

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

Kęstutis NORMANTAS

VERSLO ŽINIŲ IŠGAVIMO IŠ EGZISTUOJANČIŲ PROGRAMŲ SISTEMŲ TYRIMAS

DAKTARO DISERTACIJOS SANTRAUKA

TECHNOLOGIJOS MOKSLAI,
INFORMATIKOS INŽINERIJA (07T)



LEIDYKLA
Vilnius TECHNIKA 2013

Disertacija rengta 2009–2013 metais Vilniaus Gedimino technikos universitete.
Mokslinis vadovas

prof. dr. Olegas VASILECAS (Vilniaus Gedimino technikos universitetas, technologijos mokslai, informatikos inžinerija – 07T).

Disertacija ginama Vilniaus Gedimino technikos universiteto Informatikos inžinerijos mokslo krypties taryboje:

Pirmininkas

doc. dr. Dalius MAŽEIKA (Vilniaus Gedimino technikos universitetas, technologijos mokslai, informatikos inžinerija – 07T).

Nariai:

prof. habil. dr. Antanas ČENYS (Vilniaus Gedimino technikos universitetas, technologijos mokslai, informatikos inžinerija – 07T),

dr. Robertas DAMASEVIČIUS (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija – 07T),

prof. dr. Lina NEMURAITĖ (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija – 07T),

prof. habil. dr. Antanas ŽILINSKAS (Vilniaus universitetas, fiziniai mokslai, informatika – 09P).

Oponentai:

doc. dr. Arnas KAČENIAUSKAS (Vilniaus Gedimino technikos universitetas, technologijos mokslai, informatikos inžinerija – 07T),

doc. dr. Minija TAMOŠIŪNAITĖ (Vytauto Didžiojo universitetas, fiziniai mokslai, informatika – 09P).

Disertacija bus ginama viešame Informatikos inžinerijos mokslo krypties tarybos posėdyje 2014 m. sausio 3 d. 13 val. Vilniaus Gedimino technikos universiteto senato posėdžių salėje.

Adresas: Saulėtekio al. 11, LT-10223 Vilnius, Lietuva

Tel.: (8 5) 274 4952, (8 5) 274 4956; faksas (85) 270 0112;

el. paštas doktor@vgtu.lt

Disertacijos santrauka išsiuntinėta 2013 m. gruodžio 2 d.

Disertaciją galima peržiūrėti Vilniaus Gedimino technikos universiteto (Saulėtekio al. 14, LT-10223 Vilnius, Lietuva) ir Vilniaus universiteto Matematikos ir informatikos instituto (Akademijos g. 4, LT-08663 Vilnius, Lietuva) bibliotekose.

VGTU leidyklos „Technika“ 2198-M mokslo literatūros knyga.

© Kęstutis Normantas, 2013

VILNIUS GEDIMINAS TECHNICAL UNIVERSITY

Kęstutis NORMANTAS

RESEARCH ON BUSINESS KNOWLEDGE EXTRACTION FROM EXISTING SOFTWARE SYSTEMS

SUMMARY OF DOCTORAL DISSERTATION

TECHNOLOGICAL SCIENCES,
INFORMATICS ENGINEERING (07T)



LEIDYKLA
Vilnius TECHNIKA 2013

Doctoral dissertation was prepared at Vilnius Gediminas Technical University in 2009–2013.

Scientific Supervisor

Prof Dr Olegas VASILECAS (Vilnius Gediminas Technical University, Technological Sciences, Informatics Engineering – 07T).

The dissertation is being defended at the Council of Scientific Field of Informatics Engineering at Vilnius Gediminas Technical University:

Chairman

Assoc Prof Dr Dalius MAŽEIKA (Vilnius Gediminas Technical University, Technological Sciences, Informatics Engineering – 07T).

Members:

Prof Dr Habil Antanas ČENYS (Vilnius Gediminas Technical University, Technological Sciences, Informatics Engineering – 07T),

Dr Robertas DAMAŠEVIČIUS (Kaunas University of Technology, Technological Sciences, Informatics Engineering – 07T),

Prof Dr Lina NEMURAITĖ (Kaunas University of Technology, Technological Sciences, Informatics Engineering – 07T),

Prof Dr Habil Antanas ŽILINSKAS (Vilnius University, Physical Sciences, Informatics – 09P).

Opponents:

Assoc Prof Dr Arnas KAČENIAUSKAS (Vilnius Gediminas Technical University, Technological Sciences, Informatics Engineering – 07T),

Assoc Prof Dr Minija TAMOŠIŪNAITĖ (Vytautas Magnus University, Physical Sciences, Informatics – 09P).

The dissertation will be defended at the public meeting of the Council of Scientific Field of Informatics Engineering in the Senate Hall of Vilnius Gediminas Technical University at 1 p. m. on 3 January 2014.

Address: Saulėtekio al. 11, LT-10223 Vilnius, Lithuania.

Tel.: +370 5 274 4952, +370 5 274 4956; fax +370 5 270 0112;

e-mail: doktor@vgtu.lt

The summary of the doctoral dissertation was distributed on 2 December 2013.

A copy of the doctoral dissertation is available for review at the Library of Vilnius Gediminas Technical University (Saulėtekio al. 14, LT-10223 Vilnius, Lithuania) and at the Library of Vilnius University Institute of Mathematics and Informatics (Akademijos str. 4, LT-08663 Vilnius, Lithuania).

© Kęstutis Normantas, 2013

Įvadas

Problemos formulavimas ir darbo aktualumas. Darbe nagrinėjama programų sistemų palaikymo ir vystymo problema. Nustatyta, jog sąnaudos šiose programų sistemų gyvavimo ciklo fazėse siekia iki 80% visų sąnaudų skiriamų programų sistemoms kurti. Pagrindinis šio reiškinio veiksnys yra nuolatinis poreikis pritaikyti sistemų funkcionalumą prie besikeičiančių verslo reikalavimų, o pritaikymo užduotys apima apie 40% visų palaikymo veiklų. Darbe apžvelgti tyrimai rodo, jog 40–60% palaikymo užduočių resursų skiriama įgyvendintai verslo logikai suvokti. Paprastai tai yra dėl to, kad atsakingi už sistemų palaikymą žmonės nėra jų projektuotai, todėl turi sugaišti nemažai laiko, kad išsiaiškintų ir perprastų sistemos veikimą bei verslo logikos įgyvendinimo principus. Be to, pakeitimai, atliekami palaikymo metu, yra retai dokumentuojami ar net nedokumentuojami visai, todėl žinios, įgytos įgyvendinant pakeitimus, lieka individualių programuotojų galvose.

