemos sprendimų medžiui sudaryti ir duomenims suklasifikuoti užtrunka iki 10 min. (3 pav.). Didinant duomenų apimtį, klasifikavimo laikas didėja. Naudojant 400 000 objektų aibę WEKA sistema klasifikavimo rezultatus gauna po 1 val. 27 min., KNIME – po 1 val. 48 min., o ORANGE sistema jau nesusidoroja su 150 000 objektų aibe. 600 000 objektų aibė yra per didelė, ir WEKA bei KNIME sistemose pranešama apie kompiuterio operatyviosios atminties trūkumą.

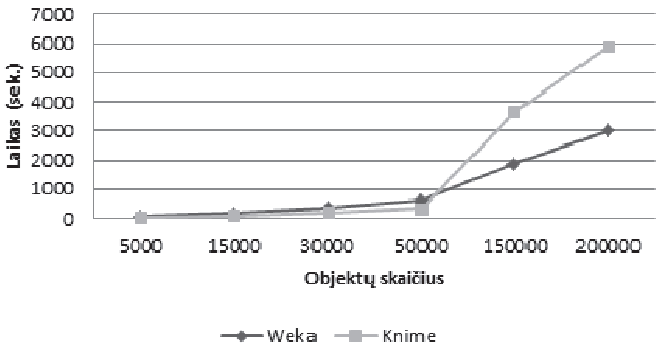
KNIME sistemai prireikė mažiausiai laiko *k* artimiausių kaimynų metodui (*k* = 3) įvykdyti (4 pav.). Nustatyta, kad KNIME užtrunka beveik 19 min., kol suklasifikuoja 50 000 objektų, WEKA – šiek tiek daugiau nei pusvalandį, o ORANGE – beveik dvi valandas. Objektų skaičių padidinus tris kartus, t. y. iki 150 000 objektų, KNIME vykdymo laikas pailgėja 7,2 karto, WEKA – beveik 9 kartus, o ORANGE sistema išsijungia dėl kompiuterio operatyviosios atminties trūkumo. KNIME ir WEKA sistemos suklasifikuoja ir 200 000 objektų aibę, tačiau tai trunka atitinkamai 4 val. 30 min. ir 9 val., ir toks laikas jau dažnai nėra priimtinas tyrėjui. Naudojant 400 000 objektų aibę KNIME sistema objektus suklasifikuoja per 18 val. 55 min., o WEKA sistema po 20 val. darbo įvykdo tik 50 proc. skaičiavimų, taigi toliau vykdyti eksperimentinius skaičiavimus naudojant 600 000 objektų aibę nebuvo prasminga.



3 p a v. ***Sprendimų medžio vykdymo laiko priklausomybė nuo klasifikuojamų objektų skaičiaus***

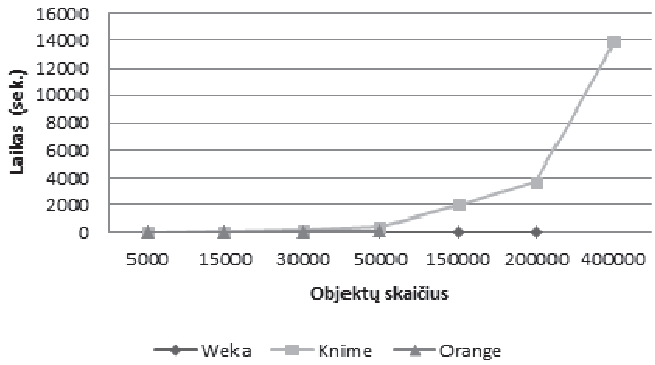


4 p a v. ***k artimiausių kaimynų metodo vykdymo laiko priklausomybė nuo klasifikuojamų objektų skaičiaus***



5 p a v. ***Neuroninio tinklo vykdymo laiko priklausomybė nuo klasifikuojamų objektų skaičiaus***

Neuroninio tinklo metodas yra įgyvendintas tik KNIME ir WEKA sistemose. Lyginant šio metodo vykdymo laiką nustatyta, kad naudojant duomenų aibes iki 50 000 objektų, sistemų vykdymo laikai skiriasi nedaug, tačiau peržengus 50 000 objektų aibės ribą, KNIME vykdymo laikas gerokai padidėja ir 200 000 objektų suklasifikuoja per 1 val. 38 min., WEKA – 49 min. (5 pav.). Naudojant 400 000 objektų aibę tiek WEKA, tiek KNIME sistemos praneša apie kompiuterio operatyviosios atminties trūkumą.



6 p a v. ***Atraminių vektorių klasifikatoriaus vykdymo laiko priklausomybė nuo klasifikuojamų objektų skaičiaus***

WEKA sistemoje atraminių vektorių klasifikatoriaus vykdymo laikas, naudojant tyrimo duomenų aibes iki 200 000 objektų, yra labai trumpas. Net 200 000 objektų suklasifikuojama per 27 sek. (6 pav.). Analizuojant 400 000 objektų aibę, WEKA sistemai pritrūksta kompiuterio operatyviosios atminties. KNIME sistema suklasifikuoja 400 000 objektų, nors tai užtrunka 3 val. 51 min. Naudojant 600 000 objektų aibę ir KNIME sistemai nepakanka kompiuterio operatyviosios atminties. Atraminių vektorių klasifikatoriaus vykdymo laikas ORANGE sistemoje yra trumpesnis nei KNIME, analizuojant tyrimo aibes iki 50 000 objektų. 50 000 objektų aibę KNIME suklasifikuoja per 6 min. 31 sek., ORANGE sistema užtrunka kiek mažiau nei 2 min. Naudojant 150 000 objektų aibę ORANGE sistema išsijungia dėl kompiuterio operatyviosios atminties trūkumo.

Vertinant skaičiavimo laiką, tikslinga vertinti ir klasifikavimo kokybę. Klasifikavimo kokybės matai parodė, kad WEKA sistema *k* artimiausių kaimynų, atraminių vektorių ir neuroninio tinklo klasifikatoriais visus duomenis klasifikuoja 100 proc. tikslumu (2 lentelė). Naïve Bajeso klasifikatoriaus teisingai suklasifikuotų stebėjimų dalis kito nuo 96,48 proc. iki 97,60 proc., o sprendimų medžio – nuo 99,40 proc. iki 99,97 proc.

KNIME sistema *k* artimiausių kaimynų ir atraminių vektorių klasifikatoriais visus duomenis suklasifikuoja 100 proc. tikslumu (3 lentelė). Naïve Bajeso klasifikatoriaus teisingai suklasifikuotų objektų dalis kito nuo 92,22 proc. iki 97,50 proc., sprendimų medžio ir neuroninio tinklo – atitinkamai 99,02–99,97 proc. ir 99,66–99,87 proc.

2 l e n t e l ė. ***WEKA sistemos klasifikavimo kokybės matų reikšmės***

Metodas	Klasė	Jautrumas	Bendras klasifikavimo tikslumas, %	Bendra klasifikavimo klaida, %
Naïve Bajeso klasifikatorius	I klasė	1	96,48–97,60	2,4–3,52
	II klasė	0,934–0,968		
k artimiausių kaimynų klasifikatorius	I klasė	1	100	0
	II klasė	1		
Neuroninis tinklas	I klasė	1	100	0
	II klasė	1		
Sprendimų medis	I klasė	0,996–1	99,40–99,97	0,03–0,60
	II klasė	0,991–1		
Atraminių vektorių klasifikatorius	I klasė	1	100	0
	II klasė	1		

3 l e n t e l ė. ***KNIME sistemos klasifikavimo kokybės matų reikšmės***

Metodas	Klasė	Jautrumas	Bendras klasifikavimo tikslumas, %	Bendra klasifikavimo klaida, %
Naïve Bajeso klasifikatorius	I klasė	1	92,22–97,50	2,50–7,78
	II klasė	0,89–0,97		
k artimiausių kaimynų klasifikatorius	I klasė	1	100	0
	II klasė	1		
Neuroninis tinklas	I klasė	0,998–0,999	99,66–99,87	0,13–0,34
	II klasė	0,995–0,998		
Sprendimų medis	I klasė	0,996–1	99,02–99,97	0,23–0,98
	II klasė	0,984–1		
Atraminių vektorių klasifikatorius	I klasė	1	100	0
	II klasė	1		

4 l e n t e l ė. ***ORANGE sistemos klasifikavimo kokybės matų reikšmės***

Metodas	Klasė	Jautrumas	Bendras klasifikavimo tikslumas, %	Bendra klasifikavimo klaida, %
Naïve Bajeso klasifikatorius	I klasė	1	97,34–97,62	2,38–2,66
	II klasė	0,947–0,952		
k artimiausių kaimynų klasifikatorius	I klasė	1	100	0
	II klasė	1		
Sprendimų medis	I klasė	0,997–0,999	99,06–99,89	0,11–0,94
	II klasė	0,984–998		
Atraminių vektorių klasifikatorius	I klasė	1	100	0
	II klasė	1		

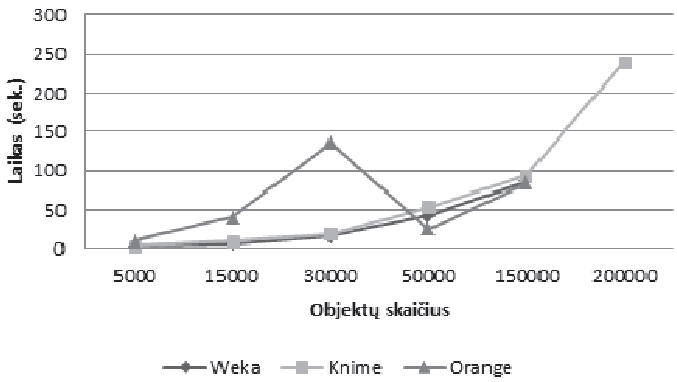
ORANGE sistemoje Naïve Bajeso klasifikatoriaus teisingai suklasifikuotų objektų dalis kito nuo 97,34 proc. iki 97,62 proc., o sprendimų medžio – nuo 99,06 proc. iki 99,89 proc. (4 lentelė). *k* artimiausių kaimynų ir atraminių vektorių klasifikatoriais klaidingai suklasifikuotų objektų visai nėra analizuojant visas duomenų aibes iki 50 000 objektų.

2.2. Klasterizavimo rezultatai

WEKA, KNIME, ORANGE sistemos, naudodamos hierarchinio klasterizavimo metodą, 500 ir 1000 objektų aibes suklasterizuoja per kelias sekundes (5 lentelė), o klasterizavimo rezultatai 100 proc. sutampa su stebimomis objektų klasėmis, t. y. visus pirmos klasės objektus priskiria prie vieno klasterio, antros – prie kito. Daug ilgiau užtrunka 5000 objektų klasterizavimas, be to, klasterizavimo kokybė yra ypač prasta, nes vienas objektas priskiriamas pirmajam klasteriui, o likę 4999 objektai – antrajam. Naudojant 15 000 objektų aibę hierarchinio klasterizavimo metodo veikimas WEKA ir ORANGE sistemose sustoja dėl kompiuterio operatyviosios atminties trūkumo, o KNIME sistema po 9 val. darbo įvykdo tik 14 proc. skaičiavimų, todėl laikoma, kad tai nėra priimtinas laikas, ir skaičiavimai sustabdomi. Hierarchinio klasterizavimo metodo trūkumas – atstumų matricai apskaičiuoti bei jos elementams išsaugoti reikia daug išteklių. Dideliems masyvams (>300) klasterizuoti dažnai naudojami nehierarchiniai klasterizavimo metodai (Čekanavičius, Murauskas, 2002).

5 l e n t e l ė. ***Hierarchinio klasterizavimo metodo vykdymo laiko sekundėmis priklausomybė nuo klasterizuojamų objektų skaičiaus***

Objektų skaičius	WEKA	KNIME	ORANGE
500	2	4	1
1000	6	30	4
5000	1059	4554	265



7 p a v. ***k vidurkių metodo vykdymo laiko priklausomybė nuo klasterizuojamų objektų skaičiaus***

k vidurkių metodu nagrinėjamų duomenų aibių objektai suklasterizuojami į du klasterius visomis trimis sistemomis labai greitai (7 pav.). WEKA ir KNIME sistemoms suklasterizuoti 15 0000 objektų prireikia apie 90 sek. WEKA sistema 200 000 objektų aibės nesuklasterizuoja dėl kompiuterio operatyviosios atminties trūkumo. KNIME sistema 200 000 objektų suklasterizuoja per 4 min., o naudojant 400 000 objektų aibę, pritrūksta kompiuterio operatyviosios atminties. ORANGE sistema nepajėgia klasterizuoti 200 000 ir daugiau objektų aibių dėl kompiuterio operatyviosios atminties trūkumo, be to, ir mažesnės apimties duomenis ji klasterizuoja lėčiau nei kitos sistemos, išskyrus atvejus, kai objektų skaičius 50 000 ir 150 000.

6 l e n t e l ė. ***k vidurkių metodo klasterizavimo rezultatų palyginimas su duomenų klasėmis: neteisingai klasterizuotų stebėjimų dalis (proc.)***

Objektų skaičius	Sistema		
	WEKA	KNIME	ORANGE
5 000	1,9	1,9	4,0
15 000	1,7	1,7	2,0
30 000	1,6	1,6	1,9
50 000	2,5	2,4	4,3
150 000	1,7	1,7	2,0
200 000	*	2,3	*

* trūksta kompiuterio operatyviosios atminties

Palyginus *k* vidurkių metodo klasterizavimo rezultatus su duomenų klasėmis pastebėta, kad KNIME ir WEKA sistemų neteisingai klasterizuotų objektų dalis, analizuojant visas duomenų aibes, vienoda arba beveik vienoda – kinta nuo 1,6 proc. iki 2 proc. (6 lentelė). ORANGE sistemos neteisingai klasterizuotų stebėjimų dalis didesnė (1,9–4,3 proc.). Čia neteisingai klasterizuotų objektų dalis buvo apskaičiuojama taip: pradžioje suskaičiuojama, kiek vienam klasteriui yra priskirta objektų iš kitos klasės nei dauguma to klasterio objektų; tuomet apskaičiuojama procentinė dalis nuo visų tos klasės objektų skaičiaus; skaičiavimai atliekami abiem klasteriams ir gauti rezultatai sumuojami.

Išvados

Atliktas tyrimas parodė, kad ORANGE sistemą galima naudoti kaip duomenų tyrybos įrankį analizuojant duomenų aibes iki 50 000 objektų, kai kiekvieną objektą charakterizuoja 100 požymių. Norint atlikti didesnės aibės analizę, vertėtų rinktis WEKA arba KNIME sistemą. Analizuojant duomenų aibes iki 50 000 objektų, nustatytas panašus WEKA ir KNIME sistemų skaičiavimo laikas vykdant visus nagrinėtus algoritmus, o ORANGE sistema užtrunka ilgiau tiems patiems skaičiavimams atlikti. Nors WEKA sistemai reikia mažiau laiko atlikti skaičiavimams taikant didžiąją dalį nagrinėtų algoritmų, tačiau KNIME sistema tam tikrais atvejais pajėgi apdoroti didesnės apimties duomenis nei WEKA. Galima teigti, kad turint tik ORANGE sistemą didelės apimties duomenys yra tie, kurie sudaryti iš daugiau nei 50 000 objektų. Analizuojant duomenis WEKA ar KNIME sistemomis, didesnės nei 200 000 objektų duomenų aibės jau yra didelės apimties, nors naudojant nesudėtingus klasifikavimo metodus pastarosios dvi sistemos pajėgios apdoroti

ir didesnės apimties duomenis – 400 000 objektų, o KNIME – dar ir 600 000 objektų. Jei duomenų apimtys yra didesnės, būtinos didelėms duomenų aibėms pritaikytos duomenų tyrybos sistemos, pajėgios pasitelkti lygiagrečiuosius ir paskirstytuosius skaičiavimus. Be abejo, tyrimams naudojant kitų charakteristikų kompiuterį, rezultatai skirtųsi, tačiau bendros tendencijos išliktų, t. y. KNIME ir WEKA sistemos būtų pranašesnės už ORANGE.

Tyrimo rezultatai parodė, kad taikyti klasifikavimo metodai duoda tikslius klasifikavimo rezultatus, sprendžiant testinį uždavinį, kai klasės tik šiek tiek susikloja. Įprastai praktikoje kylančiuose uždaviniuose klasių sanklota būna didesnė, todėl klasifikavimo rezultatai gali būti ir šiek tiek blogesni. Prieš pasirenkant duomenų tyrybos sistemą derėtų ne tik atsižvelgti į turimų duomenų aibės dydį, bet ir įvertinti pasirinktų algoritmų sudėtingumą, kuris daro įtaką skaičiavimo laikui, nes mažai skaičiavimų reikalaujantis algoritmas gali susidoroti ir su didesnės apimties duomenimis, skaičiavimams imlus algoritmas gali užstrigti analizuojant ir mažesnę duomenų aibę.

Ateityje būtina atlikti eksperimentinius tyrimus naudojant įvairesnius, daugiau nei dviejų klasių duomenis, sudarytus iš įvairių požymių skaičių. Taip pat verta nagrinėti ir kitas populiarias duomenų tyrybos sistemas. Tai leistų daryti tikslesnes išvadas apie sistemų galimybes analizuoti įvairių apimčių duomenis.

LITERATŪRA

BERTHOLD, M.; CEBRON, N.; DILL, F.; GABRIEL, T., KOTTER, T.; MEINL, T., et al. (2008). KNIME: The Konstanz Information Miner. *Data Analysis, Machine Learning and Applications*. In: *Studies in Classification, Data Analysis, and Knowledge Organization (GfKL)*. Freiburg: Springer, p. 319–326.

BOUCKAERT R. R.; FRANK E.; HALL M.; KIRKBY R.; REUTEMANN P.; SEEWALD A.; SCUSE D. (2012). *WEKA Manual for Version 3-6-7* [interaktyvus] [žiūrėta 2013 m. kovo 15 d.]. Prieiga per internetą: <http://www.cs.waikato.ac.nz/ml/weka/documentation.html/>.

CHEN, X.; YE, Y.; WILLIAMS, G.; XU, X. (2007). A Survey of Open Source Data Mining Systems. Emerging Technologies in Knowledge Discovery and Data Mining. *PAKDD 2007, International Workshops, Nanjing, China, May 22–25, 2007, Revised Selected Papers,* Lecture Notes in Computer Science, vol. 4819, p. 3–14.

CURK, T.; DEMŠAR, J.; XU, Q.; LEBAN, G.; PETROVIČ, U.; BRATKO, I., et al. (2005). Microarray data mining with visual programming. *Bioinformatics*, vol. 21(3), p. 396–398.

ČEKANAVIČIUS, V.; MURAUSKAS G. (2002). *Statistika ir jos taikymai*. II dalis. Vilnius: TEV. 268 p. ISBN 9955-491-16-7.

DUNHAM, M. H. (2003). *Data Mining Introductory and Advanced Topics*. New Jersey: Pearson Education, Inc. Prentice Hall. 315 p. ISBN 0-13-088892–3.

HALL, M.; FRANK, E.; HOLMES, G.; PFAHRINGER, B.; REUTEMANN, P.; WITTEN, I. H. (2009). The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, vol. 11(1), p. 10–18.

HAN, J.; KAMBER, M. (2006). *Data Mining Concepts and Techniques*. Second Edition. San Francisco, CA: Morgan Kaufmann Publishers. 743 p. ISBN 13:978-1-55860-901-3.

MADASAMY, B.; TAMILSELVI, J. J. (2012). Assesement of Freeware Data Mining Tools over Some Wide-Range Characteristics. In: *International conference on information processing, Wireless Networks and Computational Intelligence, ICIP 2012,* Communications in Computer and Information Science, vol. 292, p. 529–535.

WAHBEH, A. H.; AL-RADAIDEH Q. A.; AL-KABI, M. N.; AL-SHAWAKFA, E. M. (2011). A Comparison Study between Data Mining Tools over some Classification Methods. *International Journal of Advanced Computer Science and Applications, Special Issue on Artificial Intelligence*, vol. 0(3), p. 18–25.

WITTEN, I. H.; FRANK E. (2005). *Data Mining: Practical Machine Learning. Tools and Techniques*. Second Edition. San Francisco, CA: Morgan Kaufmann Publishers. 525 p. ISBN 0-12-088407-0.

ZUPAN, B.; DEMSAR, J. (2008). Open-Source Tools for Data Mining. *Laboratory and Clinical Medicine*, vol. 28, p. 37–54.

INVESTIGATION OF THE ABILITIES OF DATA MINING SYSTEMS TO ANALYSE VARIOUS VOLUME DATASETS

Kotryna Paulauskienė, Olga Kurasova

S u m m a r y

The aim of the paper is to determine what volume of data the popular data mining systems are able to analyse within a reasonable period of time, when solving classification and clustering problems. Three open source data mining systems are investigated: WEKA, KNIME, and ORANGE. The experiments have been carried out with eight datasets, where the number of attributes was fixed – 100 and the number of instances ranged between 5000 and 600 000. The experimental investigation has shown that when the ORANGE system is used, the data of more than 50 000 instances are of too large volume. In order to analyse larger datasets, the WEKA and KNIME systems need to be used. The data of more than 200 000 instances are of too large volume for WEKA and KNIME, however, when simple classification methods are used, both systems are able to handle 400 000 instances, and KNIME – 600 000 instances. The results have showed that KNIME can handle larger datasets than WEKA, when applying some classification methods. The accuracy of classification is high enough, when the classification methods, implemented in the systems, are used.

Mobiliosios duomenų tyrybos paslaugų analizė

Gediminas Bazilevičius

Vilniaus universiteto Matematikos ir informatikos instituto doktorantas
Vilnius University, Institute of Mathematics and Informatics, PhD student
Akademijos g. 4, Vilnius
El. paštas: gediminas.bazilevicius@mii.vu.lt.

Mobilioji medicina užima vis didesnę sveikatos priežiūros dalį. Naudojamos pažangesnės, spartesnės, mobilesnės technikos, stebinčios pacientų sveikatos būklę bei informuojančios apie galimus sveikatos sutrikimus. Identifikuoti sveikatos pokyčius neužtenka vien pažangios technikos. Įrenginiams kuriama programinė įranga, taikomosios programos, o šioms naudojami metodai, kurie galėtų kuo tiksliau identifikuoti menkiausius sveikatos pokyčius. Šis straipsnis skirtas apžvelgti, kokie žinių gavybos proceso duomenų tyrybos metodai gali būti naudojami mobiliuosiuose ir dėvimuosiuose įrenginiuose, taikant paslaugų stiliaus architektūros modelį ir saityno paslaugas medicinos sistemose.

Įvadas

Pirmiausia apibrėžkime įrenginius, kuriais vyksta duomenų gavyba bei tyryba. Mobilusis įrenginys – tai rankinis mažas kompiuteris, kuris dažniausiai turi operacinę sistemą, ekranėlį, integruotus belaidžio ryšio įrenginius (*Wireless*, *Bluetooth*), procesorių, operatyviąją atmintį, duomenų saugyklą, bateriją bei kitą techninę ir programinę įrangą. Mobiliųjų įrenginių klasei galima priskirti mobilųjį telefoną, delninį kompiuterį, planšetinį kompiuterį, knygų skaityklę. Šie įrenginiai gali atlikti duomenų gavybos, tyrybos, saugyklos funkcijas. Dėvimiesiems įrenginiams galima priskirti prastesnių nei mobiliųjų įrenginių techninių charakteristikų įrenginius su jutikliais, kurie įprastai atlieka tik duomenų gavybos bei jų siuntimo funkciją. Šie elektroniniai įrenginiai dažniausiai turi specifinę programinę įrangą, minimalias įrenginio valdymo galimybes, bateriją, vieną belaidžio ryšio įrenginį (Wireless ar Bluetooth). Šiuo metu jau yra dėvimųjų įrenginių, kuriuose integruojamas mažas ekranas, procesorius, operatyvioji atmintis. Dėvimųjų įrenginių klasei galima priskirti: *Google* akinius, elektroninę tekstilę, sveikatos stebėjimo sistemas (pvz.: laikrodžiai, apyrankės, krūtinės ląstos diržai ir pan.) ir daug kitų įrenginių.

Dėvimųjų jutiklių siunčiami duomenys gali būti surenkami ir iš dalies kaupiami mobiliuosiuose įrenginiuose arba stacionariuose kompiuteriuose esančiose duomenų saugyklose. Klasikinis duomenų gavybos procesas atliekamas su duomenimis, esančiais duomenų saugyklose (Han, Kamber 2006). Tobulėjant mobiliesiems įrenginiams vis daugiau duomenų gavybos proceso etapų perkeliama į šiuos įrenginius. Duomenų surinkimo ir išankstinio apdorojimo atlikimas mobiliajame įrenginyje palengvina ir paspartina duomenų perdavimą į duomenų saugyklas.

Šiuolaikinis medicinos sektorius yra sudėtingas, nes daugybė programinės ir įvairios kompiuterinės įrangos yra naudojama skirtingose platformose: internetinių sistemų komunikavimui, medicinos sistemų integravimui, medicininių duomenų kaupimui, apdorojimui ir t. t. Didžioji dalis šiuolaikinės programinės įrangos priklauso paskirstytosioms sistemoms, kurios yra sudarytos klientas-serveris (angl. *client-*

server) architektūros pagrindu. Šios sistemos paprastai aptarnauja didelį skaičių vartotojų, kurie naudojasi serverio teikiamomis paslaugomis. Dažniausiai paskirstytąsias sistemas sudaro duomenų bazės bei veikiančios paslaugos (angl. *services*).

Šis straipsnis skirtas apžvelgti, kokie žinių gavybos proceso duomenų tyrybos metodai gali būti naudojami mobiliuosiuose ir dėvimuosiuose įrenginiuose, taikant paslaugų stiliaus architektūros modelį ir saityno paslaugas medicinos sistemose.

Į paslaugas nukreipta architektūra

Į paslaugas nukreipta architektūra (angl. *Service Oriented Architecture* – SOA) – tai programinės įrangos architektūros modelis, skirtas sujungti verslo ir skaičiavimo išteklius, kad būtų galima pasiekti galutinius vartotojo pageidaujamus rezultatus (Thomas 2005).

Šios architektūros apibrėžimas tapatinamas su saityno paslaugų naudojimu, nes remiantis šia architektūra kuriamos į paslaugas nukreiptos technologijos.

Į paslaugas nukreiptoje architektūroje gali būti jungiamos programos, nepriklausomai nuo platformos (t.y. operacinės sistemos, naudojamo serverio architektūros) ir programavimo kalbos, kurių funkcionalumas prieinamas per saityno paslaugas. SOA leidžia pagrindinį įmonės verslą padaryti nepriklausomą nuo jos informacinių technologijų. Į paslaugas nukreiptos architektūros pagrindas – standartizuotas centrinis modulis (angl. *Enterprise service bus* – ESB), prie kurio jungiami likę tam tikrą paslaugą atliekantys moduliai, pavyzdžiui, santykių su klientais valdymas (angl. *Customer Relationship Management* – CRM), įmonės išteklių planavimas (angl. *Enterprise Resource Planning* – ERP), finansų apskaitų sistemos, sandėliavimas, sąskaitų išrašymas ar net paprasčiausia dokumentų redagavimo programa. Standartizuotas centrinis modulis yra tarpinė programinė įranga, skirta atskirų programų arba sistemų integracijai ir atliekanti duomenų transformavimo ir persiuntimo funkciją (IBM 1986).

Saityno paslauga – tai modulinės save aprašančios taikomosios programos, suprojektuotos siekiant užtikrinti heterogeninių kompiuterinių sistemų sąveiką tinkle (Kreger 2001). Išskirtinė saityno paslaugų savybė yra ta, kad saityno paslaugos skirtos programa-programa (mašina-mašina) bendravimui. Antra, saityno paslaugos pasiekiamos per internetą. Trečia, joms aprašyti naudojami XML pagrindu sukurti sprendimai (Akinci 2004). Norint paslaugą padaryti lanksčią ir plačiau pritaikomą, duomenų dalijimuisi būtina, kad paslauga atitiktų standartus: XML (angl. *Extensible Markup Language*), SOAP (angl. *Simple Object Access Protocol*), WSDL (angl. *Web Service Description Language*), UDDI (angl. *Universal Description, Discovery and Integration*). Kitos sistemos sąveikauja su saityno paslaugomis, atsižvelgdamos į paslaugas nustatytomis taisyklėmis, SOAP protokolo pranešimais ar REST pagalba.

Duomenų tyryba

Duomenų tyryba (angl. *data mining*) – tai anksčiau nežinomos ir potencialiai naudingos informacijos ištraukimas iš didelių duomenų saugyklų (Margaret 2002). Tai tendencijų, šablonų, ryšių ir kategorijų aptikimo procesas, naudojant šablonus bei mašininius mokymosi algoritmus. Šio proceso metu, naudojant šablonų atpažinimo protokolus ir taikant statistinius bei matematikos metodus, išsamiai tiriami duomenys. Tradiciniai duomenų tyrybos metodai taikomi tuomet, kai turima visa duomenų aibė, t. y. ji yra fiksuota. Jeigu duomenys gaunami srautu ir taip greitai, kad juos nenaudinga išsaugoti duomenų saugykloje, o reikia tirti realiu laiku, – turime srauto duomenų tyrybos atvejį (Anand ir Jeffrey 2012).

Galima išskirti šias srauto duomenų informaciją apibūdinančias charakteristikas:

- duomenys gaunami nenutrūkstamai iš srauto duomenų;
- nėra iškeltų srauto duomenų tvarkymo prielaidų;
- gaunamas srauto duomenų kiekis yra nenumatomas ir nesuskaičiuojamas.

Kai turime srautą duomenų, tirti juos tradiciniais duomenų tyrybos metodais negalima, nes neįmanomas daugkartinis duomenų skenavimas informacijai išgauti. Efektyviai ir veiksmingai fiksuoti žinias iš srauto duomenų yra sudėtinga dėl tinkle atliekamos stebėsenos bei saityno srauto. Šiuo atveju reikalingi dalinai automatiniai interaktyvūs metodai, kuriais išgaunamos paslėptos žinios ir informacija realiu laiku.

Tiriamo srauto duomenų problema gali būti išspręsta remiantis šiomis metodologijomis (Neha, Indejeet 2013):

- viso duomenų rinkinio poaibių nagrinėjimas arba duomenų transformavimas siekiant sumažinti duomenų dydį;
- efektyvus algoritmų panaudojimas laiko ir vietos atžvilgiu.

Pirmoji metodologija susijusi su įvykiu, kurio metu visi duomenys turi būtų sumažinami iš viso kiekio duomenų arba rankiniu būdu parenkama tam tikra rinkinio dalis (poaibis). Tam naudojami šie metodai: atranka (angl. *sampling*), apkrova (angl. *shedding*), piešimas (angl. *sketching*), konspektavimas (angl. *synopsis*), duomenų struktūrinimas ir kaupimas (angl. *aggregation*) (Neha ir Indejeet 2013).

Antroji metodologija stengiasi efektyviau išnaudoti esamus duomenų tyrybos algoritmus, tam juos modifikuodama ir pritaikydama srauto duomenų charakteristikas bei ypatybes. Siekiama, kad srauto duomenų tyrybos algoritmai veiktų realiu laiku. Į šią kategoriją patenka apytikslio skaičiavimo algoritmas (angl. *approximation algorithm*), slenkančio lango algoritmas (angl. *sliding windows algorithm*) ir detalios išeigos algoritmas (angl. *output granularity algorithm*) (Neha, Indejeet 2013).

Pagrindinai skirtumai tarp srauto ir tradicinės duomenų tyrybos pateikiami lentelėje.

Pastaraisiais metais vis labiau populiarėjant mobiliesiems įrenginiams bei didėjant jų techninėms galimybėms, kuriamos mobiliosios duomenų tyrybos sistemos. Šių sistemų pagrindas yra belaidė į paslaugas orientuota architektūra. Tokiu būdu mobiliųjų įrenginių klientinės taikomosios programos gali taikyti nuotolinį duomenų tyrybos užduočių vykdymą ir atvaizduoti duomenų rezultatus mobiliajame įrenginyje. Mobiliosios duomenų tyrybos paslaugos leidžia vartotojams naudotis saugomais duomenimis, kurie yra geografiškai nutolę nuo vartotojo buvimo vietos, bei leidžia išgauti žinias iš duomenų saugyklų. Apibendrinant pasakytina, kad mobilioji duomenų tyryba – tai mobiliųjų įrenginių procesas, vykstantis duomenų tyrybos taikomosiose programose, kurios naudojasi nutolusiais ištekliais (kompiuteriais) ir nutolusiais duomenų centrais (Domenico, Paolo 2010).

Mobilioji duomenų tyryba – tai technologija, suteikianti galimybę realiu laiku analizuoti srauto duomenis, generuotus pačiame telefone arba iš dėvimųjų įrenginių, kurie yra netoli šio įrenginio. Atskirais atvejais mobilusis įrenginys gali atstoti tarpininką tarp serverio ir jutiklio.

Lentelė. ***Duomenų apdorojimo skirtumai srauto ir tradicinės duomenų tyrybose (Mohamed et al., 2010)***

Srauto duomenų tyryba	Duomenų tyryba
Apdorojimas realiu laiku	Tradicinis apdorojimas
Greita duomenų tyryba, priklausanti nuo skaičiavimo išteklių	Normali arba lėta duomenų tyryba, santykinai priklausanti nuo turimų skaičiavimo šaltinių
Duomenų saugojimas nėra įgyvendintas	Duomenų saugojimas yra įgyvendintas
Priimami apytiksliai rezultatai	Reikalingi tikslūs rezultatai
Duomenų mėginių apdorojimas yra įprastas darbas	Kiekvieno duomenų elemento/įrašo apdorojimas yra įprastas darbas
Agreguota saugykla, kurioje saugomi tik apibendrinti duomenys	Neapdorotų duomenų saugykla
Erdvinis ir laikinis kontekstas yra ypač svarbus	Erdvinis ir laikinis kontekstas yra laikomas tam tikroms taikomųjų programų klasėms, pvz.: *Geographical Information Systems* (GIS)
Naudojami linijiniai ir daugialinijiniai skaičiavimų metodai	Naudojami metodai, reikalaujantys didelio kiekio atminties ir daug laiko

Mobiliosios srauto duomenų tyrybos metodai paprastai sutelkti į prisitaikančius srauto duomenų tyrybos algoritmus, kurie veiktų veiksmingai mobiliuosiuose prietaisuose. Taigi tikslas yra toks: srauto duomenų tyryba turi būti atliekama taip, kad neviršytų ribotų skaičiavimo išteklių, ekrano dydžio, energijos suvartojimo bei nurodytų žemiausių ir aukščiausių tikslumo ribų. Tam naudojamos įvairios duomenų tyrybos strategijos, leidžiančios dinamiškai keisti algoritmų parametrus atsižvelgiant į esamų išteklių būklę, situaciją ir kitus apribojimus. Tam algoritmai yra kuriami naudojant lengvos formos (angl. *light-weight*) technikas:

- klasterizavimo algoritmams:
  - *Light-Weight Clustering*;
  - *RA-Cluster* ir *DRA-Cluster*;
- pasikeitimui aptikti:
  - CHANGE-DETECT;
- klasifikavimo algoritmams:
  - *Light-Weight Class*;
- dažnai pasitaikantiems elementams ir asociacijoms nustatyti:
  - *Light-Weight Frequent Items*;
  - *High Correlated Energy-Efficient*.

Iš daugybės duomenų tyrybos tematikos tyrimų galima teigti, kad įmanoma analizuoti srauto duomenis realiu laiku mobiliajame įrenginyje. Rezultatų atvaizdavimas realiu laiku tebėra iššūkis. Tačiau galima paminėti, kad dirbant šia kryptimi susiduriama su šiomis problemomis:

1. Dėl įvairių mobiliųjų įrenginių gausos ir jų mažų ekranų sudėtinga efektyviai išnaudoti skirtingų matmenų ekranus, kad duomenis būtų galima pateikti naudingai ir informatyviai;
2. Vizualizuojant duomenis reikia nuolat vykdyti skaičiavimus, kad vartotojas neprarastų atvaizduojamų duomenų tikroviškumo;
3. Greitai ir dažnai keičiant vaizdus reikėtų fiksuoti ir saugoti duomenis taip, kad nebūtų prarastas tikslumas (Shonali et al., 2012).