Tuo tarpu kiti tyrimai atskleidė, jog paprastai tik trečdalis programų sistemos kodo įgyvendina verslo logiką, o kita dalis skiriama infrastruktūros funkcionalumui (pvz.: duomenų bazės jungties valdymui) užtikrinti ar platformai specifinėms funkcijoms (pvz.: pranešimų eilės valdymui) įgyvendinti. Iš to darytina išvada, jog automatizuojant verslo žinių išgavimą iš egzistuojančių programų sistemų, galima sumažinti sistemų palaikymo ir vystymo sąnaudas.

Atsižvelgdamas į šių problemų svarbą, tarptautinis standartizacijos konsorciumas OMG (angl. *Object Management Group*) inicijavo architektūra grindžiamo sistemų modernizavimo darbo grupę ADM (angl. *Architecture-Driven Modernization*), kurios tikslas yra suformuoti rinkinį standartų, įgalinančių kurti modernizavimo procesą automatizuojančias priemones. Žinių išgavimo meta-modelis (KDM – angl. *Knowledge Discovery Meta-model*) atlieka pagrindinį vaidmenį OMG ADM standartų šeimoje, kadangi jis suteikia galimybę žinias apie programų sistemas formaliai aprašyti ir analizuoti skirtingais aspektais įvairiuose abstrakcijos lygmenyse. Be to, KDM yra nepriklausomas nuo specifinės programavimo kalbos ar technologinės platformos, todėl jo pritaikymo spektras žinioms išgauti iš skirtingo tipo programų sistemų yra pakankamai platus. Nors mokslinėje literatūroje vyrauja susidomėjimas ADM standartais, tačiau tyrimų šioje srityje atliekama nepakankamai, o metodų, skirtų verslo žinioms išgauti yra siūloma vos keletas.

Tyrimų objektas. Šių tyrimų objektas yra procesas apimantis verslo žinių išgavimą iš egzistuojančių programų sistemų bei išgautų žinių vaizdavimą verslo žodynu, taisyklėmis bei verslo užduotimis.

Darbo tikslas. Patobulinti verslo žinių išgavimo iš egzistuojančių programų sistemų procesą pasiūlant metodą bei jį palaikančių priemonių karkasą.

Darbo uždaviniai. Siekiant įgyvendinti užsibrėžtą tikslą, suformuojami šie uždaviniai:

1. Atlikti metodų, skirtų verslo žinioms išgauti iš egzistuojančių programų sistemų, analizę ir nustatyti kokios verslo žinių rūšys yra išgaunamos šiais metodais, kokios technikos naudojamos išgaunant, kokie sistemų artefaktai nagrinėjami išgavimo metu bei kaip išgaunamos žinios yra atvaizduojamos.
2. Apibrėžti formalų pagrindą išgaunamoms žinioms vaizduoti ir analizuoti.
3. Pasiūlyti ir apibrėžti verslo žinių išgavimo iš egzistuojančių programų sistemų metodą bei sukurti jį palaikančių priemonių karkasą.
4. Atlikti eksperimentinį verslo žinių išgavimo iš egzistuojančios programų sistemos tyrimą bei įvertinti siūlomo metodo įgyvendinamumą ir efektyvumą.

Tyrimų metodai. Darbas atliekamas vadovaujantis projektavimo mokslo (angl. *Design Science*) metodikos direktyvomis, pasiūlytomis Hevner et al. Susijusių mokslinių darbų apžvalga atliekama vadovaujantis sisteminės literatūros apžvalgos (angl. *Systematic Literature Review*) metodikos rekomendacijomis, pasiūlytomis Kitchenham et al. Pasiūlytas metodas įvertinamas atliekant atvejo tyrimą (angl. *Case Study*).

Mokslinis naujumas. Darbe pasiūlytas verslo žinių išgavimo iš programų sistemų metodas – BKESS (akron. *Business Knowledge Extraction from Software Systems*) bei jį palaikančių priemonių karkasas. Siūlomas metodas yra paremtas žinių išgavimo meta-modeliu (KDM), kuris yra patvirtintas tarptautinės standartų organizacijos ISO (ISO/IEC 19506:2012(E)):

1. Metodas apima rinkinį transformacijų, skirtų programų sistemų tarpiniam atvaizdavimui išgauti naudojant KDM.
2. Metodas apima rinkinį KDM praplėtimų, skirtų programų analizei ir šablonų paieškai įgalinti.
3. Metodas apima programų analizės ir šablonų paieškos algoritmus KDM vaizduojamoms programų sistemoms.
4. Metodas apima praplėtimus tarpiniam verslo žinių vaizdavimui (kandidatų atrinkimui) naudojant KDM.
5. Metodas įgalina išgautų žinių tolimesnę transformaciją į aukštesnio abstrakcijos lygmens žinių vaizdavimo modelius.

Ankstyvieji tyrimų rezultatai buvo pritaikyti aukštųjų technologijų kūrimo programos 2007–2009 projekte „Veiklos taisyklių sprendimai informacinėms sistemoms kurti (VeTIS)“, finansuojamo Lietuvos mokslo ir studijų fondo.

Praktinė vertė

Siūlomas metodas yra skirtas palengvinti programų sistemų palaikymą ir vystymą bei sumažinti šių sistemos gyvavimo ciklo etapų kaštus. Be to, išgaautos žinios gali būti pakartotinai panaudotos peržiūrint ir patikslinant programų sistemų funkcinis reikalavimus. KDM kaip infrastruktūros tarpiniam žinių vaizdavimui bei analizei pasirinkimas užtikrina šio metodo pritaikomumą skirtingiems programų sistemų tipams, o palaikančių priemonių karkasas suprojektuotas siekiant jį integruoti su sistemų kūrimo aplinkomis bei įgalinti papildomų priemonių kūrimą.

Metodas buvo pritaikytas įmonės turinio valdymo sistemoje įgyvendintoms verslo žinioms išgauti. Tyrimo rezultatai patvirtino metodo įgyvendinamumą bei efektyvumą taip pabrėždami tokių metodų praktinę svarbą.

Ginamieji teiginiai

1. Darbe apibrėžtas KDM ribojimų rinkinys leidžia užtikrinti taisyklių programinio kodo ir duomenų struktūros modelių sudarymą.
2. Darbe pasiūlyti KDM plėtiniai įgalina programų sistemos analizės technikų taikymą verslo žinioms išgauti bei vaizduoti.
3. Pasiūlytas metodas palengvina verslo žinių išgavimą iš egzistuojančių programų sistemų ir užtikrina atsekamumą tarp išgautų žinių bei jų įgyvendinimo.

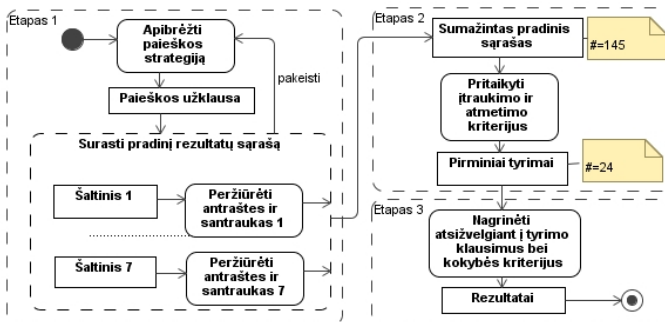
Darbo apimtis. Darbą sudaro įvadas, keturi skyriai, išvados, literatūros sąrašas, autoriaus publikacijų sąrašas bei 6 priedai. Bendra disertacijos apimtis 162 puslapiai, 52 iliustracijos ir 28 lentelės.

1. Sisteminė verslo žinių išgavimo iš egzistuojančių programų sistemų metodų analizė

Pirmame darbo skyriuje, remiantis Kitchenham et al. pateikiamomis rekomendacijomis, atliekama sisteminė susijusių darbų analizė, kuria siekiama atsakyti į šiuos klausimus:

- kokios verslo žinių rūšys yra išgaunamos susijusiuose tyrimuose,
- kokios technikos naudojamos ar siūlomos joms išgauti,
- iš kokių programų sistemos artefaktų išgaunamos žinios,
- kokios yra siūlomos išgautų žinių vaizdavimo formos?

Analizės metu suformuojama susijusių darbų paieškos strategija, kurioje apibrėžiami paieškos šaltiniai (5 skaitmeninės bibliotekos bei 2 paieškos sistemos), remiantis tyrimo klausimais sudaroma paieškos užklausa, suformuojami įtraukimo bei atmetimo kriterijai, nustatomi kokybės vertinimo kriterijai. Susijusių tyrimų atranka apima tris etapus (1 pav.), kurių rezultate skirtingais aspektais apžvelgiami 24 straipsniai, spausdinti recenzuojamoje mokslinėje literatūroje.



1 pav. Susijusių darbų paieškos procesas

Analizuojant literatūrą nustatoma, jog dažniausiai siūlomi metodai yra skirti verslo taisyklėms bei verslo procesams išgauti iš programų sistemų. Verslo žodyno, kuris sudaro pagrindą taisyklėms formuluoti, išgavimas akcentuojamas tik keliuose darbuose. Be to, nustatyta, jog retai atsižvelgiama į plačiai naudojamas verslo taisyklių klasifikacijas, o taisyklės suprantamos tik kaip sąlygos (pvz.: *if-then-else*) sakiniai ar su duomenų įvedimu ir išvedimu susiję kodo fragmentai, kuriems išgauti naudojamos kodo pjaustymo (angl. *program slicing*), šablonų paieškos bei transformacijų technikos. Svarbu pažymėti, jog didžioji dalis nagrinėtų tyrimų, siekiant išgauti tikslesnius rezultatus, taiko šių technikų kombinacijas. Tačiau, tik pavieniai darbai žinias mėgina išgauti ne tik iš programinio kodo, bet ir iš kitų programų sistemos artefaktų, tokių kaip naudotojo sąsajos formų, ataskaitų, platformos resursų failų, įvykių žurnalų, duomenų bazių ar dokumentacijos.

Nagrinėjant išgaunamų žinių vaizdavimo aspektą, nustatoma, jog žinios paprastai vaizduojamos keliais abstrakcijos lygmenimis: tarpiniu ir galutiniu. Tarpiniame lygmenyje formaliai apibrėžiama sistemos struktūra ir elgsena, naudojant kontrolės srauto grafą (angl. *control flow graph*), kvietimų grafą (angl. *call graph*), sistemos priklausomybių grafą (angl. *system dependence graph*) ar panašias notacijas. KDM išgaunamoms žinioms vaizduoti bei analizuoti panaudojamas tik keliuose nagrinėtuose metoduose. Galutiniame

vaizdavime, žinias siūloma pateikti kaip grafų pjūvius ar programinio kodo fragmentus; verslo procesų išgavimą tiriantys darbai išgautas žinias siūlo vaizduoti BPMN (akron. *Business Process Modelling Notation*) standartu.

Vertinant susijusių tyrimų kokybę, pastebima, jog daugelis darbų pateikia neformalius ar dalinai formalius žinių išgavimo algoritmų apibrėžimus. Tai įtakoja tiek siekis siūlomus metodus taikyti ne vienai technologinei bazei, tiek įvairūs darbų apimties ribojimai galiojantys mokslo leidiniuose. Be to, pastebima, jog daugelis siūlomų metodų apima ir palaikančias priemones, tačiau tik keliuose tyrimuose jos yra pritaikomos praktiniuose bandymuose nagrinėjant didelės apimties programų sistemas.