Šiame straipsnyje panagrinėsime keletą į paslaugas nukreiptoje architektūroje sukurtų mobiliosios duomenų tyrybos paslaugų.

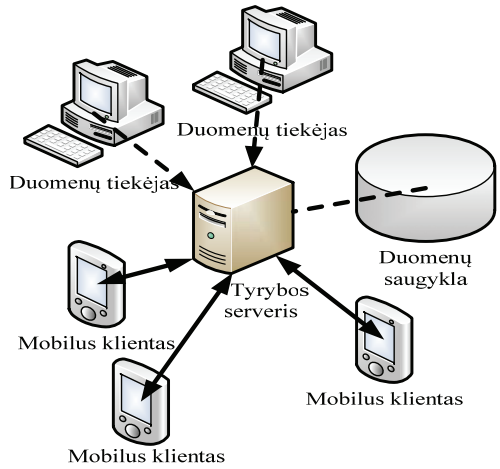
Mobilioji duomenų tyryba, naudojanti saityno paslaugas

Mobiliųjų įrenginių, kaip saityno paslaugų klientų, valdančių duomenų tyrybos užduotis, naudojimas yra vienas iš klasikinių mobiliųjų įrenginių naudojimų būdų. Tokia sistema leidžia nutolusiems vartotojams vykdyti duomenų tyrybos užduotis iš mobiliojo telefono ar asmeninio skaitmeninio asistento (angl. PDA) ir gauti analizuojamų duomenų rezultatus. Tokios sistemos pagrindiniai komponentai (1 pav.):

- duomenų tiekėjas: taikomosios programos, gaminančios duomenis apdoroti;
- mobilusis klientas: taikomoji programa, kuri reikalauja vykdyti duomenų tyrybą nutolusiuose duomenų centruose;
- tyrybos serveris: serverio mazgai, kurie naudojami generuotiems ir iš tiekėjų gautiems duomenims saugoti bei duomenų tyrybos užduotims vykdyti.

Generuoti tiekėjų duomenys yra surenkami tyrybos serverių ir laikomi lokalioje duomenų saugykloje. Priklausomai nuo taikomosios programos reikalavimų, iš tiekėjų gauti duomenys įkeliami daugiau nei į vieną tyrybos serverį.

Pagrindinė tyrybos serverių taisyklė yra leidimas klientams vykdyti duomenų tyrybą nu-



1 p a v. ***Mobiliosios tyrybos sistemos architektūra (Domenico, Paolo, 2010)***

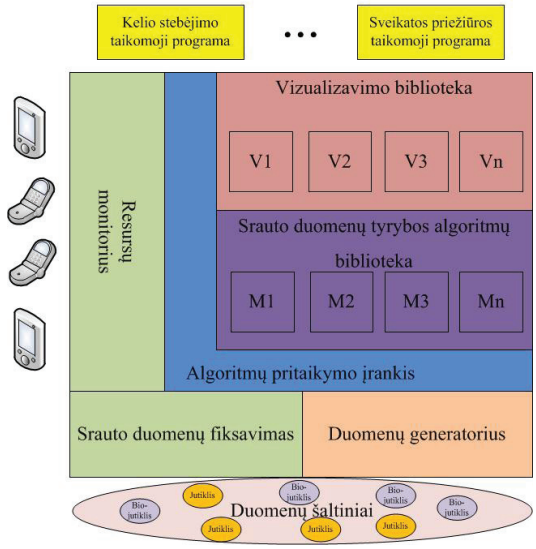
tolusiame duomenų centre naudojant duomenų tyrybos algoritmų rinkinius. Kai mobilusis klientas yra prijungtas prie serverio, jam leidžiama pasirinkti duomenų dalį bei algoritmą. Pasibaigus duomenų tyrybos užduočiai tyrybos serveryje, skaičiavimo rezultatai yra siunčiami ir atvaizduojami kliento įrenginyje tekstine arba vaizdine forma.

Duomenų tyrybos paslauga mobilieji klientai naudojasi tam, kad atliktų duomenų tyrybos uždavinius. Duomenų tyrybos paslaugos sąsaja apibrėžia veiksmų rinkinį, kuris leidžia gauti prieinamų duomenų rinkinių ir algoritmų sąrašą, pateikti duomenų tyrybos užduotis, gauti esamą skaičiavimo būseną ir rezultatus. Duomenų tyrybos paslauga atlieka duomenų analizę naudodama algoritmų rinkinius, kuriuos turi taikomosios programos Weka biblioteka (Hall et al., 2009). Kai duomenų tyrybos uždaviniai yra pateikti duomenų tyrybos paslaugai, tuomet naudotojo pasirinktas Weka bibliotekos algoritmas yra naudojamas lokaliam duomenų rinkiniui analizuoti.

Plačiau paties mobiliojo įrenginio galimybes stengiamasi išnaudoti pasitelkiant „Open Mobile Miner“ (Krishnaswamy et al., 2009) – programinį paketą, veikiantį mobiliajame įrenginyje ir leidžiantį apdoroti srautu gaunamus duomenis. Šis įrankių rinkinys gali būti lengvai pritaikomas sveikatos duomenims apdoroti, akcijų ar nekilnojamojo turto duomenims analizuoti. Taip pat jis leidžia kurti duomenų tyrybos programas, pritaikytas daugeliui mobiliųjų įrenginių, integruoti į įrankį naujus ar esamus srauto duomenų tyrybos algoritmus, suteikia sąsają gauti duomenis iš įvairių srauto šaltinių (pvz., Bluetooth daviklių).

Pagrindinės *Open Mobile Miner* architektūros dalys (2 pav.): duomenų šaltiniai, srauto duomenų fiksavimas, srauto duomenų tyrybos algoritmų biblioteka, algoritmų pritaikymo įrankis, vizualizavimo biblioteka, išteklių monitorius.

Duomenų šaltiniai gali būti srauto duomenys, siunčiami iš aplinkos jutiklių (temperatūros, spaudimo ir kt.), biojutiklių (širdies ritmo, kardiogramos, judėjimo lygio ir t. t.). Šie jutikliai gali komunikuoti / siųsti duomenis per įvairius komunikavimo kanalus (pvz., 802.11, *Bluetooth*). Tai pat duomenų šaltiniais gali būti lokalūs CSV failai. Srauto duomenų fiksavimo komponentė perduoda iš duomenų šaltinių gautus duomenis į duomenų tyrybos algoritmus per algoritmų pritaikymo įrankį.



2 p a v. ***Konceptuali „Open Mobile Miner“ architektūra (Krishnaswamy eal., 2009)***

Srauto duomenų tyrybos algoritmų bibliotekoje yra algoritmai, skirti mobiliajai duomenų tyrybai. Šie algoritmai atlieka įvairių tipų analizę: klasterizavimą (*LightWeight Cluster* (LWC), *RA-Cluster*, RA-VFKM), klasifikavimą (*LightWeight Classification* (LWClass), dažnų šablonų aptikimą (*LightWeight Frequent items*), pasikeitimų aptikimą (*Change-Detection*) ir laiko eilutės analizę (RA-SAX). Algoritmų pritaikymo įrankis naudoja įvairias strategijas šių algoritmų dinamiškam valdymui atžvelgiant į srauto duomenų charakteristikas: duomenų šaltinį, įrenginio išteklių lygį (t. y. skaičiavimo išteklių būseną, įskaitant baterijos lygį), norimo rezultato tikslumo lygį. Išteklių monitorius atsakingas už atminties užpildymo, procesoriaus apkrovos ir baterijos talpos įvertinimą. Ši informacija yra perduodama algoritmų pritaikymo

įrenginiui. Vaizdavimo biblioteka leidžia analizės proceso rezultatus parodyti įvairiais metodais. Šis komponentas yra pakankamai lankstus, kad galėtų prisiderinti prie skirtingų duomenų tyrybos algoritmų bei mobiliųjų įrenginių techninių charakteristikų.

Išvados

Literatūros analizė parodė tendenciją, kad sveikatos priežiūros srityje vis dažniau pritaikomos pažangesnės, spartesnės, mobilesnės technikos. Jos stebi, kaupia duomenis apie pacientų sveikatos būklę ir informuoja apie galimus sveikatos sutrikimus. Menkiausiems sveikatos pokyčiams identifikuoti kuriamos bei tobulinamos mobiliųjų įrenginių programinės įrangos, taikomosios programos naudojant tam tikrus metodus ir technologijas. Dėl nesudėtingo integralumo su kitomis sistemomis imta plačiau naudoti saityno paslaugas, kurios pagrįstos į paslaugas orientuota architektūra.

Šiame darbe apžvelgtos mobiliosios duomenų tyrybos, srautinės duomenų tyrybos problemos, įrankiai. Analizė parodė, kad nors technologija vystoma gana seniai, tačiau sukurtų ar modifikuotų algoritmų, skirtų mobiliajai duomenų tarybai, nėra daug. Duomenų tyrybą mobiliuosiuose įrenginiuose riboja skaičiavimo ištekliai (atminties kiekis, procesoriaus greitis, tinklo pralaidumas, baterijos talpa), mobiliųjų įrenginių gausa ir techninių charakteristikų skirtumai, todėl dažnai duomenų tyrybos rezultatų neįmanoma turėti realiu laiku. Analizuojant literatūroje minimus algoritmus nustatyta, kad modifikuojant algoritmus mobiliesiems įrenginiams prarandamas rezultatų tikslumas. Siekiant gauti tikslesnius skaičiavimų rezultatus, padidėja laiko sąnaudos arba kyla problemų dėl nepakankamų mobiliųjų įrenginių techninių išteklių.

Padėka. Šis tyrimas atliktas Europos socialinio fondo finansuojamo projekto „Paslaugų interneto technologijų kūrimo ir panaudojimo našių skaičiavimų platformose teoriniai ir inžineriniai aspektai" (Nr. VP1-3.1-ŠMM-08-K-01-010) lėšomis.

LITERATŪRA

AKINCI, H. 2004. Geospatial web services for e-municipality. *International archives of photogrammetry remote sensing and spatial information sciences*, no. 35(2). ISSN: 1682-1750.

Anand, R.; JEFFREY, D. U. 2012. *Mining of massive datasets.* United Kingdom: Cambridge University Press. ISBN-13: 9781107015357.

DOMENICO, T.; PAOLO, T. 2010. *Service-oriented distributed knowledge discovery.* Minnesota: Chapman & Hall/CRC. ISBN: 9781439875315.

HALL, M.; FRANK, E.; HOLMES, G.; PFAHRINGER, B.; REUTEMANN, P.; WITTEN, I. H. 2009. The WEKA data mining software: an update. *SIGKDD Explorations*, vol. 11(1). ISSN: 1931-0145

HAN, J.; KAMBER, M. 2006. *Data mining: concepts and techniques*. 2nd. San Francisco: Morgan Kaufmann Publishers. ISBN-13: 9781558609013.

IBM. 1986 m. kovo 19 d. *Į paslaugas orientuota architektūra (SOA)* (IBM) [žiūrėta 2013 m. balandžio 14 d.]. In: IBM – Lietuva: Prieiga per internetą: <http://www-05.ibm.com/lt/ondemand/insights/soa.html>.

KREGER, H. 2001. *Web services conceptual architecture* (WSCA 1.0). New York: IBM Software Group.

KRISHNASWAMY, S.; GABER, M. M.; HARBACH, M.; HUGUES, C.; SINHA, A.; GILLICK, B.; et al. 2009. Open mobile miner: a toolkit for mobile data stream mining. In: *Proccedings of the ACM KDD'09*. Paris: ACM.

MARGARET, H. D. (2002). *Data mining: introductory and advanced topics.* USA, USA: Prentice Hall. ISBN: 0130888923.

MOHAMED, M. G.; ARKADY, Z.; SHONALI, K. (2010). Data stream mining. In: M. Oded, R. Lior. *Data Mining and Knowledge Discovery Handbook*. New York: Springer, p. 761.

Neha, G.; Indejeet, R. (2013). Stream data mining: a survey. *International Journal of Engineering*

Research and Applications, vol. III(1), p. 1113–1118. ISSN: 2248–9622.

SHONALI, K.; JOAO, G.; MOHAMED, M. G. (2012). *Mobile data stream mining: from algorithms to applications*. Washington, DC, USA: IEEE Computer Society Washington, p. 360–363.

THOMAS, E. (2005). *Service-oriented architecture: concepts, technology, and design*. New York, USA: Prentice Hall. ISBN: 0131858580.

ANALYSIS OF SERVICES FOR MOBILE DATA MINING

Gediminas Bazilevičius

S u m m a r y

Medicine takes an increasing share of health care in web services. Smarter, faster mobile equipment is applied to monitor a patient's medical condition and to inform about his possible health problems. A special software for mobile devices is developed to recognize the smallest changes in a patient's health state. In the paper mobile data mining and mobile stream data mining methods and solutions, based on the service-oriented architecture, as well as Web services are analysed in the context of medical systems.

KOMPIUTERINIS MODELIAVIMAS

On the experimental investigation of investment strategies in the real and virtual financial markets

Jonas Mockus

Vilnius University, Institute of Mathematics and Informatics, Professor, Doctor Habil.
Vilniaus universiteto Matematikos ir informatikos instituto profesorius, habil. daktaras.
Akademijos str. 4, LT-08412 Vilnius
E-mail: jmockus@gmail.com

Igor Katin

Vilnius University, Institute of Mathematics and Informatics, Doctoral student
Vilniaus universiteto Matematikos ir informatikos instituto doktorantas
Akademijos str. 4, LT-08412 Vilnius
E-mail: igor@getweb.lt

Joana Katina

Vilnius University, Institute of Mathematics and Informatics, Doctoral student
Vilniaus universiteto Matematikos ir informatikos instituto doktorantė
Akademijos str. 4, LT-08412 Vilnius
E-mail: joana@getweb.lt

The optimal financial investment (Portfolio) problem was investigated by leading financial organizations and scientists. The aim of these works was to define the optimal diversification of the assets depending on the acceptable risk level.
The aim of the paper is to evaluate different investment strategies in the real and virtual financial markets. This aim is the new element of the proposed simulation system since optimization is performed in the space of investment strategies; both daily and long-term. A number of different investment strategies are presented, including the ones based on the Modern Portfolio Theory (MPT).
The simulated investment procedures include different prediction methods. The methods that minimize the mean absolute error (MAE) are added to the traditional ones that minimize the least squares error (MSE). The results of the virtual financial market are compared with historical data.
The model is designed as a tool to represent the behavior of an individual investor which wants to predict how the expected profit depends on different investment strategies using different forecasting methods of real and virtual stocks.

Introduction

The traditional approach is represented by the Modern Portfolio Theory (MPT). MPT was created by Harry Markowitz (1959) and William Forsyth Sharpe (1994). A number of investment organizations are making decisions using the software based on the theoretical results of Robert C. Merton and Myron S. Scholes (1972). The recent developments and applications of MPT are discussed in (Jack Clark Francis, 2013; Baker and Filbeck, 2013; Sortino, 2009). Some limitations of this theory have been noticed during the recent financial crisis when the investors experienced considerable losses (Krugman, 2009).

Financial market simulators are developed to satisfy the needs of small individual investors. The examples are the StockTrak global portfolio simulator, the MarketWatch, virtual stock exchange“, and the „Stock Simulator“ of Investopedia. Some banks offer their own investment simulators such as the Barclays Fantasy Investment Game. Users of these simulators working with „Virtual Stocks“ are informed about the results. The graphical user interfaces are friendly. However, the theoretical basis of these models and computing algorithms remains unknown. So the users cannot grasp the reasons why they win and why they experience losses.

To present the individual stockholders with a tool where everything is open is one of the aims of the PORTFOLIO model introduced in this paper. To accomplish this task, we deviate from the traditional portfolio models since this model is for defining the best investment strategies but not just the best diversification of assets what is a new element of this work.

In contrast to well-known results, this work not only simulates traditional results of utility and portfolio theories, but also complements them by various investment procedures and renders a possibility to users to develop and implement their own investment strategies.

Thus, the model can be useful to studies, to scientific collaboration, and to stockholders who are solving optimal investment problems and regard risk in an individual way. The market prediction and portfolio optimization were regarded in most of the financial market research. The investigation of different investment strategies is the feature of this work.

In Mockus and Raudys (2010); Mockus (2012), a preliminary investigation of the virtual stock exchange of a single stock is discussed. The results of these papers helped to initiate this work that simulates the optimal investment problem in the multi-stock market. Therefore we just refer to paper of Mockus (2012) for a mathematical description of the singe stock model and describe mainly the new resultas of experimental investigation. However, some definitions and expressions that describe the buying-selling strategies and investors‘ profit are repeated for convenience of reading.

1. The PORTFOLIO Model of a Virtual Financial Market

1.1 Buying and Selling Strategies

We consider a virtual market of I major players $i = 1,...,I$ and $j = 1,...,J$ stocks. The following notation is used:

$z(t, j) = z(t, i, j)$ is the price of stock j at time t, predicted by the player i,

$Z(t, j)$ is the actual* price at time t,

$U(t, j) = U(t, i, j)$ is the actual profit accumulated at time t by the player i buying-selling stock j,

$\delta(t, j)$ is the dividend of stock j at time t,

$\alpha(t)$ is the yield at time t,

$\gamma(t)$ is the interest rate at time t,

$h(t) = \gamma(t) - \alpha(t)$ is the haircut**,

$\beta(t, i)$ is the relative stock price change at time t as predicted by the player i

$$\beta(t, i, j) = (z(t+1, i, j) - Z(t, j)) / Z(t, j), \quad (1)$$

Expected profitability*** (relative profit) $p(t, i, j)$ of an investment at time t depends on the predicted change of stock prices $\beta_i(t, j)$, dividends $\delta_i(t, j)$, the bank rate $\alpha(t)$, and haircut $h(t)$

$$p(t, i, j) = \beta(t, i, j) - \alpha(t) + \delta(t) - h(t) = \\ = \beta(t, j) + \delta(t) - \gamma(t). \quad (2)$$

The aim is profit, thus a customer i will buy some amount $n_b(t, i, j) \geq n(t, j)$ of stocks j, if profitability is greater comparing with the relative transaction cost $\tau(t, n)$; $p(t, i, j) > \tau(t, n)$, and will sell stocks, if the relative loss (negative profitability $-p(t, i, j)$) is greater as compared with the transaction cost $p(t, i, j) < -\tau(t, n)$, or will do nothing, if $-\tau(t, n) \leq p(t, i, j) \leq \tau(t, n)$. Here the relative transaction cost is defined as the relation

$$\tau(t, n) = \frac{\tau_0}{n(t, j) Z(t, j)}, \quad (3)$$

* The term ‘actual’ means simulated

** In finance, a haircut is a part that is subtracted from the value of the assets that are being used as collateral. The size of the haircut reflects the perceived risk associated with holding the assets.