2. Žinių vaizdavimas ir analizė

Antrame darbo skyriuje formuojamas metodo teorinis pagrindas. Išanalizuojamas žinių išgavimo meta-modelis (KDM): jo galimybės atvaizduoti programų sistemą skirtingais aspektais bei skirtingais abstrakcijų lygmenimis, technikos naudotinos skirtingiems modeliams sudaryti bei jo plėtimo galimybės specifinėms žinioms apibrėžti ir vaizduoti. Nustatoma, jog KDM palaikomas stereotipais paremtas modelio elementų plėtimo mechanizmas yra pakankami ribotas, nesuteikiantis galimybių stereotipus susieti tarpusavyje, naudojant specializacijos ar asociacijos ryšius, dėl to ribojant specifinių dalykiniai sričiai žinių apibrėžimą.

Siekiant išsiaiškinti programų sistemos kodo ir duomenų struktūrų vaizdavimo KDM kodo ir duomenų modeliais principus, apibrėžiama pavyzdinė imperatyvi programavimo kalba, atitinkanti pagrindinius kalbų konstruktus, bei supaprastinta struktūrizuota užklausų kalba, atitinkanti SQL (angl. *Structured Query Language*) standartą. Identifikuojamos šių kalbų konstrukto atvaizdavimo KDM modelių elementais taisyklės. KDM specifikacijoje pateikiamų modelio elementų ribojimų aprašymų pagrindu bei panaudojant objektų ribojimų kalbą (OCL – angl. *Object Constraint Language*), formaliai apibrėžiamos kodo ir duomenų struktūrų modelių sudarymo taisyklingumo taisyklės (angl. *well-formedness rules*), padedančios užtikrinti išgaunamų modelių atitikimą KDM standartui.

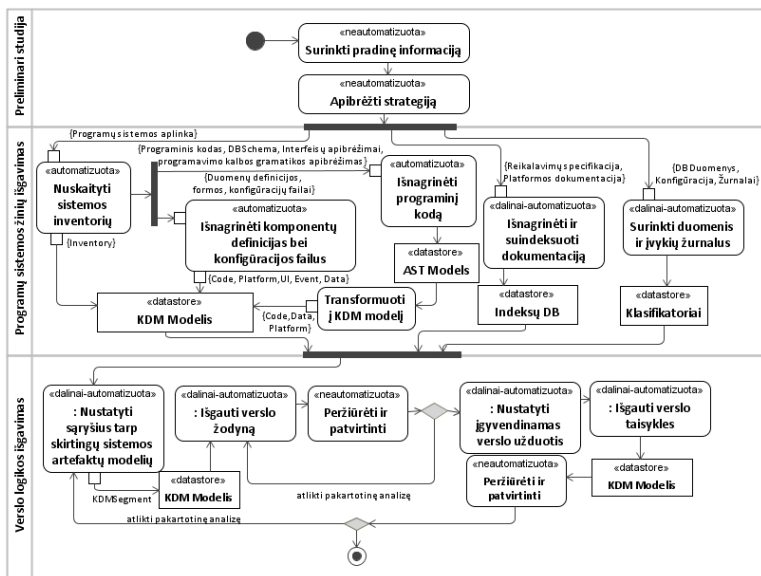
Remiantis klasikine duomenų srautų analizės literatūra, KDM kontekste apibrėžiamos analizės technikos, tokios kaip kintamųjų gyvavimo (angl. *live variable*), nenaudojamų kintamųjų (angl. *dead variable*), kintamųjų apibrėžimo (angl. *reaching definitions*), išraiškų gyvavimo (angl. *available expressions*) bei kitos susijusios technikos. Pasiūlomi šių technikų skaičiavimus įgyvendinantys KDM elementų plėtiniai, kurie apibrėžiami naudojant OCL atributų išvedimą (angl. *derived properties*). Apibrėžtos technikos vėliau panaudojamos žinių

išgavimo procese modelio elementams susieti bei išgaunamoms žinioms išgryninti.

3. Verslo žinių išgavimo iš programų sistemų metodas

Trečiame darbo skyriuje pasiūlomas verslo žinių išgavimo iš programų sistemų metodas, apimantis trys pagrindinius etapus: preliminarų įvertinimą, žinių išgavimą bei žinių išgryninimą ir vaizdavimą (2 pav.).

Preliminaraus įvertinimo metu yra surenkama pradinė informacija apie programų sistemą, nustatoma aukšto lygmens architektūra, identifikuojant pagrindinius komponentus bei juos įgyvendinančius artefaktus. Suformuojama žinių išgavimo strategija, kurioje apibrėžiama kiek ir kokių priemonių reikės artefaktams nagrinėti (angl. *parse*) ir analizuoti, kokie žmogiškieji resursai yra reikalingi, kokios numatomos užduotys bei šių užduočių vykdymo terminai.



2 pav. Verslo žinių išgavimo procesas

Antrame etape išgaunamas tarpinis žinių vaizdavimas: nuskaitomas programų sistemos inventoriųs (t. y. kaip organizuojamas programų sistemų artefaktų fizinis išdėstymas), nagrinėjamos ir verčiamos į KDM modelius resursų definicijos (formų, duomenų struktūrų, ataskaitų, konfigūracijos failų), išnagrinėjamas bei transformuojamas į modelius programinis kodas. Pagal suformuotą dokumentų struktūros modelį, išnagrinėjami esami dokumentai