*** The term “profit” can define losses if negative terms prevail.

where τ_0 is the actual transaction cost and $n = n(t, j)$ is the number of transaction stocks. It follows from the equality $\tau(t,n) = p(t,i,j)$ that the minimal number of stocks to cover transaction expenses is

$$n(t, j) = \frac{\tau_0}{p(t,i,j)Z(t,j)}. \quad (4)$$

Therefore, the buying-selling strategy $S(t,i,j)$ of stock j by the customer i at time t in terms of profitability levels is as follows

$$S(t,i,j) = \begin{cases} \text{buy } n_b(t,i) \geq n(t,j) \text{ stocks,} \\ \quad \text{if } p(t,i,j) \geq \tau(t,n) \text{ and } n \leq n_b^{\max} \\ \text{sell } n_s(t,i,j) \geq n(t,j) \text{ stocks,} \\ \quad \text{if } p(t,i,j) \leq -\tau(t,n) \text{ and } n \leq n_s^{\max}, \\ \text{wait, if } |p(t,i,j)| \leq \tau(t,n^{\max}). \end{cases} \quad (5)$$

Here $n^{\max} = \max(n_b^{\max}, n_s^{\max})$, where $n_b^{\max}$ is the maximal number of stocks to buy, and $n_s^{\max}$ is the maximal number of stocks to sell.

If $n_b(t,i,j) = n_b^{\max}$ *and* $n_s(t,i,j) = n_s^{\max}$, (6)

then this buying/selling strategy reflects the behavior of risk-neutral stockholders which invest all available resources if the expected profitability is higher than the transaction cost. If the expected losses are greater, then all the stocks are sold. It means that stockholders may tolerate a considerable probability of losses, if the expected profits are positive. In this way, the maximal expected profit is provided. However, the probability to get losses instead of profits could be near to 0.5.

From expressions (1) and (2), the buying-selling strategy $S(t,i,j)$ in terms of stock price levels is

$$S(t,i,j) = \begin{cases} \text{buy } n_b(t,i,j) \geq n(t,j) \text{ stocks,} \\ \quad \text{if } Z(t,j) \leq z_b(t,n,i,j) \text{ and } n \leq n_b^{\max}, \\ \text{sell } n_s(t,i,j) \geq n(t,j) \text{ stocks,} \\ \quad \text{if } Z(t,j) \geq z_s(t,n,i,j) \text{ and } n \leq n_s^{\max}, \\ \text{wait, otherwise.} \end{cases} \quad (7)$$

Here the price level of the player i to buy at least $n = n(t, j)$ stocks at time t is

$$z_b(t,n,i,j) = z(t+1,i,j)/(1 - \delta(t) + \\ + \alpha(t) + h(t) + \tau(t,n)). \quad (8)$$

The price level of the player i to sell at least $n = n(t, j)$ stocks at time t is

$$z_s(t,n,i,j) = z(t+1,i,j)/(1 - \delta(t) + \\ + \alpha(t) + h(t) - \tau(t,n)), \quad (9)$$

where $z(t+1,i,j)$ is the stock j price predicted by the investor i at time $t+1$.

The market buying price at time t is the largest buying price of players $i = 1,...,I$

$$z_b(t,n) = z_b(t,n,i,j^{\max}),$$

where

$$i^{\max} = \arg\max_i z_b(t,n,i,j).$$

The market selling price at time t is the lowest selling price of players $i = 1,...,I$

$$z_s(t,n) = z_s(t,n,i,j^{\min}),$$

$$i^{\min} = \arg\min_i z_s(t,n,i,j).$$

1.2 Price Simulation

1.2.1 Buying-selling price

The market buying price of stock j at time t is the largest buying price of players $i = 1,...,I$

$$z_b(t,n,j) = z_b(t,n,i,j^{\max}),$$

where

$$i^{\max} = \arg\max_i z_b(t,n,i,j).$$

The market selling price at time t is the lowest selling price of players $i = 1,...,I$

$$z_s(t,n,j) = z_s(t,n,i,j^{\min}),$$

$$i^{\min} = \arg\min_i z_s(t,n,i,j).$$

The number of stocks j owned by the player i at time $t+1$ is

$$N(t+1,i,j) = \begin{cases} N(t,i,j) + n_b(t,n,i,j) \\ \quad \text{if } Z(t,j) < z_b(t,n,j), \\ N(t,i,j) - n_s(t,n,i,j) \\ \quad \text{if } Z(t,j) > z_s(t,n,j), \\ N(t,i,j) \quad \text{if no deal.} \end{cases} \quad (10)$$

Here $n_b(t,n,i,j)$ and $n_s(t,i,j)$ are the numbers of stocks *j* for buying and selling operations by the player i at time t.

1.3 Investors Profit

The product $N(0,i,j)\,Z(0,j)$ is the initial investment to buy $N(0,i,j)$ shares j using an investors‘ own capital at the initial price $Z(0,j)$. The initial funds to invest are $C_0(0,i)$ and the initial credit limit is $L(0,i)$. $L(t,i)$, $t=1,...,T$ is the credit available for a customer i at time t. The investors‘ own funds in cash $C_0(t,i)$ available for investing at time t are defined by the recurrent expression

$$C_0(t,i)=C_0(t-1,i)-\sum_j(N(t,i,j)-$$
$$-N(t-1,i,j)\;\;Z(t,j), \qquad (11)$$

where $t=1,...,T$. Here the product

$$(N(t,i,j)-N(t-1,i,j)\;\;Z(t,j)$$

defines the money involved in buying-selling stocks.

Stocks are obtained using both investor‘s own money $C_0(t,i)$ and the funds $b(t,i)$ borrowed at the moment *t*. The borrowed sum of the stockholder i accumulated at time t is

$$B(t,i)=\sum_{s=1}^{t}b(s,i)), \qquad (12)$$

The symbol $b(t,i)$ shows what the user i borrows at the moment t

Part of Profit by Stock j

In long-term investment strategies using the Sharpe ratio, the general profit should be divided between different stocks. Denote by $C_0(0,i,j)\le L(0,i,j)$ the initial funds to be invested in the stock $j=1,...,J$, where $L(0,i,j)$ is the initial credit limit for stock j and

$$\sum_j L(0,i,j)=L(0,i) \qquad (13)$$

For example, the initial funds may be divided into equal parts

$$\sum_j C_0(0,i,j)=C_0(0,i)/J \qquad (14)$$

The investors' own funds in cash $C_0(t,i,j)$, accumulated buying-selling stocks j and available for investing at time t in the stock j, are defined by the recurrent expression below

$$C_0(t,i,j)=C_0(t-1,i,j)-(N(t,i,j)-$$
$$-N(t-1,i,j)\;\;Z(t,j), \qquad (15)$$

where $t=1,...,T$. Here the product

$$(N(t,i,j)-N(t-1,i,j)\;\;Z(t,j)$$

defines the money involved in buying-selling stocks. j

Stocks are obtained using both investors‘ own money $C_0(t,i,j)$ and the funds $b(t,i,j)$ borrowed at the moment t. The borrowed sum of the stockholder i for the stock j accumulated at time t is

$$B(t,i,j)=\sum_{s=1}^{t}b(s,i,j)), \qquad (16)$$

where

$$\sum_j B(t,i,j)=B(t,i) \qquad (17)$$

The symbol $b(t,i,j)$ shows what the user i borrows at the moment t for the stock j.

The general borrowing expenses for stock *j* are

$$B_{sum}(t,i,j)=B(t,i,j)+\sum_{s=1}^{t}B(s,i,j)\;\;\gamma(s,i)), \qquad (18)$$

where the first term denotes the loan accumulated at time t, the second term shows the interest, and

$$\sum_j B_{sum}(t,i,j)=B_{sum}(t,i). \qquad (19)$$

The investor i gets a profit as the difference between the income from selling and buying stocks $D(t,i,j)$ and expenses for the borrowing funds $B_{sum}(t,i,j)$

$$U(t,i,j)=C_0(t,i,j)+D(t,i,j)-B_{sum}(t,i,j), \qquad (20)$$

where

$$D(t,i,j)=N(t,i,j)\,Z(t,j)-N(0,i,j)\,Z(0,j), \qquad (21)$$

The investor‘s i profit from the stock j at the end of investment period is denoted as

$$U_{i.j}=U(T,i.j), \qquad (22)$$

where

$$\sum_j U_{i.j}=U_i, \qquad (23)$$

If for some reason equalities (17), (19), and (23) are violated, then the normalization of components may be applied to restore them.. the funds invested in buying stocks j at time $T.x_j=$ The bank profit expressions are the same as in the single stock market model of Mockus (2012).

1.4 Multi-Level Operations

1.4.1 Strategy No. 1, ordinary stockholder: selling and buying by profitability levels

In the opinion of some professional brokers we have interviewed, one needs at least three buying profitability levels $p_b(t,i,j,l),\ l=1,2,3$, where

$$p_b(t,i,j,l+1) > p_b(t,i,j,l),\ \ p_b(t,i,j,1) = \tau(t), \quad (24)$$

and three selling profitability levels

$p_s(t,i,j,l),\ l=1,2,3$, where

$$p_s(t,i,j,l+1) < p_s(t,i,j,l),\ \ p_s(t,i,j,1) = -\tau(t),$$
$$p_b(t,i,j,l) > p_s(t,i,j,1), \quad (25)$$

to explain the behavior of major stockholders. The level $l=1$ means to buy-sell the minimal number of stocks. The level $l=3$ means to buy-sell as much stocks as possible, and the level $l=2$ is in the middle. Details of multi-level operations are in Mockus (2012).

2. Price Prediction

In this model, two versions of Autoregressive (AR(p)) and two versions of Autoregressive Moving Average (ARMA(p,q)) models are considered for stock rate predictions. The first versions – AR(p) and ARMA(p,q) – use traditional least squared approach, the second ones – AR-ABS(p) and ARMA-ABS(p,q) – minimize the absolute errors. The development and implementation of ARMA-ABS(p,q) is a new feature of this work.

The actual price of a stock at time $t+1$ is defined as the price of the previous deal of major stockholders plus the truncated Gaussian noise that represents the remaining small stockholders. Thus, the actual stock j price at time $t+1$ determined by buying-selling operations of the stockholder i is as follows

Here a reflects the market inertia, where $a = 0$ means there is no inertia and $a = 1$ describes maximal inertia (no market reaction to the last deal).

2.0.2 Strategy No. 2, risk-aware stockholders: selling all unprofitable stocks – buying the best ones

First, the stockholder i sells all the non-profitable stocks

$$p_s(t,i,j) \le -\tau(t,i,j), \quad (27)$$

and then invests all the available funds to buy the most profitable stock. The stockholder i does not sell the stock j, if the expected loss is smaller than the transaction cost $|\,p(t,i,j)\,| < \tau(t,i,j)$.

This selling strategy reflects risk-aware users which keep some less profitable stocks to avoid possible losses if predictions happen to be wrong.

2.0.3 Strategy No. 3, risk-neutral stockholders: buying the best stocks and selling all the rest

The risk-neutral stockholders use all available resources to buy the stock $j^{\max}$ which provides the highest expected profit:

$$j^{\max} = \arg\max_j p(t,i,j), \quad (28)$$

2.0.4 Strategy No. 4, risk-averse stockholders: selling and buying in proportion to profitability

Denote by J_+ a set of stocks with a positive profitability and by J_- the stocks with a negative profitability. Denote $J_b = |\,J_+\,|$ and $J_s = |\,J_-\,|$.

$$j_+^{\max} = \arg\max_{j\in J_+} p(t,i,j), \quad (29)$$

and

$$j_-^{\min} = \arg\min_{j\in J_-} p(t,i,j). \quad (30)$$

First, we sell stocks in proportion to $l = 1,\ldots,j_-^{\min}$ selling profitability levels $p_s(t,i,l) = p(t,i,j=l)$,

$$Z(t+1,i,j) = \begin{cases} (1-a)z_b(t,i,j^{\max}) + aZ(t,j^{\max}) + \varepsilon(t+1), & \text{if } p(t,i,j^{\max}) > 0, \\ (1-a)z_s(t,i,j,1) + aZ(t,j) + \varepsilon(t+1,j), & \text{if } p(t,i,j) \le p_s(t,i,j,1) = -\tau(t,j), \\ & \text{and } p(t,i,j) > p_s(t,i,j,2), \\ (1-a)z_s(t,i,j,2) + aZ(t,j) + \varepsilon(t+1,j), & \text{if } p(t,i,j) \le p_s(t,i,j.2), \\ & \text{and } p(t,i.j) > p_s(t,i,j,3), \\ (1-a)z_s(t,i,j,3) + aZ(t,j) + \varepsilon(t+1,j), & \text{if } p(t,i,j) \le p_s(t,i,j,3), \\ Z(t,j) + \varepsilon(t+1,j), & \text{if no deal.} \end{cases} \quad (26)$$

$l = 1,..., j_-^{\min}$. Then we use all accumulated resources to buy stocks in proportion to $l = 1,..., j_+^{\max}$ profitability levels

$$p_b(t,i,l) = p(t,i,j=l),\ l = 1,..., j_-^{\max}.$$

Strategy No. 5, Long Term Investment

In the previous sections, we analyzed short-term investing by daily decisions using different investment strategies. So, the search was in the strategy space. In this section, we consider the maximization the Sharp Ratio (Sharpe (1994)).

In the long term investing models, the time series are split into learning and testing sets. In the learning stage, the mean and variance of portfolio $P(x)$ profit are estimated using the first part of observations $1 \le t_0 < T$, where $x = (x_j,\ j = 1,..., J)$. Usually t_0 is about $T/2$. and the initial funds are equally divided among the stocks, meaning that $x_j^0 = C_0(0)/J$. Here $C_0(0)$ denotes initial funds of a single user.

Note that in virtual markets, the stock prices are generated by the interaction of different virtual investors. The search for the optimal distribution of funds is performed by maximizing the Sharpe Ratio. During the testing stage the profits of optimized portfolios are calculated using the remaining observations $s\ :\ t_0 < s \le T$.

The data of the learning stage are used to estimate average deviations and variances. The sample mean of portfolio $P(x)$ that contains stocks with weights $x_j,\ j = 1,..., J$ is as follows

$$m^u(x) = \sum_j x_j / x_j^0 \frac{1}{t_0} \sum_{t=1}^{t_0} U(t,j). \tag{31}$$

Here $U(t,j)$ follows from (20) by omitting the investor's index i. The estimator of variance of the portfolio $P(x)$ is

$$(s^u(x)\ ^2 = \frac{1}{t_0 - 1} \sum_j \sum_k \sum_{t=1}^{t_0} x_j / x_j^0\ (U(t,j) - - m^{u_j})\ x_k / x_k^0\ (U(t,k) - m^{u_k}). \tag{32}$$

We maximize the Sharpe ratio in this specific form

$$\max_x \frac{m^u(x)}{s^u(x)}, \tag{33}$$

because the risk free asset is regarded separately.

3. Experimental Results

The general aim of the experiments is to evaluate the profitability of different investment strategies in both the virtual and real markets, which involve well-known companies. In the experiments, the profits and average prediction errors of eight players by four different investmrnt strategies using eight prediction models each: *AR*(1), *AR*(3), *AR*(6), *AR*(9), *AR – ABS*(1), *AR – ABS*(3), *AR – ABS*(6), and *AR – ABS*(9) were tested. The time period is 360 days (virtual working days). This represents approximately 18 months of real time. The average daily and final values are estimated by 100 samples. Four investment strategies were tested in both the virtual and real environment including eight stocks of major companies.

3.1 Historical and Virtual Data

In this section, different investment strategies are investigated using historical data obtained automatically using the Yahoo data base. The historical prices of the following eight stocks were used: Microsoft (MSFT), Apple (AAPL), Google (GOOG), Nokia (NOK), Toyota (TM), Bank-of-America (BAC), Boeing (BA), and Oracle (ORCL). The time series covers about two years of rapid financial development (2008–2009) and include stocks eight major financial and technological organizations.

According to Figure 2, the maximal profit of $20,000 was achieved using the first strategy by the prediction model *AR*(6). Figure 1 shows average profits of eight prediction models by the first strategy using virtual data.