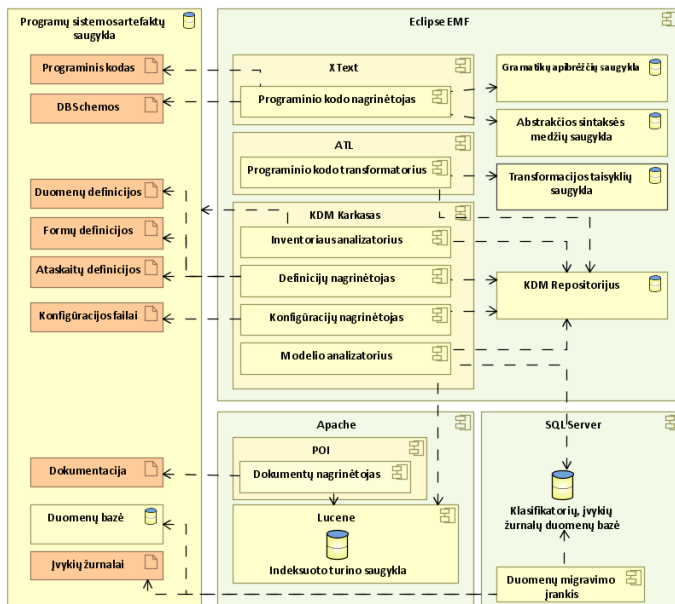
(reikalavimų specifikacija, platformos dokumentacija) bei suindeksuojami pilnatekstės paieškos saugykloje. Ištraukiamos ir perkeliamos į tarpinę duomenų bazę sistemoje naudojamų klasifikatorių reikšmės bei įvykių žurnalai.

Trečiame etape iš sukauptų žinių siekiama išgauti programų sistemoje įgyvendintą verslo logiką. Pirmiausia esami modeliai yra susiejami tarpusavyje, nustatant tiek horizontalius (tarp to paties KDM abstrakcijos lygmens modelių), tiek vertikalius (tarp skirtingų KDM abstrakcijos lygmenų modelių) ryšius. Tokiu būdu suformuojamas išsamus programų sistemos struktūrinių elementų duomenų ir kontrolės priklausomybių grafas, suteikiantis galimybę atlikti statinę modelių analizę. Susietų modelių pagrindu atliekamas verslo žodyno išgavimas, kurio metu siekiama nustatyti galimas verslo sąvokas (daiktavardinius konceptus, angl. *noun concepts*) bei sąryšius tarp jų (faktų tipus, angl. *fact types*). Užduočiai atlikti remiamasi verslo žodynų bei verslo taisyklių semantikos (SBVR – angl. *Semantics of Business Vocabulary and Business Rules*) specifikacija. Specifikacijoje apibrėžta faktų tipų klasifikavimo schema panaudojama kaip pagrindas faktų tipų išgavimo šablonams ir principams aprašyti. Jų išgavimas vykdomas analizuojant naudotojo sąsajos, duomenų struktūrų bei programinio kodo modelius. Išgavus, peržiūrėjus ir patikrinus verslo žodyną, nustatomos programų sistemos teikiamos verslo užduotys (angl. *use cases*) bei jų įgyvendinimo scenarijai (naudotojų sąsajos ar platformos įvykius įgyvendinančio kodo ir susijusių resursų modelio elementai, surinkti keliaujant priklausomybės grafo briauonais). Remiantis SBVR, išskiriamos kelios verslo taisyklių rūšys: struktūrinės ir operacinės. Struktūrinės taisyklės darbe papildomai išskiriamos į privalomumo, unikalumo ir reikšmių ribojimus bei išvedimo ir išskaičiavimo taisykles. Apibrėžiami taisyklių nustatymo šablonai bei jų išgavimo principai. Išgautoms žinioms vaizduoti pasiūlomas plėtinys, taikytinas KDM koncepciniam modeliui.

Siekiant patikrinti siūlomo metodo įgyvendinamumą ir pritaikomumą žinioms iš esamų sistemų išgauti ir analizuoti sukuriamas palaikančių priemonių karkasas (3 pav.). Palaikančios priemonės įgyvendintos *Eclipse* platformos papildiniais (angl. *plugin*), naudojant *Eclipse* modeliavimo karkasą (EMF – angl. *Eclipse Modelling Framework*) bei jo pagrindu pagamintus kitus papildinius.

Programinio kodo nagrinėjimo (angl. *parsers*) priemonėms generuoti, pasirinktas XText įrankis, kurio pagrindu pagamintos priemonės generuoja abstrakčios sintaksės medžius, atitinkančius EMF (ECore modelius). Taip suteikiama galimybė atlikti meta-modelių pagrindu aprašytas transformacijas. Transformacijoms atlikti pasirinkta plačiai naudojama ATL (akron. *Atlas Transformation Language*) priemonė. Dokumentų pilnatekstės paieškos indeksams saugoti bei paieškai atlikti siekiant susieti su išgautų modelių

elementais pasirinkta *Apache Lucene* saugykla, o esamos sistemos duomenims saugoti pasirinkta *SQL Server (Express Edition)* duomenų bazių valdymo sistema.



3 pav. Palaikančių priemonių karkaso architektūra

Suformuotas siūlomą metodą palaikančių priemonių karkasas užtikrina galimybę įtraukti kitas priemones, skirtas programų sistemų artefaktams nagrinėti (angl. *parse*), analizuoti, transformuoti ar išgaunamoms žinioms vizualizuoti panaudojant skirtingų tipų diagramas.

4. Siūlomo metodo įvertinimas

Ketvirtame darbo skyriuje aprašomas metodo taikymas verslo žinioms išgauti iš elektroninių dokumentų valdymo sistemos, naudojamos keliuose Lietuvos valstybinėse organizacijose. Tyrimo metu analizuojamas gautų dokumentų registravimo posistemis. Tiriamų artefaktų rinkinį sudaro paieškos ir registravimo formos, dokumentų bei klasifikatorių definicijos, konfigūracijos failai, duomenų bazės schemas pjūvis (lentelės, rodiniai, indeksai bei ribojimai), sistemos funkcionalumą įgyvendinantys programiniai moduliai, klasifikatorių lentelių duomenys bei esama dokumentacija (reikalavimų specifikacija ir platformos dokumentacija).