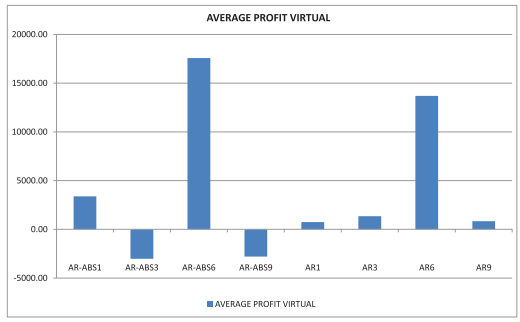
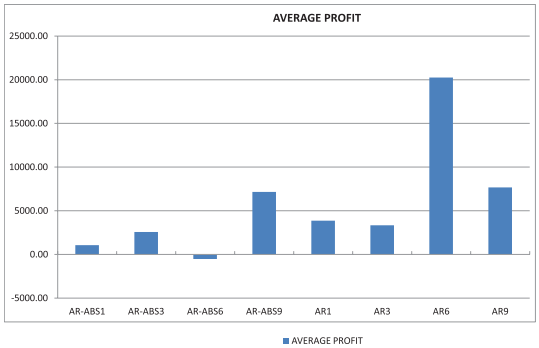
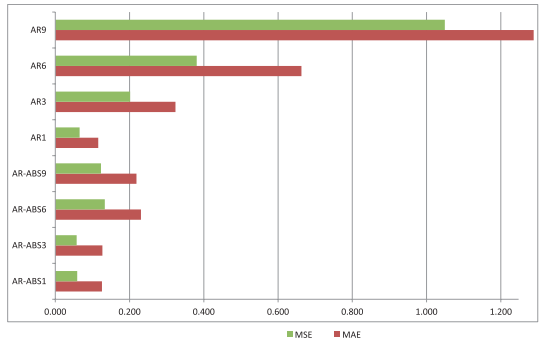


Fig. 1. ***Average profits of eight prediction models by the first strategy using virtual data***



F i g. 2. ***Average profits of eight prediction models by the first strategy using historical data***



F i g. 3. Aaverage prediction errors using different prediction models

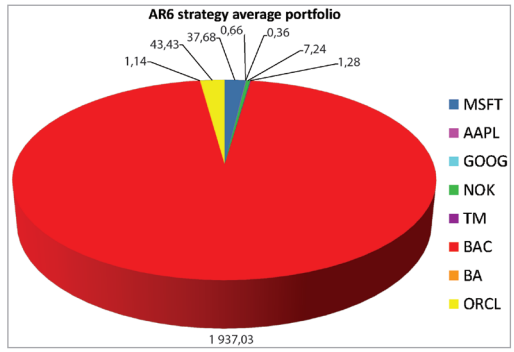
Figure 2 shows average profits of eight prediction models by the first strategy using historical data. Comparing Figures 1 and 2 shows that the profits and their distribution between different prediction models are different in real and virtual markets. The explanation is that the real data represent mainly the post-crisis period with gradual recovery of stock prices. In contrast, the virtual data were obtained assuming stable general conditions. However, a two year cycle of stock prices and profits was observed, in some experiments.

3.1.1 The Best Investment Strategy

Figure 3 shows average prediction errors using different prediction models. Figure 4 shows the most profitable portfolio defined using the first strategy and *AR*(6) prediction model. The dominant stock vas Banc-of-America (No.6), the second one was Oracle (No. 8) followed by Microsoft (No. 1).

The success of BAC can be explained by the rapid recovery of its stocks after the deep fall during depression.

Comparing average errors in Figures aapl-price.eps with profits in Figure prof-days-h1.eps we see that the minimal prediction error is provided by the model *AR*(1) and the maximal profits are achieved using the *AR*(6) model. This and other experiments indicate that minimal prediction errors do not necessarily provide maximal profits.



F i g. 4. ***The most profitable portfolio defined using the first strategy and AR(6) prediction model***

4. Concluding Remarks

The Game Theory is a suitable framework to model financial markets because the future market price of financial assets depends on predictions (and subsequent actions) of the market participants with conflicting interests.

The proposed financial market model PORTFOLIO is designed as a tool for simulating market processes in response to different changes of market parameters and for estimating the expected profits of different investment strategies using both the virtual and historical data. Convenient user interactions are provided by implementing the model as a Java applet and publishing it in an open web-site (Mockus, 2013).

A single-stock Stock Exchange Game Model (SEGM) was introduced in Mockus (2012) to simulate the behavior of several stockholders.

Since PORTFOLIO may be too simplistic for practical investing, it can serve as a useful tool for studies of market behavior by providing an easy way of simulating different scenarios of player strategies. For example, simulations can explain stock market reaction to deliberately set non-NE strategies of a major stockholder, such as manipulation of asset prices, designed to lower their value.

Thus, the PORTFOLIO model helps students of business informatics to understand better financial disasters that we are witnessing at present. The unexpected new result is the observation that the investment strategies using prediction models with minimal errors did not provide maximal profits.

REFERENCES

BAKER, HK; FILBECK, G. (2013). *Portfolio Theory and Management*. Oxford University Press, Oxford, UK.

BLACK, F.; SCHOLES, M. (1972). The pricing of options and corporate liabilities. *Journal of Political Economy, no.* 81, p. 637–654.

JACK CLARK FRANCIS, DK (2013). *Modern Portfolio Theory: Foundations, Analysis, and New Developments, + Website*. New York: Wiley.

KRUGMAN. P. (2009). How did economists get it so wrong? *The New York Times*, September 2.

MARKOWITZ, H. (1959). *Portfolio Selection: Efficient Diversification of Investments*. New York: Wiley.

MOCKUS, J. (2012). On simulation of the Nash equilibrium in the stock exchange context. *Informatica*, no. 23, p.77–104.

MOCKUS, J. (2013). PORTFOLIO. http://optimum2.mii.lt/igor/index.html/.

MOCKUS, J.; RAUDYS, A. (2010). On the efficient-market hypothesis and stock exchange game model. *Expert systems with applications*, vol. 37, p. 5673–5681.

SHARPE, WF. (1994). The Sharpe Ratio. *The Journal of Portfolio Management*, no. 21, p. 49–58.

SORTINO, F. (2009). *The Sortino Framework for Constructing Portfolios Focusing on Desired Target Return to Optimize Upside Potential Relative to Downside Risk*. Amsterdam: Elsevier Inc., Netherlands.

APIE EKSPERIMENTINĮ INVESTAVIMO STRATEGIJŲ TYRIMĄ REALIOSE IR VIRTUALIOSE FINANSŲ RINKOSE

Jonas Mockus, Igor Katin, Joana Katina

S a n t r a u k a

Darbo tikslas yra įvertinti įvairias investavimo strategijas pagal jų pelningumą realiose ir virtualiose finansų rinkose.

Darbe aprašytas atnaujintas finansų rinkos modelis, analizuojami eksperimentinių skaičiavimų rezultatai.

Critical analysis and elaboration of three prevailing approaches to model quality of service

Audronė Lupeikienė

Vilnius University, Institute of Mathematics and Informatics, Senior Researcher, Assoc. Professor, Doctor
Vilniaus universiteto Matematikos ir informatikos instituto daktarė
Akademijos str. 4, LT-08663 Vilnius
E-mail: audrone.lupeikiene@mii.vu.lt

Jolanta Miliauskaitė

Vilnius University, Institute of Mathematics and Informatics, Doctoral student
Vilniaus universiteto Matematikos ir informatikos instituto doktorantė
Akademijos str. 4, LT-08663 Vilnius
E-mail: jolanta.miliauskaite@mii.vu.lt

Albertas Čaplinskas

Vilnius University, Institute of Mathematics and Informatics, Principal Researcher, Head of Department, Professor, Doctor (HP)
Vilniaus universiteto Matematikos ir informatikos instituto daktaras (HP)
Akademijos str. 4, LT-08663 Vilnius
E-mail: albertas.caplinskas@mii.vu.lt

The service-oriented architecture (SOA) becomes dominant paradigm in the development of web-based systems. The paper critically analyses three prevailing approaches to SOA quality of service (QoS) modelling, namely, SQuaRE based Web Services Quality Model, OASIS Quality Model for Web Services and an ontology based QoS modelling approach proposed by Beanchini and co-authors. The main contribution of this paper is that it highlights both conceptual similarities and differences of the analysed approaches, examines their shortcomings and the reasons for their inability to model QoS adequately in service-oriented enterprise systems context. It also gives arguments for the need to develop a new, holistic QoS modelling approach.

Introduction

Software system is typically thought of in terms of its functionality. However, the functionality does not completely describe the system. In order to specify software system non-functional properties must also be defined. The term *Quality of Service* (QoS) is used to refer to the non-functional properties of a software service. According to (Ludwig, 2003) quality of service is expressed by observable parameters relating to the non-functional properties.

Quality of service (QoS) in service-oriented enterprise information systems is key factor for a number of reasons (Aiello, Giorgini, 2004): 1) autonomous services depend on one another; 2) services can compete one another – a service consumer's decision on a service may be based on its QoS properties; 3) a service provider can offer the same functionality but different qualities, and different qualities of the same service should be declared. However, common consensus on QoS in general is not achieved, as well as on QoS concept in service-oriented enterprises.

For example, the view inherited from computer network community considers QoS in terms of performance and availability of service only. Some researchers assume that any custom characteristic that can be modelled as non-functional service requirement may be considered as a constituent of quality of service (Aiello, Giorgini, 2004). In addition, there are different understandings of the nature of quality in general and different points of view on software quality. A number of new issues of QoS are raised by service-oriented computing, service-oriented architecture (SOA), and service-oriented enterprise systems:

- quality of service parameters should be defined from different stakeholders' perspectives;
- many requestors access services via Internet which is not under control of service owner;
- business processes are supported by composite services, so it is necessary to understand how QoS properties of the constituent services contribute to the overall quality of composite ones.

The first step to solve these issues is to understand and define precisely the concept *quality of SOA service*. The analysis of different understandings of QoS, systematisation and generalisations of these understandings is the necessary prerequisite to get the required result. The next step towards defining service quality should take into account different human roles in different service life cycle phases. These actors are individuals, solve different problems, have diverse knowledge, perceive quality differently, and express different preferences. The notions of views, viewpoints, and perspectives (Leite, 1988) can be employed as basic tools in communicating about service quality. A viewpoint describes from where an actor is looking – a standing or mental position used when observing or specifying quality of service. A perspective is a set of facts observed or modelled from a particular viewpoint and according to a particular aspect. A view is an integration of perspectives.

This paper shows the diversity of points of view on quality of service and critically assesses service quality modelling approaches. Short analysis and evaluation of three prevailing approaches to model quality of service is intended to highlight their similarities and differences, and give special attention to their inability to model QoS adequately in service-oriented enterprise systems context. It also presents reasons for the need of an alternative holistic QoS modelling approach.

The remainder of this paper is organised as follows: section 1 considers web service and SOA service concepts and their similarity, short analysis and evaluation of representative quality of service modelling approaches is presented in section 2. Section 3 provides related works and finally we conclude with a summary and remarks on the state of affairs on the field.

1. Web services vs SOA services

During the past few years "service" and "service oriented architecture" are the most exploited concepts in the domain of enterprise systems engineering. However, these concepts are being interpreted differently in different contexts. The analysis of definitions and classification of services puts some light on this ambiguity.

Categorising services by their visibility public and private services are identified. European Interoperability Framework defines public services as services "developed primarily for direct use by the public administration that created them, or by their direct customers, i.e. businesses and citizens" (EIF, 2010). A public service can be accessed by any potential consumer, a private service has restricted accessibility. In order to access such service it is required to contact the provider's organisation. Categorising services by their business value it is important their relations with business tasks or processes. In addition, any activity depends on the organisation's competencies. For example, (ERL, 2005) makes a distinction between application, business and process services. Such the services, i.e. services

in an enterprise or, to be more precise, in an enterprise information system context, we discuss in this paper. The term "SOA service" is used by many authors to indicate this type of services.

SOA is the architectural style for crafting distributed business applications when the quick response to the endless changes in the business environment is needed. The SOA has several substyles (Tang et al., 2010). One among most popular substyles is web-based SOA. Web-based SOA is SOA that is implemented using web service standards called WS-*. The SOA based on Representation State Transfer (REST) architectural style and enterprise Web 2.0 is another example of SOA substyle. Therefore, web services is a technology used to implement service oriented architecture.

The concept "service" in SOA context can be differently interpreted. OASIS Quality Model for Web Services (Kim, Lee, 2005) defines service as the means by which the needs of a consumer are brought together with the capabilities of a provider. This definition puts emphasis on the interaction over a network (Fig. 1 a). In other words, web service plays role of a standard means of interoperating between different software applications (W3C, 2004).

According to SOA Reference Architecture specification (SOA, 2012) web service is "a mechanism to enable access to one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description". It means that web services are used to wrap the distributed components and to implement service providers' interfaces, i.e. service provider is encapsulated to web service (Fig. 1 b). The purpose of using the capability is to produce the real world effect. So, in this context services and web services can be seen as synonyms, and web service quality models, whose authors use this second web service interpretation, can be also analysed as SOA service quality models.

2. Quality of service modelling approaches: analysis and evaluation

The analysis of various service quality concepts and models showed that there is a big quantity and variety of interpretations of these concepts. Three representative of quality of services modelling approaches are selected and discussed in this paper:

- Web services quality modelling approach proposed in (Abramovicz et al., 2008) stands for the approaches based on the ISO/IEC Software Product Quality Requirements and Evaluation Model (ISO/IEC, 2005).
- Web services quality modelling approach proposed in (OASIS, 2012) stands for the approaches based on the activity-based separation of quality concerns.
- Quality of services modelling approach proposed in (Bianchini et al., 2004) stands for the approaches based on the service ontology (Guo, 2011; Dobson et al., 2005).

The choice of representatives also demonstrates the scheme for classification of QoS models. It expresses different points of view on QoS modelling.

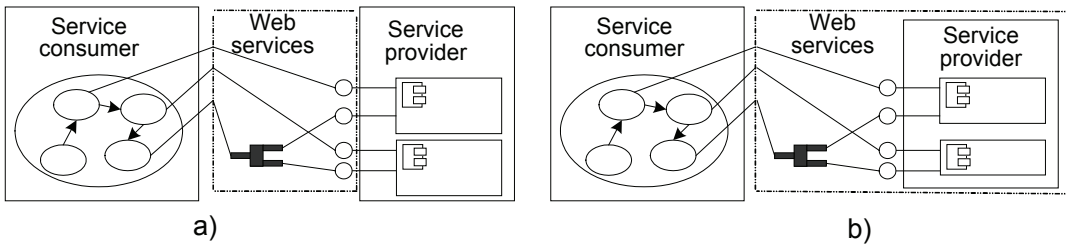


Fig. 1. ***The roles of web services***

The group of researches (Abramovicz et al., 2008), (Abramovicz et al., 2009) suggest that web services quality model should be compatible with traditional software quality model. For this moment, software quality model ISO/IEC 9126 (ISO/IEC, 2001) is widely used, but it is being replaced with ISO/IEC 25010 (SQuaRE) model (ISO/IEC, 2005). The reason for web services and software models compatibility is the following – "definition of quality requirements starts from the same set of requirements both for Web Service and software module" (Abramovicz et al., 2008). In addition, if web service can be replaced by software module then quality requirements for this service should be analogous to the ones of traditional software.

Web services quality model proposed in (Abramovicz et al., 2008) is based on SQuaRE model and consists of three parts: external web service quality, internal web service quality, and web service quality in use. External quality is capability of a web service to provide the effects satisfying needs when this service is used under specific conditions. In other words, external quality characterise "black box" behaviour. Internal quality gives "white box" view to quality. The quality characteristics are the same for both external and internal qualities – security, interoperability, reliability, usability, efficiency, maintainability, and portability. Quality in use defines quality as utility for a specific user to achieve its specific goals in specific context. Quality in use covers usability in use, context in use, safety in use, security in use, support in use, and adaptability in use. External and internal web service qualities are the viewpoints for service provider, while quality in use – of web service consumer.

It should be noted that only the top-level quality characteristics were listed above. Quality model proposed in (Abramovicz et al., 2008) uses three levels – main characteristics can be decomposed to sub-characteristics and those – to quality measures.

In summary, the main shortcoming of this QoS model is that is not enough takes in account the specifics of services. Already (Reeves et al., 1994) pointed out that although services and products share many similarities, they differ in a number of ways: 1) services are intangible, cannot be stocked, and their attributes are difficult to demonstrate (intangibility); 2) services are heterogeneous and it is their fundamental characteristic, because results of service varies from day-to-day or from customer-to-customer and of this reason it is hard to standardise their quality (heterogeneity); 3) services are inseparable, because to a large extent they are simultaneously produced and consumed (inseparability); 4) services are extremely perishable, that is, they have zero inventory, cannot be saved for later use, can be used only once else they perish and once sold, they stand sold and cannot be returned (perishability). In addition, a service is a process rather than a thing and consumer's involvement in the production of many services creates additional quality control difficulties for managers. In research literature, the abovementioned four service characteristics are usually referred to as IHIP characteristics (Moeller, 2010). Although some criticism exists whether services are really different from goods and whether the IHIP characteristics are characterising services, today, the service concept is operationalised mainly through these characteristics (Edvadsson, 2005). Even more complicated is the question whether or not existing service concepts, including IHIP characteristics and definitions of quality, are applicable to Internet services. Conflicting opinions exist on these issues. For example, (Moeller, 2010) argues that "The characteristics of intangibility, heterogeneity, inseparability, perishability (IHIP) that have been regularly applied to services have been subjected to substantial criticism, as more and more exceptions occur. The reasons for the criticism are twofold. The focus of services marketing has changed and the development of information and communication technology has advanced dramatically."

Edvardsson (2005) and many other researchers advocate also that technology-based services are, in fact, storable, repeatable, often standard-

ised and last, but not least, the service production does not involve any direct interactions with humans. On the other hand, (Hofacker, 2007) states that e-services are less tangible as traditional services, possible, more heterogeneous, taking into account instability of hardware, software and network environment, highly flexible in terms of physical separation between consumer and producer, and can be stored indefinitely by the provider (on server disk) or user. Abramowicz and his colleagues ignore the above-mentioned specific of services as well as service related business issues.

These issues are considered in some other QoS models including OASIS Committee Specification (OASIS, 2005; OASIS, 2012). It points out that web services differ from installation-based software. The differences cause the distinct web service quality model and characteristics. First of all, service consumer and provider as a rule belong to different ownership domains and relationships between their instances can be established ad-hoc. This includes possibility of web client to dynamically change the server. The changes can also be done in real time when quality is not sufficient. Secondly, the quality of web services depends on the run-time environment. Consequently, variation of service quality can occur. Thirdly, service consumer must tolerate some acceptable deviation of required quality because it may be not obtainable.

OASIS' web service quality model consists of three components: quality factors, quality associates, and quality activities (OASIS, 2005). A quality factor is a group of characteristics, which represent web service's properties. A quality associate is the person or organisation (in other words, role) related to web services life cycle stages. The quality activity refers to various actions performed by associates to ensure web services quality and its stability. OASIS Specification emphasis on quality model establishment from the view of service but not product quality. It implies different views of using a service; so, quality can be considered in different layers: user's view layer, interoperability view layer, and management and security view layer (OASIS, 2005).

OASIS (2012) divides web service quality factors* into two groups – business quality group and system quality group. Business quality group enables evaluating the business value of web services, i.e. the economic worth delivered by applying these services on a business (OASIS, 2012). The business value depends on quality subfactors such as price, penalty and incentive, business performance, service recognition, service reputation, and service provider reputation. OASIS Specification notifies that in addition to those factors business benefit, profit and return of investment can be included to this group. System quality group consists of variant quality part and invariant quality part. The values of quality factors of variant quality subgroup can be dynamically varied in run-time, while the values of quality factors of invariant quality subgroup can be determined immediately after the service development process is completed. This subgroup includes interoperability, business processing quality, manageability and security. OASIS (2012) points out that dynamically vary the values of response time, maximum throughput, availability, accessibility, and successability.

The activity-based layered approach to quality model gives good separation of quality concerns. The main shortcoming of OASIS Web Service Model is that it does not consider domain specific nature of some quality characteristics. It should be also noted here, that, first, as a rule more layers gain attention in SOA enterprise context, i.e. model gives poor views on quality. Second, layers are related with each other – the layer may be related directly with any of the layers above or only with the layer below it. However, these two aspects are not discussed in (OASIS, 2005; OASIS, 2012). Thus, there is a need to develop framework that integrates all viewpoints and perspectives on service quality at a higher abstraction level.

---

* The concept *quality factor* is broader than *quality attribute* in the other papers; it includes dimensions and measures of the quality.

Domain specific nature of quality characteristics is considered at the top level of QoS model proposed in (Bianchini et al., 2004). The authors argue that ontology-based approach should be used to categorise services. The proposed ontology classifies services on the basis of their functional characteristics enriched with a quality of service model. It provides three abstraction levels – subject categories, abstract services, and concrete services. Concrete services stand for implemented services and can be accessed by service consumer. For each quality characteristic a range of values it can guarantee should be provided. Similar concrete services are grouped into clusters*. Abstract service represents the cluster of similar concrete services. Subject categories represent application domains.

The subject categories are organised into *is-a* hierarchies and give the possibility to classify abstract services. In addition, even at the top level (i.e. inside of subject categories) quality characteristics are divided into generic and domain-specific characteristics (Bianchini et al., 2003). The authors propose to use as generic quality characteristics a set of quality parameters provided by the international standard ISO 8402 (part of ISO 9000 standard), which describes quality of any type of service. Examples are reliability, response time, and latency. Examples of domain specific characteristics include payment mode, type of credit card accepted, and legal constraints. So, a domain, a set of admissible values, the measure units, and a set of rules to convert value from one to the other measure form a description set for quality characteristics.

Quality of services characteristics are added to the all three abstraction layers. Quality characteristics in the category layer are associated to each leaf of the defined taxonomies. If an abstract service is associated to more than one subject category, it inherits the union of corresponding quality characteristics sets. A concrete service from the similarity cluster inherits its quality characteristics of corresponding abstract service, and the range of its values is defined by the set of admissible values for the quality characteristics for this abstract service.

The ontology based QoS model enables dynamic definition of quality characteristics because of possibility to specialise the top-level concepts. So, the specifics of particular domain can be made available by defining an abstract service quality and its characteristics. The shortcomings of this model are: 1) it purports to be for service classification only, and 2) like the other above discussed models it ignores quality of experience (QoE) (Ullah et al., 2012).

3. Related works

The reviews of various service quality models are presented in (Seth et al., 2005; Benbernou et al., 2010), the critical analysis of these models can also be found in many papers including (Frankova et al., 2006; Kuyoro et al., 2012). The meta-level analysis of quality models is presented in (Shekhovtsov, 2011).

Seth et al. (2005) examine 19 different service quality models. The review shows that neither some model of service quality is commonly accepted, nor some operational definition of how to measure service quality is commonly approved. Its authors conclude that service quality outcome and measurement depends on the type of service, situation, time, and other factors. In (Benbernou et al., 2010) the review of the main approaches that have been proposed in the literature is presented. This survey also revealed the lack of a well-established and standard QoS model for services. It pointed out that the majority of QoS models for services were proposed by the Web service community and describe only technical attributes. In addition, authors conclude that most of the models lack the richness needed in specifying the QoS of different types of services.

* A hierarchical clustering algorithm is used to form the similarity sets (Bianchini et al., 2004).

To specify the provided or required quality of service some language is necessary. WSDL (W3C, 2001) and its extensions as a rule are used in SOA and web services context. The conceptualisations behind the languages can be quite different (Čaplinskas et al., 2003). Consequently, WSDL is associated with the particular service quality model based on the particular conceptualisation and ontology. The group of researches from University of Trento (Frankova et al., 2006) pay attention to the main idea of such a language – it provides syntax to define terms, which refer to non-functional properties of operations. However, this ties quality of service to its individual operations, not to the service as a whole.

The framework giving base for comparing different software quality models, more exactly – quality conceptualisations* behind these models, is defined in (Shekhovtsov, 2011). The above cited conclusion, that service quality outcome and measurement depends on the type of service, situation, and other aspects, motivates the need for further research to answer question on the applicability of this framework for the comparison of different SOA service models.

Conclusions and remarks

The critical analysis of different approaches to QoS modelling demonstrates that these approaches can roughly be divided into these categories: taxonomy-based, activity-based and ontology-based ones. Although all the analysed approaches are intended to be used in different service life cycle phases and for different purposes, they have some conceptual similarities. They do not take into account differences caused by business-to-business, business-to-consumer and utility service models and completely ignore the QoS specifics in the enterprise systems context. Besides, they are predominantly technology-oriented.

On the other hand, SQuaRE based Web Services Quality Model, OASIS Quality Model for Web Services and an ontology based QoS modelling approach proposed by Beanchini and co-authors conceptualise quality of services quiet differently They model QoS using different quality characteristics, group these characteristics to different categories, and organise them in different ways forming sets, trees, taxonomies, etc. Each of these approaches focuses on different viewpoints and perspectives on service quality and any approach do not relate them to each other. It means that it is not possible to balance conflicting service quality requirements among different stakeholders of an enterprise system applying QoS modelling approach of any mentioned category. For this reason, a holistic modelling approach, for example, viewbased approach that uses viewpoints and perspectives to structure QoS models is required. It is a solvable, but enough complicated problem. The development of such approach is the subject of our further research.

Finally, to the best of our knowledge, any of current QoS modelling approaches does not define the meaning of QoS at fundamental, theoretical level. Generally they are ad hoc ones. So, further research on this subject is required. The present paper gives deeper understanding of this subject and can be considered as a step towards this goal.

Acknowledgments

This work has been supported by the project VP1-3.1-ŠMM-08-K-01-010 "Theoretical and Engineering Aspects of e-Service Technology Development and Application in High-Performance Computing Platforms", Task 1.1.1 Investigation, generalisation and improvement of QoS prediction for Internet services, funded by the European Social Fund.

* In this paper a conceptualisation of quality is thought of as "representation of the system quality on the same level of abstraction as the conceptual model of the functionality of this system" (Shekhovtsov, 2011).

REFERENCES

ABRAMOWICZ, W.; HOFMAN, R.; SURYN, W.; ZYSKOWSKI, D. (2008). SQuaRE based web services quality model. In: *Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS 2008)*, Vol. I, 19–21 March 2008, Hong Kong, p. 827–835.

ABRAMOWICZ, W.; HANIEVICZ, K.; HOFMAN, R.; KACZMAREK, M.; ZYKOWSKI, D. (2009) Decomposition of SQuaRE based requirements for the needs of SOA applications. In: AO, S.-I.; HUANG, X.; WAI, P.-K. A. (eds.) *Trends in Communication Technologies and Engineering Science*, Springer, p. 81–94.

AIELLO, M.; GIORGINI, P. (2004). Applying the Tropos methodology for analysing web services requirements and reasoning about qualities of services. *Upgrade*, vol. V, no. 4, p. 20–26.

BENBERNOU, S.; BRANDIC, I.; CAPPIELLO, C.; CARRO, M.; COMUZZI, M.; KERTESZ, A.; KRITIKOS, K.; PARKIN, M.; PERNICI, B.; PLEBANI, P. (2010) Modeling and negotiating service quality. In: Papazoglou, M.; Pohl, K.; Parkin, M.; Metzger, A. (eds.) *Service Research Challenges and Solutions for the Future Internet*. Springer, p. 157–208.

BIANCHINI, D.; De ANTONELLIS, V.; MELCHIORI, M. (2003) E-service classification techniques to support discovery in a mobile multichannel environment. In: *Proceedings of 4th International Conference on Web Information Systems Engineering (WISE 2003)*, 10–12 December 2003, Rome, Italy, IEEE Computer Society, p. 252–259.

BIANCHINI, D.; De ANTONELLIS, V.; MELCHIORI, M. (2004) QoS in ontology-based service classification and discovery. In: *Proceedings of the 15th International Workshop on Database and Expert Systems Applications* (DEXA'04), 30 August – 3 September 2004, Zaragoza, Spain, IEEE Computer Society, p. 145–150.

ČAPLINSKAS, A.; LUPEIKIENĖ, A.; VASILECAS, O. (2003) The role of ontologies in reusing domain and enterprise engineering assets. *Informatica*, vol. 14, no. 4, p. 455–470.

DOBSON, G.; LOCK, R.; SOMMERVILLE, I. (2005) QoSOnt: a QoS ontology for service-centric systems. In: *Proceedings of the 31st EUROMICRO Conference on Software Engineering and Advanced Applications (EUROMICRO-SEAA 2005)*, 30 August – 3 September 2005, Porto, Portugal. IEEE Computer Society, p. 80–87.

EDVARDSSON, B. (2005). Service quality: beyond cognitive assessment. *Managing Service Quality*, vol. 15, no. 2, p. 127–131.

EIF (2010) *European Interoperability Framework (EIF) for European Public Services*. COM(2010) 744 final, European Commission. 35 p.

ERL, T. (2005) *Service-Oriented Architecture: Concepts, Technology and Design*. Prentice Hall.

FRANKOVA, G.; MALFATTI, D.; AIELLO, M. (2006) Semantics and extensions of WS-agreement. *Journal of Software*, vol. 1, no. 1, p. 23–31.

GUO, G. (2011) A method for semantic web service selection based on QoS ontology. *Journal of Computers*, vol. 6, no. 2, p. 377–386.

HOFACKER, CH. F.; GOLDSMITH, R. E.; BRIDGES, E.; SWILLEY, E. (2007) E-services: a synthesis and research agenda. *Journal of Value Chain Management*, vol. 1, no. 1/2, p. 13–44.

ISO/IEC (2001) *ISO/IEC 9126-1:2001 Software Engineering – Product Quality*. International Standardisation Organisation.

ISO/IEC (2005) *ISO/IEC 25000-2005 Software Engineering – Software Product Quality requirements and Evaluation (SQuaRE)*. International Standardisation Organisation, JTC 1/SC 7.

KIM, E.; LEE, Y., eds. (2005) *Quality Model for Web Services (WSQM-2.0)*. OASIS, September 2005, http://www.clip.dia.fi.upm.es/Projects/S-CUBE/papers/oasis05:WSQM-2.0.pdf.

KUYORO, S. O.; AWODELE, O.; AKINDE R., O.; Okolie, S. O. (2012) Quality of service (Qos) issues in web services. *IJCSNS International Journal of Computer Science and Network Security*, vol. 12, no. 1.

LEITE, J. C. S. P. (1988) *Viewpoint Resolution in Requirements Elicitation*. PhD Thesis, Department of Computer Science, University of California, Irvine.

LUDWIG, H. (2003) Web services QoS: external SLAs and internal policies or: how do we deliver what we promise? In: *Proceedings of the Fourth International Conference on Web Information Systems Engineering Workshops*, p. 115–120.

MOELLER, S. (2010) Characteristics of services – a new approach uncovers their value. *Journal of Services Marketing*, vol. 24, no. 5, p. 359–368.

OASIS (2005) *Quality Model for Web Services*. OASIS, September 2005.

OASIS (2012) *Web Services Quality Factors Version 1.0*. OASIS Standard, 31 October 2012.

REEVES, C. A.; BEDNAR D. A. (1994) Defining quality: alternatives and implications. *Academy of Management Review*, vol. 19, no. 3, p. 419–445.

SETH, N.; DESHMUKH, S.G. (2005) Service quality models: a review. *International Journal of Quality & Reliability Management*, vol. 22, no. 9, p. 913–949.

SHEKHOVTSOV, V. A. (2011) On the evolution of quality conceptualization techniques. Kaschek, R.; Delcambre, L. (eds.), *The Evolution of Conceptual Modeling*. LNCS 6520, Springer-Verlag Berlin: Heidelberg, p. 117–136.

SOA (2012) *Reference Architecture Foundation for Service Oriented Architecture*. OASIS Committee Specification 01 (04 December 2012), Version 01.

TANG, L.; DONG, J.; PENG, T.; TSAI, W.-T. (2010) Modeling enterprise service-oriented architectural styles. *Service Oriented Computing and Applications*, vol. 4, no. 2, p. 81–107.

ULLAH, M.; FIEDLER, M.; WAC, K. (2012). On the ambiguity of quality of service and quality of experience requirements for eHealth services. In: *Proceedings of the International Symposium on Medical Information and Communication Technology (ISMICT)*, IEEE Computer Society, La Jolla, 1–4.

W3C (2001) *Web Services Description Language (WSDL) 1.1*. W3C Note 15 March 2001. http://www.w3.org/TR/wsdl.

W3C (2004) *Web Service Architecture*. W3C Working Group Note 11 February, 2004. 99 p.

TRIJŲ DOMINUOJANČIŲ PASLAUGŲ KOKYBĖS MODELIAVIMO BŪDŲ KRITINĖ ANALIZĖ IR VERTINIMAS

Audronė Lupeikienė, Jolanta Miliauskaitė, Albertas Čaplinskas

S a n t r a u k a

Paslaugų stiliaus architektūra tampa vyraujančia saityno paslaugomis grindžiamų sistemų kūrimo paradigma. Šiame straipsnyje pateikiama trijų dominuojančių paslaugų kokybės modeliavimo būdų: SQuaRE grindžiamo saityno paslaugų kokybės modelio, OASIS sudaryto saityno paslaugų kokybės modelio ir ontologija grindžiamo paslaugų kokybės modeliavimo būdo, pasiūlyto Beanchini su bendraautoriais, kritinė analizė. Pagrindinis darbo rezultatas yra straipsnyje parodomi analizuojamų būdų koncepciniai panašumai ir skirtumai, nagrinėjami trūkumai ir priežastys, neleidžiančios juos taikant adekvačiai modeliuoti paslaugų kokybės paslaugų stiliaus architektūros įmonių sistemose. Taip pat pateikiami argumentai, kad reikia sukurti naują, holistinį, paslaugų kokybės modeliavimo būdą.

Statistinių metodų taikymas daugiatikslių sprendimų patikimumui įvertinti

Rūta Simanavičienė

Vilniaus Gedimino technikos universiteto Fundamentinių mokslų fakulteto Matematinės statistikos katedros docentė
Docent of Mathematical statistics department of the Faculty of Fundamental sciences in Vilnius Gediminas Technical University, Assoc. Professor
Saulėtekio al. 11, LT-10223 Vilnius
El. paštas: ruta.simanaviciene@vgtu.lt

Daugelio socialinių, statybos investicinių, medicinos ir kitų sričių problemų sprendimui gali būti formuluojami daugiatiksliai sprendimo priėmimo uždaviniai. Dažnai tokie uždaviniai yra susijęs su neapibrėžtimi dėl įvairių priežasčių – tiek objektyvių, tiek subjektyvių. Tuomet kyla klausimas, ar gautas sprendimas bus patikimas. Vieningos metodikos, skirtos daugiatikslių sprendimų patikimumui įvertinti pagal rodiklių reikšmingumus ir rodiklių reikšmes, nagrinėtoje literatūroje neaprašyta. Minėtos priežastys skatina ieškoti metodų, kurie padėtų nustatyti, ar gautas sprendimas yra patikimas. Šio straipsnio tikslas – pritaikyti statistinius metodus daugiatikslių sprendimų patikimumui įvertinti. Tam siūloma taikyti imitacinį duomenų modeliavimą ir šiuos statistinius metodus: 1) rezultatų imčių neparametrinių hipotezių tikrinimą; 2) rezultatų imčių vidurkių pasikliautinųjų intervalų skaičiavimą; 3) rezultatų imčių vidurkių parametrinių hipotezių tikrinimą. Taikant minėtus metodus pasiūlytas sprendimo patikimumo įvertinimo algoritmas.

Įvadas

Daugiakriteriai sprendimų priėmimo metodai, naudojami optimaliam sprendimui rasti, skirstomi į dvi grupes: daugiaobjekčius ir daugiatikslius (Hwang, Yoon, 1981). Šiame straipsnyje nagrinėjami daugiatiksliai (angl. *Multi-Attribute Decision Making – MADM*) sprendimo priėmimo metodai, grindžiami kiekybiniais matavimais. Šių metodų pradiniai duomenys yra rodiklių reikšmingumo vektorius ir sprendimo priėmimo matrica, sudaryta iš rodiklių reikšmių. Pradiniai duomenys nustatomi remiantis statistiniais ar ekspertiniais duomenimis. Daugiatiksliu metodu gautas rezultatas yra nagrinėjamų alternatyvų išrikiavimas prioritetine eilute (Hwang, Yoon, 1981).

Dažnai daugiatikslių uždavinių sprendimas yra susijęs su pradinių duomenų neapibrėžtumu, kurį lemia tiek objektyvios, tiek subjektyvios priežastys. Tuomet kyla klausimas, ar gautas sprendimas bus patikimas. Minėtos priežastys skatina ieškoti būdų, kaip įvertinti daugiatikslio sprendimo patikimumą pradinių duomenų paklaidų atžvilgiu.