1 lentelė. Verslo žodyno elementų išgavimo efektyvumo vertinimas

<i>Elementas</i>	<i>tp</i>	<i>fp</i>	<i>fn</i>	<i>tn</i>	<i>P</i>	<i>R</i>	<i>A</i>	<i>F-matas</i>
Daiktavardžių konceptai	239	35	10	1110	0,87	0,96	0,97	0,91
Individualūs konceptai	1145	60	-	-	0,95	1,00	0,95	0,97
<i>Faktų tipai</i>								
Charakteristikos	29	4	1	-	0,88	0,97	0,85	0,92
Asociacijos	15	4	2	18	0,79	0,88	0,85	0,83
Savybės	171	-	2	151	1,00	0,99	0,99	0,99
Specializacijos	4	-	1	-	1,00	0,80	0,80	0,89
Dalyvavimo	1	2	-	6	0,33	1,00	0,78	0,50
Rūšiavimo	8	2	-	-	0,80	1,00	0,80	0,89
	Vidurkis				0,92	0,94	0,91	0,93
	Mediana				0,95	0,97	0,93	0,93
	Nuokrypis				0,09	0,09	0,09	0,07

2 lentelė. Verslo taisyklių išgavimo efektyvumo vertinimas

<i>Elementas</i>	<i>tp</i>	<i>fp</i>	<i>fn</i>	<i>tn</i>	<i>P</i>	<i>R</i>	<i>A</i>	<i>F-matas</i>
<i>Taisyklės</i>								
Elgsenos taisyklės	23	6	2	7	0,79	0,92	0,79	23
<i>Struktūrinės taisyklės</i>								
Privalomumo ribojimai	27	8	2	10	0,77	0,93	0,79	27
Unikalumo ribojimai	8	2	1	0	0,80	0,89	0,73	8
Reikšmių srities ribojimai	11	3	0	8	0,79	1,00	0,86	11
Išvedimo taisyklės	63	9	2	13	0,88	0,97	0,87	63
	Vidurkis				0,81	0,94	0,81	0,81
	Mediana				0,79	0,93	0,79	0,79
	Nuokrypis				0,04	0,04	0,06	0,04

Tyrimo metu gautiems rezultatams įvertinti, naudojamos informacijos išgavimo ir klasifikavimo metodams taikomos efektyvumo vertinimo metrikos: tikslumas (angl. *Precision*, **P**), atkuriamumas (angl. *recall*, **R**), atitikimas (angl. *accuracy*, **A**) bei **P** ir **R** harmoninį vidurkį nusakantis **F**-matas (angl. *F-measure*, **F**). Metrikos apskaičiuojamos remiantis šiais vertinimo kriterijais: teisingi rezultatai (*true positive*, tp), netikėti rezultatai (*false positive*, fp), trūkstami rezultatai (*false negative*, fn), teisingai atmesti rezultatai (*true negative*, tn).

Tyrimo dalyvavo sistemų analitikas bei verslo analitikas. Sistemų analitikas naudodamasis sukurtais priemonėmis taikė darbe siūlomą metodą sistemoje įgyvendintoms verslo žinioms išgauti, o verslo analitikas vertino išgautų rezultatų atitikimą tikrovei. Tyrimo metu buvo išgauti verslo žodynas,

taisyklių rinkinys ir devyni sistemos palaikomų verslo užduočių įgyvendinimo algoritmai, kurie apėmė gautų dokumentų paiešką (įskaitant peržiūrą, redagavimą, šalinimą bei pasirašymą), gautų dokumentų saugojimą (įskaitant nukreipimą atsakingiems bei pasirašymą) ir registravimą (įskaitant nukreipimą atsakingiems, pasirašymą bei įtraukimą į bylą). Verslo žodyno ir taisyklių rinkinio efektyvumo vertinimas pateikiami 1 ir 2 lentelėse.

Tyrimo metu gauti rezultatai parodo, jog metode buvo pakankamai tiksliai apibrėžtas verslo žodyno išgavimas, tačiau ne taip tiksliai apibrėžtas verslo taisyklių išgavimas. Apibendrinant, galima būtų teigti, jog metodas yra pakankamai efektyvus, nors ir tyrimų rezultatams galėjo turėti įtakos nedidelė tiriamųjų duomenų imtis, verslo analitiko subjektyvus vertinimas bei pasirinktos programų sistemos specifika, lėmusi žinių paieškos principų formavimą ir pritaikymą.

Reikėtų pažymėti, jog siūlomo metodo taikymas praktikoje atskleidė, jog žinių išgavimas iš egzistuojančių sistemų neišvengiamai lemia poreikį taikytis prie programų sistemų įgyvendinimo specifikos. Tyrime žinioms apibrėžti naudotas žinių išgavimo meta-modelis (KDM) įgalino suformuoti rinkinį skirtingus sistemos aspektus vaizduojančių modelių, tačiau, siekiant tiksliai suformuoti šiuos modelius, turėjo būti sukurtos specifinės sistemos artefaktų (definicijų, programinio kodo) nagrinėjimo ir transformavimo priemonės. Be to, tyrimų metu pasitvirtino teiginiai, jog didžioji dalis programinio kodo paprastai skiriama specifinėms platformos užduotims atlikti (pvz.: duomenų tipų konvertavimas ar resursų kontrolė), o KDM standarto panaudojimas suteikė galimybę išgaunamą verslo logiką atskirti į aukštesnio abstrakcijos lygmens modelį, išlaikant atsekamumą su modeliais, vaizduojančiais logikos įgyvendinimą kode.

Bendrosios išvados

1. Sisteminės literatūros analizės metu nustatyta, jog:
 - a) nagrinėtuose verslo žinių išgavimo iš programų sistemų metoduose skiriamas nepakankamas dėmesys verslo žodynui išgauti bei nesiremiam visuotinai pripažintomis verslo žinių klasifikacijos schemomis;
 - b) nagrinėtuose metoduose žinių išgavimas retai suprantamas kaip kompleksinė problema, kuri apima ne tik programinio kodo, bet ir kitų sistemos artefaktų bei sąryšių tarp jų analizę;
 - c) žinioms išgauti dažniausiai naudojamos kodo pjaustymo, šablonų paieškos ir transformacijų technikos ar jų kombinacijos;
 - d) nepakankamas dėmesys skiriamas žinių išgavimo meta-modeliui (angl. *Knowledge Discovery Meta-model*, KDM) kaip tarpinio

žinių vaizdavimo standartui, todėl yra trūkumas algoritmų modeliams sudaryti bei analizuoti.

2. Atliekant KDM standarto analizę nustatyta, jog KDM palaikomas plėtimo mechanizmas yra pakankamai ribotas. Stereotipų elementais paremtas plėtimo mechanizmas nesuteikia galimybės apibrėžti sąryšius tarp stereotipų, taip ribojant dalykiniai sričiai ar technologinei platformai specifinių žinių vaizdavimo galimybes bei apsunkinant modelių analizę.
3. KDM specifikacijoje meta-modelio elementų semantika nėra formaliai apibrėžiama ir tai neužtikrina taisyklingų modelių sudarymo. Todėl darbe formaliai apibrėžiamas dalinis, kodo ir duomenų modeliams taikomas taisyklingumo taisyklių (angl. *well-formedness rules*) rinkinys, kuris suteikia galimybę patikrinti išgaunamų modelių atitikimą KDM standartui.
4. Formuojant teorinį pagrindą nustatyta, jog KDM nėra pilnai pritaikytas programų analizės technikoms naudoti, todėl pasiūlyta praplėsti kontrolės srautui vaizduoti naudojamus elementus naujais atributais, kurių išvedimo taisyklės formaliai apibrėžiamos panaudojant objektų ribojimą kalbą (angl. *Object Constraint Language*).
5. Formuojant žinių išgavimo iš egzistuojančių programų sistemų metodą nustatyta, jog KDM nepakankamai pritaikytas išsamiam išgaunamų verslo žinių vaizdavimui. Todėl, naudojant KDM palaikomą plėtimo mechanizmą, apibrėžiamas plėtinys, skirtas išgaunamoms žinioms vaizduoti bei įgalinti jų transformavimą į abstraktesnes formas (pavyzdžiui SBVR ar sprendimų lenteles).
6. Atlikus metodo praktinio pritaikymo tyrimą verslo žinioms išgauti iš egzistuojančios dokumentų valdymo sistemos, buvo įrodytas metodo įgyvendinamumas ir įvertintas jo efektyvumas. Išgauto verslo žodyno tikslumo vidurkis siekia 92%, o atkuriamumo vidurkis siekia 94%, tuo tarpu abiejų rodiklių standartinis nuokrypis yra 9%. Išgauto taisyklių rinkinio tikslumo vidurkis siekia 81%, atkuriamumo – 94%, o šių rodiklių standartinis nuokrypis yra 4%. Nors pasirinkta tiriamų duomenų imtis yra pakankamai nedidelė, tačiau rezultatų efektyvumo rodikliai leidžia daryti išvadą, jog metodas yra pakankami efektyvus ir gali būti taikomas praktikoje.

Autoriaus mokslinių publikacijų disertacijos tema sąrašas

Straipsniai recenzuojamuose mokslo žurnaluose

Normantas, K.; Vasilecas, O. 2013. A systematic review of methods for business knowledge extraction from existing software Systems, *Baltic Journal of Modern Computing* 1(1–2): 29–51. ISSN 2255-8942.

Normantas, K.; Vasilecas, O. 2012. Business rules discovery from existing software Systems, *International journal of scientific & engineering research* 3(10): 1–7. ISSN 2229-5518. (EBSCO, DOAJ)

Normantas, K.; Vasilecas, O. 2009. Modelling of the business rules using UML/OCL, *Innovative Technologies for Science, Business and Education* 1(6): 2.1–2.10. ISSN 2029-1035.

Straipsniai kituose leidiniuose

Normantas, K.; Vasilecas, O. 2012. Extracting business rules from existing enterprise software systems. *Information and software technologies: 18th International Conference*. Communications in computer and information science, Vol. 319. New York: Springer, 482–496. (ISI Proceedings)

Normantas, K.; Sosunovas, S.; Vasilecas, O. 2012. An overview of the knowledge discovery meta-model. *Proceedings of the 13th international conference on Computer Systems and Technologies: CompSysTech'13*. New York: ACM, 52–57.

Vasilecas, O.; Normantas, K. 2011. Deriving business rules from the models of existing information systems. *Proceedings of the 12th International Conference on Computer Systems and Technologies: CompSysTech'12*. New York: ACM, 95–100. (Scopus, ACM Digital Library)

Vasilecas, O.; Normantas, K. 2010. Decision table based approach for business rules modelling in UML/OCL. *Proceedings of the 11th International Conference on Computer Systems and Technologies: CompSysTech'10*. New York: ACM Press. ACM international conference proceedings series, Vol. 471, 77–82.

Normantas, K.; Vasilecas, O. 2010. Business rules approach to e-learning systems development. *Proceedings of the 6th International Conference on E-learning and the Knowledge Society: e-Learning'10*. No. 6, Riga: RTU, 113–118.

Normantas, K.; Vasilecas, O.; Sosunovas, S. 2009. Defining well-formedness constraints with OCL. *Information technologies'2009: proceedings of the 15th*

international conference on Information and Software Technologies. Kaunas: Technologija, 355–364. (ISI Proceedings)

Normantas, K.; Vasilecas, O.; Sosunovas, S. 2009. Augmenting UML with decision tables technique. *Proceedings of the International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing: CompSysTech'09*. The Academic Society of Computer Systems and Information Technology. Varna: ACMBUL&UAI, 21.1–21.6.

Normantas, K.; Vasilecas, O.; Sosunovas, S. 2009. Management of business rules using decision tables. *Information technologies'2009: 15th international conference on Information and Software Technologies*. Kaunas: Technologija, 144–150.

Normantas, K.; Vasilecas, O. 2009. On expressing business rules with combination of UML and OCL. *Information technologies'2009: 15th international conference on Information and Software Technologies*. Kaunas: Technologija, 135–143.

Normantas, K.; Vasilecas, O. 2009. OCL as a specification language for business rules. *Akademinių jaunimo siekiai: ekonomikos, vadybos ir technologijų išvalgos: 6-oji studentų mokslinė konferencija*. Klaipėda: Klaipėdos universiteto leidykla, 152–157.

Normantas, K. 2009. Decision tables for the specification of business rules in object-oriented analysis and design. *Respublikinė jaunųjų mokslininkų konferencija „Fundamentiniai tyrimai ir inovacijos mokslų sandūroje“*. Klaipėda: Klaipėdos universiteto leidykla, 57–65.