Nagrinėtoje literatūroje yra mokslinių darbų, kuriuose tiriamas daugiakriterio sprendimo patikimumas, susijęs su pradinių duomenų neapibrėžtumu. Mokslininkai šią problemą dažnai siūlo spręsti taikant neapibrėžtų aibių teoriją (Chen, 2012; Balli, Korukoglu, 2009), intervalines neapibrėžtas aibes (Liu, 2010), pilkuosius skaičius (Zavadskas ir kt., 2009). Daugiatikslio sprendimo rezultatas taikant minėtus sprendimo metodus pateikiamas intervaline išraiška, o alternatyvos racionalumo patikimumo įvertis neskaičiuojamas.

Sprendimų patikimumo klausimą svarstė P. Pyy. Autorius taikė Bradley-Terry modelį, skirtą apskaičiuoti tikimybei, jog *i*-oji alternatyva

yra pranašesnė už *j*-ąją, tačiau autorius nagrinėjo patikimumą tik tokių sprendimų, kurie susiję su žmogaus valdomomis sistemomis (Pyy, 2000).

Daugiatiksliuose sprendimo priėmimo metoduose naudojami ne tik objektyvūs, bet ir subjektyvūs duomenys. Pastarieji, gauti iš ekspertų, reikalingi rodiklių reikšmingumui nustatymui. Šių duomenų patikimumui užtikrinti pasiūlyta skaičiuoti konkordancijos koeficientą (Satty, 1980; Podvezko, 2005), kuris parodo, ar ekspertų nuomonės dėl rodiklių reikšmingumų yra pakankamai suderintos.

Patikimumo funkcijos apibrėžimą pateikė A. Birolini, tačiau jo apibrėžtas patikimumas yra taikomas tik elektroninių, gamybinių sistemų ir jų elementų patikimumui įvertinti (Birolini, 2010).

Remiantis nagrinėta literatūra galima teigti, jo vienos metodikos, kuria būtų galima įvertinti, ar daugiatikslis sprendimas yra patikimas rodiklių reikšmingumų ir rodiklių reikšmių atžvilgiu, nėra aprašyta.

J. M. Aughenbaugho ir J. W. Herrmanno teigimu, patikimumui įvertinti tinka įvairūs statistiniai metodai (Aughenbaugh, Herrmann, 2009). Tuo remiantis buvo suformuluotas straipsnio tikslas – pritaikyti statistinius metodus daugiatikslių sprendimų patikimumui įvertinti. Šiam tikslui pasiekti darbe siūloma naudoti daugiatikslio sprendimo alternatyvų racionalumo (naudingumo) įverčių imitacinį modeliavimą Monte Karlo metodu ir šiuos statistinius metodus: 1) neparametrinių hipotezių tikrinimą; 2) vidurkių pasikliautinųjų intervalų skaičiavimą; 3) vidurkių parametrinių hipotezių tikrinimą.

Daugiatikslio uždavinio formulavimas

Tarkim, turime daugiatikslį sprendimo priėmimo uždavinį, kurio sprendinys yra nagrinėjamų alternatyvų išdėstymas prioritetine eilute, remiantis alternatyvų naudingumo reikšmėmis. Daugiatikslį sprendimo priėmimo uždavinį sudaro šie etapai:

1. Sudaromas nagrinėjamų alternatyvų, iš kurių išrenkama racionali alternatyva, vektorius
$A = (A_1, A_2, \ldots, A_i, \ldots, A_m)$.

2. Suformuojamas rodiklių, pagal kuriuos vertinamos alternatyvos, vektorius
$X = (X_1, X_2, \ldots, X_j, \ldots, X_n)$.

3. Turint alternatyvų ir rodiklių aibes, formuojama sprendimo priėmimo matrica $\mathbf{X}_{[m\times n]}$, kurią sudaro *i*-osios alternatyvos kiekybiniai įverčiai pagal *j*-uosius rodiklius $x_{ij}, \left(i = \overline{1,m},\ j = \overline{1,n}\right)$. Sprendimo matrica užpildoma kiekybiniais įverčiais, kurie gaunami remiantis statistiniais arba ekspertiniais duomenimis:

$$\mathbf{X}_{[m\times n]} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix} \quad (1)$$

4. Įvedami rodiklių reikšmingumai $q_j, \left(j = \overline{1,n}\right)$, nustatyti ekspertų.

5. Alternatyvų naudingumo reikšmės randamos pasirinktu daugiatiksliu sprendimo priėmimo metodu. Racionalia laikoma ta alternatyva, kurios naudingumo reikšmė yra didžiausia:

$$A^* = \max_{i\in(1,2,\ldots,m)} \left(f(A_i)\right); \quad (2)$$

čia $f(A_i), i = \left(\overline{1,m}\right)$ *i*-osios alternatyvos naudingumo reikšmė.

6. Alternatyvos išrikiuojamos prioritetine eilute pagal jų naudingumo reikšmes.

Jeigu $f(A_i) > f(A_l)$,

tai $A_i \succ A_l, (A_i, A_l \in A)$. (3)

Imitacinio modeliavimo taikymas daugiatikslio sprendimo patikimumo analizei

Kaip yra žinoma, socialinių ir elgsenos mokslų metodologijoje galioja norma, jog mokslinę prasmę turi tik atsitiktinės imties pagrindu atlikti tyrimai**.** Remiantis literatūroje aprašytais tyrimais, kuriems taikomas imitacinis duomenų modeliavimas (Mun, 2006; Saltelli et al., 2000), šiame darbe sprendimo matricų generavimui pasirinkta taikyti būtent jį. Straipsnyje daugia-

tikslio sprendimo patikimumo analizei siūlomas algoritmas (1 pav.), kurį sudaro šie žingsniai:

1. Sudaroma pradinė sprendimų matrica:

$$X_{[m\times n]} = \{x_{ij}\}, \left(i = \overline{1,m},\ j = \overline{1,n}\right),$$

kurioje *i*-osios alternatyvos *j*-ojo rodiklio reikšmė yra kiekybinis dydis.

2. Nustatomi kiekvieno rodiklio vidutiniai kvadratiniai nuokrypiai $\sigma_j, \left(j = \overline{1,n}\right)$, gaunami remiantis statistiniais arba ekspertiniais duomenimis.

3. Pasirenkamas pasiskirstymo dėsnis sprendimo matricos elementams generuoti.

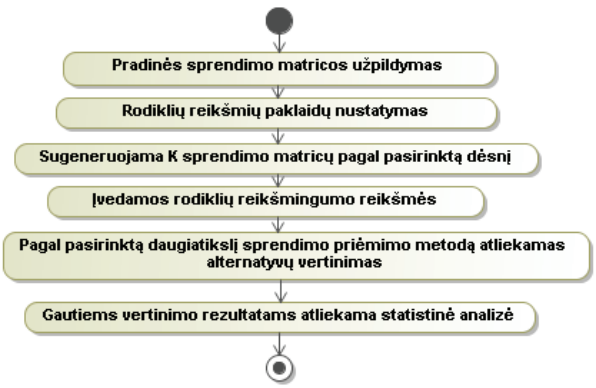
4. Atsižvelgiant į pradinę sprendimo matricą, pagal pasirinktą dėsnį generuojama *K* sprendimo matricų $X^k_{[m\times n]}, \left(k = \overline{1,K}\right)$, kur *m* – alternatyvų skaičius, *n* – rodiklių skaičius.

5. Fiksuojami rodiklių reikšmingumai $q_j, \left(j = \overline{1,n}\right)$, tenkinantys sąlygą: $\sum_{j=1}^{n} q_j = 1$.

6. Pasirenkamas daugiatikslis sprendimo priėmimo metodas.

7. Atitinkamai imant po vieną generuotą matricą ir rodiklių reikšmingumo vektorių, atliekamas alternatyvų racionalumo vertinimas pasirinktu metodu. Skaičiavimo rezultatai pateikiami vektoriais $A^k = \left(a_1^k, a_2^k, \ldots, a_m^k\right), \left(k = \overline{1,K}\right)$, kuriuose alternatyvos išrikiuojamos į prioritetinę eilutę (3) pagal alternatyvoms nustatytus rangus.

8. Galutinis sprendimas ir jo patikimumas nustatomas atlikus alternatyvų rangų ir racionalumo reikšmių statistinę analizę.



1 pav. ***Daugiatikslių sprendimų patikimumo analizės algoritmas***

Statistinių metodų taikymas sprendimo patikimumui įvertinti

Kaip jau buvo minėta, daugiatikslio sprendimo patikimumui nustatyti siūloma taikyti statistinius metodus, kurie aprašyti šiais žingsniais:

1. Pagal gautas alternatyvų racionalumo reikšmes alternatyvoms suteikiamos rangų reikšmės. Nustatoma dažniausiai pasitaikanti rango reikšmė kiekvienai alternatyvai ir apskaičiuojamas tos rango reikšmės pasikartojimų dažnis:

$$p(A_i) = \frac{n_i(l)}{K} \cdot 100\%; \qquad (4)$$

čia $p(A_i)$ vadinama alternatyvai A_i priskirto rango *l* patikimumo lygiu; *K* – sprendimo matricų skaičius; $n_i(l)$ – alternatyvai A_i dažniausiai pasitaikiusio rango reikšmės *l* dažnis (Simanaviciene, Ustinovichius, 2010).

2. Nustatoma, kuris pasiskirstymo dėsnis pakankamai gerai reprezentuoja nagrinėjamą imtį; tam formuluojama neparametrinė hipotezė apie stebimo kintamojo pasiskirstymą *F(x)* ir šiai prielaidai patikrinti skaičiuojama χ^2 suderinamumo kriterijaus reikšmė.

3. Kai nustatoma, kuris pasiskirstymo dėsnis geriausiai reprezentuoja alternatyvų racionalumo reikšmes, skaičiuojami alternatyvų racionalumo reikšmių vidurkių pasikliautinieji intervalai. Jeigu alternatyvų A_i *ir* A_{i+1} vidurkių $\boldsymbol{\mu}_i$ ir $\boldsymbol{\mu}_{i+1}$ pasikliautinieji intervalai (PI) susikerta, t. y.

$$\boldsymbol{P}\ (\mu_i) \cap \boldsymbol{P}\ (\mu_{i+1}) \neq \varnothing, \quad \left(i = \overline{1,m}\right), \qquad (5)$$

daroma išvada, jog teiginys, kad viena iš alternatyvų A_i *ar* A_{i+1} yra racionalesnė už kitą, yra nepatikimas. Priešingu atveju teiginys apie alternatyvų prioritetiškumą yra neabejotinas.

4. Alternatyvų vidurkių pasikliautinųjų intervalų susikirtimo atveju tikrinama hipotezė apie dviejų nepriklausomų imčių vidurkių lygybę. Jeigu nulinė hipotezė priimama, vadinasi, skirtumas tarp alternatyvų racionalumo reikšmių vidurkių yra statistiškai nereikšmingas ir daroma išvada, jog teiginys, kad viena iš lyginamų alternatyvų A_i *ar* A_{i+1} yra racionalesnė

už kitą, yra nepatikimas. Priešingu atveju alternatyvų racionalumo vidurkių reikšmės laikomos statistiškai reikšmingai skirtingomis ir tuomet daroma išvada, jog gautas sprendimas dėl racionaliausios alternatyvos yra neabejotinas.

Daugiatiksliai sprendimo priėmimo metodai SAW ir TOPSIS

Siūlomas daugiatikslio sprendimo patikimumo vertinimo algoritmas buvo pritaikytas daugiatiksliams sprendimo priėmimo metodams SAW ir TOPSIS.

SAW metodo pagrindas yra rodiklių reikšmių ir reikšmingumų sandaugų suma (Hwang, Yoon, 1981). Nustatant varianto racionalumą, atitinkami normalizuotos sprendimo matricos nariai dauginami iš rodiklių reikšmingumų ir gautos sandaugos sumuojamos. Racionalaus varianto sandaugų suma bus maksimali:

$$A = \left\{ A_i \,\middle|\, \max_i \sum_{j=1}^{n} q_j \bar{x}_{ij} \right\} \qquad (6)$$

čia q_j – rodiklių reikšmingumai; $\bar{x}_{ij}$ – normalizuotos sprendimo matricos elementai $i = \left(\overline{1,m}\right),\ j = \left(\overline{1,n}\right)$. Taikant SAW metodą, sprendimo matricos elementai normalizuojami pagal tiesinio normalizavimo formules (Hwang, Yoon, 1981).

TOPSIS metodo pagrindas yra atstumo iki „idealiai geriausios“ alternatyvos (angl. *ideal solution*) (Hwang, Yoon, 1981) Euklido erdvėje skaičiavimas. Atstumas tarp lyginamosios *i*-osios ir „idealiai geriausios“ A^+ alternatyvų nustatomas skaičiuojant atstumą *n*-matėje Euklido erdvėje pagal formulę:

$$L_i^+ = \sqrt{\sum_{j=1}^{n} \left(q_j \bar{x}_{ij} - a_j^+\right)^2},\ (i = \overline{1,m}). \qquad (7)$$

Atstumas tarp *i*-osios alternatyvos ir „neigiamai idealios“ alternatyvos A^- (angl. *negative ideal solution*) (Hwang, Yoon, 1981) skaičiuojamas pagal formulę:

$$L_i^- = \sqrt{\sum_{j=1}^{n} \left(q_j \bar{x}_{ij} - a_j^-\right)^2},\ (i = \overline{1,m}). \qquad (8)$$

Taikant TOPSIS metodą, sprendimų matrica $X_{[m\times n]} = \{x_{ij}\},\ \left(i = \overline{1,m},\ j = \overline{1,n}\right)$ normalizuojama vektorinės normalizacijos metodu. Galutiniu TOPSIS metodo žingsniu nustatomas kiekvienos *i*-osios alternatyvos santykinis atstumas iki „idealiai geriausios“ alternatyvos:

$$K_i = \frac{L_i^-}{L_i^+ + L_i^-},\ i = \overline{1,m},\ \text{kai } K_i \in [0,1]. \qquad (9)$$

Kuo K_i reikšmė artimesnė vienetui, tuo *i*-oji alternatyva artimesnė A^+, t. y. racionalus variantas bus tas, kurio K_i reikšmė yra didžiausia (Hwang, Yoon, 1981).

Pavyzdys

Suformuluojamas daugiatikslis sprendimo priėmimo uždavinys, nagrinėjamos trys alternatyvos, kurios vertinamos pagal keturis rodiklius.

1 lentelė. ***Sprendimo lentelė***

	R_1	R_2	R_3	R_4
A_1	50	0,214	571	193
A_2	78	0,213	665	299
A_3	50	0,222	690	191
Max/min	Max	Min	Min	Min

Sprendimo lentelėje (1 lentelė) pateiktos trijų išorinės pastato sienos konstrukcijų alternatyvų (A_1, A_2, A_3) rodiklių (R_1, R_2, R_3, R_4) reikšmės. Čia rodikliai atitinkamai: R_1 – sienų atsparumas šalčiui (ilgaamžiškumas metais); R_2 – sienos šilumos perdavimo koeficientas (W/m^{2}K); R_3 – išorės sienų svoris (m^{2}kg); R_4 – medžiagų kaina įrengti 1 m^2 sienos (Lt/ m^2).

Remiantis statistiniais duomenimis, šių rodiklių reikšmės pakankamai gerai reprezentuojamos normaliuoju skirstiniu. Laikantis šios prielaidos, rodiklių reikšmės generuojamos pagal normalųjį dėsnį, naudojant kiekvieno rodiklio reikšmę kaip vidurkį, o standartiniai nuokrypiai σ_j (j = 1, 2, 3, 4) gaunami skaičiuojant kiekvieno rodiklio reikšmių variacijos koeficientus. Nagrinėjamu atveju: $\sigma_1 = 27\,\%$; $\sigma_2 = 2\,\%$; $\sigma_3 = 10\,\%$; $\sigma = 27\,\%$.

Sugeneruojama po 100 atsitiktinių imčių, kiekvienam sprendimo matricos elementui. Iš gautų imčių sudaroma atitinkamai 100 naujų sprendimo matricų. Tariame, kad rodiklių reikšmingumai yra vienodo dydžio $q_j = 0{,}25$ ($j = 1, 2, 3, 4$).

Šie pradiniai duomenys nuosekliai pateikiami SAW metodui ir kiekvienos sprendimo matricos atžvilgiu nustatoma, kuri alternatyva yra pirmoje prioritetinės eilutės vietoje. Gauti rezultatai rodo, jog racionali alternatyva yra A_1, tačiau pagal statistinės analizės pirmą žingsnį šio teiginio patikimumas yra tik 46 % (pagal (4) formulę) (2 lentelė).

2 lentelė. ***Dažniausiai pasitaikančių rangų reikšmės ir patikimumo lygiai, gauti taikant SAW metodą***

Alternatyva	*A1*	*A2*	*A3*
Dažniausiai pasitaikanti rango reikšmė	1	2	3
Patikimumo lygis	46 %	42 %	52 %

Norint išsamiau nustatyti sprendimo patikimumą, atliekama statistinė racionalumo reikšmių analizė:

- Tikrinama neparametrinė hipotezė apie alternatyvų racionalumo reikšmių imties empirinio skirstinio suderinamumą su normaliuoju skirstiniu. Gauta, jog empirinis skirstinys yra suderinamas su teoriniu normaliuoju skirstiniu. Analogiškos išvados galioja ir alternatyvų A_2, bei A_3 rezultatams.
- Skaičiuojami alternatyvų racionalumo reikšmių vidurkių pasikliautinieji intervalai Apskaičiavus šiuos intervalus gauta, jog alternatyvų A_1 ir A_2 racionalumo reikšmių vidurkių pasikliautinieji intervalai nesusikerta. Tai leidžia daryti išvadą, jog alternatyvų rangavimas yra patikimas. Kitaip yra su alternatyvų A_2 ir A_3 racionalumo reikšmių vidurkių pasikliautinaisiais intervalais, kurie susikerta. Tai rodo, jog yra alternatyvų racionalumo reikšmių, kurios patenka ir į antro rango intervalą, ir į trečio rango intervalą. Tokia situacija sudaro rezultatų nepatikimumo prielaidą. Skaičiavimo išvadas galima matyti stačiakampių diagramoje (2 pav., a).
- Formuluojamos hipotezės apie dviejų nepriklausomų atsitiktinių dydžių vidurkių μ_1 ir μ_2 lygybę bei vidurkių μ_2 ir μ_3 lygybę, kai dispersijos nelygios ir nežinomos. Palyginus skaičiuotą statistikos t reikšmę su Stjudento skirstinio kritine reikšme $t_{\frac{\alpha}{2},(k)} = t_{0,025(200)} = 1{,}972$, daromos išvados: alternatyvų A_1 ir A_2 racionalumo reikšmių vidurkiai skiriasi statistiškai reikšmingai, o alternatyvų A_2 ir A_3 racionalumo reikšmių vidurkių skirtumas statistiškai nereikšmingas.

Su tais pačiais pradiniais duomenimis buvo atlikti skaičiavimai TOPSI metodu. Gauti rezultatai rodo, jog racionaliausia alternatyva yra A_1, tačiau šio teiginio patikimumas yra tik 45 % (3 lentelė).

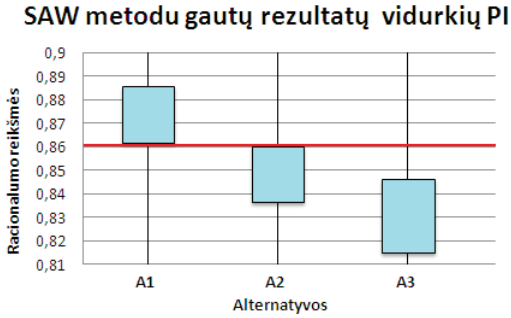
3 lentelė. ***Dažniausiai pasitaikančių rangų reikšmės ir patikimumo lygiai, gauti taikant TOPSIS metodą***

Alternatyva	*A1*	*A2*	*A3*
Dažniausiai pasitaikanti rango reikšmė	2	3	3
Patikimumo lygis	45 %	41 %	38 %

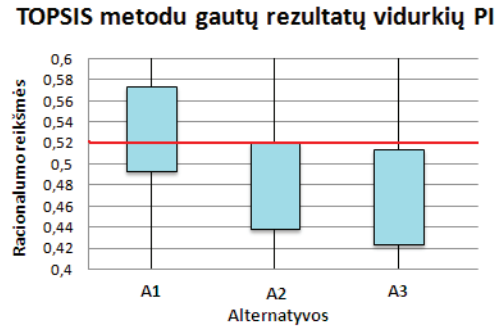
Patikrinus neparametrines hipotezes apie alternatyvų racionalumo reikšmių imties empirinio skirstinio suderinamumą su normaliuoju skirstiniu gauta, jog alternatyvų A_1, A_2 ir A_3 racionalumo reikšmių imties empirinis skirstinys yra suderinamas su teoriniu normaliuoju skirstiniu.

Apskaičiavus racionalumo reikšmių vidurkių pasikliautinuosius intervalus gauta, jog alternatyvų A_1 ir A_2 bei A_2 ir A_3 racionalumo reikšmių vidurkių pasikliautinieji intervalai susikerta, todėl daroma išvada, jog alternatyvų rangavimas yra nepatikimas. Gautos išvados vaizduojamos stačiakampių diagrama (2 pav., b).

Suformulavus ir patikrinus hipotezes apie dviejų nepriklausomų atsitiktinių dydžių vidurkių μ_1 ir μ_2 lygybę bei vidurkių μ_2 ir μ_3 lygybę, kai dispersijos nelygios ir nežinomos, daroma išvada: alternatyvų A_1 ir A_2 bei alternatyvų A_2 ir A_3 racionalumo reikšmių, gautų TOPSIS metodu, vidurkių skirtumai statistiškai nereikšmingi.



a)



b)

2 p a v. a ir 2 p a v. b. ***Racionalumo reikšmių vidurkių pasikliautinieji intervalai, gauti a) SAW ir b) TOPSIS metodais***

Išvados

Vienos metodikos, kuria būtų galima įvertinti daugiatikslio sprendimo patikimumą rodiklių reikšmingumų ir rodiklių reikšmių atžvilgiu, nagrinėtoje literatūroje neaprašyta.

Šiame darbe pasiūlytas algoritmas, skirtas daugiatikslių sprendimų patikimumui įvertinti, pagrįstas atsitiktinių skaičių generavimu ir statistiniais metodais. Atlikus eksperimentinius skaičiavimus SAW ir TOPSIS metodais, gauti rezultatai parodė:

1. Jeigu pradiniai duomenys yra generuojami pagal normalųjį dėsnį, tai sprendimo rezultatai bus pasiskirstę pagal normalųjį dėsnį nepriklausomai, ar sprendimų matricos normalizacija tiesinė, ar vektorinė.
2. Esant gretimų alternatyvų racionalumų vidurkių pasikliautinųjų intervalų susikirtimui, galima tvirtinti, jog teiginiai apie alternatyvų rangus yra nepatikimi. Tokią išvadą patvirtina ir parametrinės hipotezės apie gretimų alternatyvų racionalumo reikšmių vidurkių lygybę.
3. Tiek vidurkių pasikliautinieji intervalai, tiek parametrinės hipotezės apie vidurkių lygybę parodė, jog rodiklių reikšmių paklaidų atžvilgiu SAW metodu gauti rezultatai yra patikimesni nei TOPSIS metodu.

Apibendrinant galima teigti, jog šie trys statistiniai metodai: neparametrinės hipotezės formulavimas ir tikrinimas, pasikliautinųjų intervalų skaičiavimas ir jų susikirtimo tikrinimas bei hipotezės apie nepriklausomų imčių vidurkių lygybę tikrinimas, yra tinkami daugiatikslių sprendimų patikimumui įvertinti.

LITERATŪRA

AUGHENBAUGH, J. M.; HERRMANN, J. W. (2009) Reliability-Based Decision-Making: A Comparison of Statistical Approaches. *Journal of Statistical Theory and Practice*, vol. 3(1), p. 289–304.

BALLI, S.; KORUKOGLU, S. (2009). Operating system selecting using fuzzy AHP and TOPSIS methods. *Mathematical and Computational Applications* vol. 14(2), p. 119–130.

BIROLINI, A. (2010). *Reliability Engineering. Theory and Practice*. 6th ed. Berlin: Springer. 610 p.

CHEN, T. Y. (2012). Comparative analysis of SAW and TOPSIS based on interval-valued fuzzy sets: Discussions on score functions and weight constraints. *Expert Systems with Applications*, vol. 39(2), p. 1848–1861.

HWANG, C. L.; YOON K. (1981). *Multiple attribute decision making – methods and applications. A State of the Art Surwey.* Berlin–Heidelberg–New York: Springer Verlag. 250 p.

LIU, P. (2010). Multi-attribute decision-making method research based on interval vague set and

TOPSIS method. *Technological and Economic Development of Economy*, vol. 15(3), p. 453–463.

MUN, J. (2006). *Modeling Risk: Applying Monte Carlo Simulation, Real Options Analysis, Stochastic Forecasting, and Optimization*. ISBN 0471789003. 610 p.

PYY, P. (2000). An Approach for assessing human decision reliability. *Reliability Engineering and System Safety*, vol. 68, p. 17–28.

PODVEZKO, V. (2005). Ekspertų įverčių suderinamumas. *Technological and Economic Development of Economy* [Ūkio technologinis ir ekonominis vystymas], vol. 9(2), p. 101–107.

SAATY, T. L. (1980). *The Analytic Hierarchy Process*. New York: M.Graw-Hill.

SALTELLI, A.; CHAN, K; SCOTT, E. M. (2000). S*ensitivity analysis.* John Wiley & Sons, Ltd. 475 p. ISBN-10: 0-471-99892-3.

SIMANAVICIENE, R.; USTINOVICHIUS, L. (2010). Sensitivity Analysis for Quantitative Decision Making Methods: TOPSIS and SAW. In: *Proc. of the 16th International Conference on Information and Software Technologies (IT 2010),* April 21th–23th 2010, Kaunas, Lithuania. ISSN 2029-0020 (ISSN 2029-0063), p. 33–38.

ZAVADSKAS, E. K.; KAKLAUSKAS, A.; TURSKIS, Z.; TAMOSAITIENE, J. (2009). Multi-Attribute Decision-Making Model by Applying Grey Numbers. *Informatica*, vol. 20(2), p. 305–320.

APPLICATION OF STATISTICAL METHODS TO ASSESS THE RELIABILITY OF THE MULTIPLE ATTRIBUTES DECISIONS

Rūta Simanavičienė

S u m m a r y

Decisions making in many social, investment in construction, medical and other fields are transforming into the multiple attributes decision making problems, which are solved by the multiple attributes decision making methods. Often the multiple attributes decision making involved some uncertainty, conditioned by various reasons both objective and subjective. Consequently, the question arises "Is the final decision reliable?". The above-mentioned reasons motivate us to search the way of evaluating the reliability of the multiple attributes decision. The objective of this manuscript is to apply the statistical methods in the evaluation of reliability of the multiple attributes decision. The statistical methods used in this manuscript are: 1) nonparametric hypothesis on the sample of results; 2) parametric hypothesis on the averages of the result sample; 3) calculation of the confidence interval of averages of the result sample. The authors have proposed the algorithm for evaluating the reliability of multiple attribute decisions.

Atmena autoriams

Leidinyje spausdinami originalūs kitur neskelbti moksliniai straipsniai, kuriuose nagrinėjama informacijos ir komunikacijos mokslų srities problematika, analizuojama informacijos ir žinių visuomenės pobūdis, jos teisiniai, socialiniai, technologiniai aspektai, informacijos ir žinių visuomenės kultūros, švietimo, ekonomikos ir informacijos bei žinių vadybos klausimai, pateikiami organizacijos komunikacijos vadybos, kultūros, žiniasklaidos ir naujųjų medijų, globalios ir tarpkultūrinės komunikacijos, mokslotyros tyrimai. Skelbiamos naujų knygų, mokslinių tyrimų recenzijos, apžvalgos.

Straipsniai turi atitikti moksliniams tekstams keliamus reikalavimus. Juose turi būti aptartas nagrinėjamos problemos ištirtumas, nurodomas mokslinio tyrimo tikslas, objektas, metodai, pateikti tyrimų rezultatai, išvados. Straipsnį autorius pasirašo, nurodo savo gyvenamosios vietos adresą.

Straipsniai publikuojami lietuvių ir anglų kalbomis. Jie turi būti parengti laikantis tokios struktūros ir reikalavimų:

1) Straipsnio antraštė turi būti informatyvi. Jei reikia, rekomenduojamos paantraštės.
2) Autorius ar autoriai bei duomenys apie juos (autoriaus (-ių) vardas ir pavardė, mokslinis vardas ir laipsnis, mokslo įstaigos pavadinimas, adresas, telefono ir fakso numeris, elektroninio pašto adresas).
3) Trumpa anotacija lietuvių kalba (600–800 spaudos ženklų).
4) Pagrindinių žodžių sąrašas.
5) Straipsnio tekstas (rekomenduojama straipsnio apimtis 35 000–60 000 spaudos ženklų).
6) Literatūros sąrašas pateikiamas abėcėliškai pagal bibliografinių nuorodų standartus ISO 690:1987 (e) *Dokumentai. Bibliografinės nuorodos. Turinys, forma ir struktūra* ir LST ISO 690-2:1999 *Informacija ir dokumentai. Bibliografinės nuorodos*. D. 2: *Elektroniniai dokumentai ir jų dalys*.

Nuorodos į literatūrą tekste teikiamos lenktiniuose skliaustuose, nurodant autoriaus pavardę ir leidimo metus, pavyzdžiui, (Pavardenis, 2007).

Dokumentų aprašų pavyzdžiai:

Knygos aprašas

WEILL, Peter; ROSS, Jeanne (2004). *IT governance: how top performers manage IT decision rights for superior results.* Boston: Harvard Business Schools Press. 288 p. ISBN 1591392535.

Straipsnio aprašas

TINSLEY, Catherine; O'CONNOR, Kathleen; SULLIVAN, Brandon (2002). Tough guys finish last: the perils of a distributive reputation. *Organizational Behavior and Human Decision Processes,* No. 88, p. 621–645.

Elektroninio leidinio aprašas

Research methods tutorials [interaktyvus]. [S.l., s.a.] [žiūrėta 2007 m. gegužės 20 d.]. Prieiga per internetą: <http://www.socialresearchmethods.net/tutorial/tutorial.htm>.

Aprašai kirilika netransliteruojami ir netranskribuojami.

7) Išsami santrauka anglų kalba (straipsnio pavadinimas, autoriaus vardas ir pavardė ir straipsnio santrauka (600–800 spaudos ženklų).

Rankraštis pateikiamas dviem spausdintais egzemplioriais, viena skaitmenine .doc arba .rtf forma, dviejų eilučių intervalu, šriftas – *Times New Roman* 12 pt. Straipsnyje panaudotų nuotraukų ar iliustracijų failai pateikiami atskirai. Galima pateikti jų originalus arba kokybiškas jų kopijas, tinkamas reprodukuoti popieriuje.

Visi straipsniai vertinami dviejų mokslinių recenzentų. Rankraščiai negrąžinami.

Netinkamai parengti straipsniai negrąžinami, o jų autoriai apie tai informuojami raštu arba elektroniniu paštu.

Straipsniai publikuojami tik autoriui (autoriams) pasirašius licencinę sutartį.

Straipsniai įteikiami ir išleisti tomai įsigyjami šiuo adresu:

„Informacijos mokslų“ redaktorių kolegija
Vilniaus universiteto
Komunikacijos fakultetas,
Saulėtekio al. 9, I rūmai, 603 kab.
Tel. (+370 5) 236 61 19
Faks. (+370 5) 236 61 04
El. paštas: zenona.atkociuniene@kf.vu.lt
informacijos.mokslai@kf.vu.lt

Siunčiant paštu nurodyti tokį adresą:

„Informacijos mokslų“ redaktorių kolegija
Vilniaus universiteto
Komunikacijos fakultetas,
Universiteto g. 3,
LT-01513 Vilnius

Nuo 2007 m. mokslo darbai „Informacijos mokslai“ yra įtraukti į tarptautinę lISA duomenų bazę, kurioje referuojamų leidinių moksliniai straipsniai gali būti pripažįstami tinkami vertinant mokslo darbuotojų, kitų tyrėjų ir dėstytojų kvalifikaciją (Lietuvos mokslo tarybos 2005 m. gruodžio 19 d. nutarimas Nr. VI–24).

Guidelines for authors

"Informacijos mokslai" is a regional journal that covers topics on information and communication such as information and knowledge society, its legislative, technological aspects, cultural and economical problems of information and knowledge society, information and knowledge management questions, theoretical and practical researches on organisational communication management, investigations of culture and media, world and intercultural communication, works on scientometrics.

Only original scholar articles and surveys discussing all kinds of problems of communication and information, book reviews and academic chronicles are accepted.

The paper must contain all the necessary parts of a scientific paper – aims, objective, methods, results, conclusions, and the list of references. the paper should be signed by the author (or authors) with indication of address of residence.

The publishing languages are lithuanian and english.

Papers should be arranged as follows:

1) title,
2) author's name and surname, academic affiliation, office or private postal address, telephone and fax, e-mail address,
3) abstract (600–800 typographical units in Lithuanian or English, depending on the language of the submission),
4) keywords (up to five),
5) text,
6) reference list (alphabetically ordered sources in Latin alphabet should be placed before references in Cyrillic, notes must be placed as footnotes),
7) summary in English (for submissions in Lithuanian) or in Lithuanian (for submissions in English) of no less than 600–800 typographical units.

References in "Informacijos mokslai" should meet the requirements of the international standard ISO 690:1987 (E). *Documents. Bibliographic references. Contents, format, and structure* and the Lithuanian standard LST ISO 690-2:1999. Only mandatory data elements are recommended to use in the references. References in the text should indicate the author and the year of publishing, for example: (Surname, 2007).

Please use the following style for references:

A book:

WEILL, Peter; ROSS, Jeanne (2004). *IT governance: how top performers manage IT decision rights for superior results.* Boston: Harvard Business School Press. 288 p. ISBN 1591392535

Article in a journal:

TINSLEY, Catherine; O'CONNOR, Kathleen; SULLIVAN, Brandon (2002). Tough guys finish last: the perils of a distributive reputation. *Organizational Behavior and Human Decision Processes,* No. 88, p. 621–645.

Online Publication:

Research methods tutorials [interactive]. [S.l., s.a.] [accessed 20 May 2007]. Access through Internet: <http://www.socialresearchmethods.net/tutorial/tutorial.htm>.

Papers should be submitted in two hard copies (in double-spaced format), and an .rtf file on a diskette or via e-mail must be presented to the editing board. Tables, pictures and other graphic information should be presented on separate pages.

Submissions are peer-reviewed. Papers are reviewed by two reviewers. Rejected submissions will not be returned.

Papers before publishing are signed by the authors.

The corresponding author must sign the License to Publish form upon acceptance of the manuscript and return it to the editorial office. Failure to do so will result in a delay of the publication of the paper.

Address for submitting articles and for purchase of published ones:

Editing Board of "INFORMACIJOS MOKSLAI"
Faculty of Communication
Vilnius University
Saulėtekio Ave., Building 1, Room 603
Articles may be submitted by e-mail:
zenona.atkociuniene@kf.vu.lt
informacijos.mokslai@kf.vu.lt

Visiting address:

Editing Board of "INFORMACIJOS MOKSLAI"

Faculty of Communication
Vilnius University
Universiteto str. 3,
LT-01513 Vilnius, Lithuania

Since 2007, the scientific journal "Informacijos mokslai" is abstracted and indexed in the lISA international database and included in the list of Lithuanian serials that meet the requirements for scientific publications (Lithuanian Science Council, 19 December 2005, decision No. VI–39).

INFORMACIJOS MOKSLAI 2013 65
Mokslo darbai

Lietuvių kalbos redaktorė *Jolanta Storpirštienė*
Anglų kalbos redaktorė *Aurelija Juškaitė*
Viršelio dailininkas *Gediminas Markauskas*
Maketavo *Nijolė Bukantienė*

Už straipsnių turinį atsako autoriai

12 aut. l.; 16,5 sp. l.
Tiražas 200 egz.
Išleido Vilniaus universitetas
Universiteto g. 3, LT-01513 Vilnius
Spausdino UAB „Baltijos kopija“
Kareivių g. 13B, LT-09109, Vilnius

Kaina sutartinė