Trumpos žinios apie autorių

Kęstutis Normantas gimė 1983 liepos 9 d. Klaipėdoje. 2006 m. įgijo informatikos bakalauro diplomą Vilniaus Gedimino technikos universitete. Šiame universitete 2008 m. įgijo informatikos magistro laipsnį, 2009–2013 m. studijavo informatikos inžinerijos mokslo krypties doktorantūroje. Nuo 2012 m. dirba Vilniaus Gedimino technikos universiteto Fundamentinių mokslų fakulteto Informacinių sistemų mokslo laboratorijoje jaunesniojo mokslo darbuotoju.

Padėka

Norėčiau išreikšti didelę padėką savo darbo vadovui, Vilniaus Gedimino technikos universiteto Fundamentinių mokslų fakulteto Informacinių sistemų katedros profesoriui Olegui Vasilecui už vertingus patarimus ir konsultacijas, nuoseklų vadovavimą bei nuolatinį rūpestingumą disertacijos rengimo metu.

Taip pat norėčiau padėkoti kolegoms iš VGTU Informacinių sistemų laboratorijos, Informacinių sistemų katedros, KTU Informacijos sistemų katedros, VDU Sistemų analizės katedros bei VGTU Doktorantūros skyriaus darbuotojams už vertingas pastabas ir pasiūlymus rengiant disertaciją.

Nuoširdžiai dėkoju savo brangiai žmonai Daliai, tėvams ir artimiesiems, draugams bei kolegoms už visapusišką palaikymą ir supratingumą disertacijos rengimo metu.

RESEARCH ON BUSINESS KNOWLEDGE EXTRACTION FROM EXISTING SOFTWARE SYSTEMS

An overview of the problem. The dissertation addresses the problem of software maintenance and evolution. It identifies that spending within these software lifecycle phases may account for up to 80% of software's total lifecycle cost, whereas the inability to adopt software quickly and reliably to meet ever-changing business requirements may lead to business opportunities being lost. The main reason of this phenomenon is the fact that the most of maintenance effort is devoted to understanding the software to be modified. On the other hand, related studies show that less than one-third of software source code contains business logic implemented within it, while the remaining part is intended for platform or infrastructure relevant activities. It follows that if the most of changes in software are made due to the need to adopt its functionality to changed business requirements, then facilitating software comprehension with automated business knowledge extraction methods may significantly reduce the cost of software maintenance and evolution.

Importance of the thesis. The Architecture Driven Modernization (ADM), a recent initiative of Object Management Group, aims at developing a number of standards to facilitate modernization activities by enabling software analysis and comprehension, and by allowing software transformations. The Knowledge Discovery Metamodel (KDM) (adopted as ISO/IEC 19506:2012(E)) plays a key role in this set of standards, as it allows defining the knowledge about existing software system from different perspectives at various levels of abstraction. Although the interest in ADM is increasing in scientific literature,

there is a lack of methods that rely on these standards and propose different approaches for knowledge extraction, representation, and analysis, thus aggravating invention of different tools that would support software modernization process.

The object of research. The object of this research is the process of business knowledge extraction from existing software systems and representation of extracted knowledge in form of business vocabulary, rules, and business use cases.

The goal of the thesis. This thesis aims at improving business knowledge extraction from existing software systems process by proposing a method and supporting tool framework.

The tasks of the thesis. In order to achieve the desired goal, the following tasks were established:

1. To analyse methods for business knowledge extraction from existing software systems for the purpose of identifying what kinds of business knowledge are being extracted, what techniques are being used as a basis for knowledge extraction, what types of software artefacts are being used as input sources, and how extracted knowledge is being represented.
2. To define the formal background for the business knowledge extraction method.
3. To propose and define a method for business knowledge extraction from existing software systems and develop a supporting tool framework.
4. To evaluate the feasibility and efficiency of the method by conducting a case study on the existing software system.

Research methodology. This thesis follows the Design Science methodology guidelines proposed by Hevner et al. The overview of related methods follows the Systematic Literature Review guidelines presented by Kitchenham et al. The evaluation of the method is performed by conducting the case study.

Scientific novelty. The main contribution of this thesis is a novel method for Business Knowledge Extraction from existing Software Systems (BKES). The method is based on the Knowledge Discovery Meta-model (KDM), recently adopted as ISO standard (ISO/IEC 19506:2012(E)):

1. The method involves a set of transformations to obtain intermediate representation of software system within the KDM.
2. The method introduces a set of extensions for the Knowledge Discovery Meta-model to allow application of program analysis and

pattern matching techniques in order to support business knowledge extraction.

3. The method provides a number of algorithms for the implementation of program analysis and pattern matching techniques within the KDM.
4. The method introduces a set of extensions to supplement definition of business vocabulary, rules and use cases within the KDM.
5. The method enables transformation from the intermediate representation within the KDM to representation forms relevant for business analysts and stakeholders.

The early results of this research have been applied in the high technology development program for 2007–2009 project “Business Rules Solutions for Information Systems Development (VeTIS)”, supported by Lithuanian State Science and Studies Foundation.

Practical significance of achieved results. The method is intended to facilitate the maintenance and evolution of existing software systems and reduce the costs of these software lifecycle phases. Moreover, systematically obtained business knowledge allows revising business requirements for the existing system.

Selection of the KDM as an infrastructure for intermediate representation and analysis of extracted knowledge ensures scalability of the method. The method supporting tool framework is designed to be integrated with the software development environment and to allow introduction of supplementary tools.

The results of case study on application of the method for knowledge extraction from enterprise content management system show the feasibility and efficiency of the proposed method and emphasize the practical significance of the results.

The defended statements

1. A set of KDM constraints formally defined within this work ensures producing well-formed Code and Data models of existing software systems.
2. A set of KDM extensions provided in this work enables application of program analysis techniques for extracting and representing business logic implemented within software systems.
3. The proposed method facilitates business knowledge extraction from existing software systems and ensures traceability between extracted knowledge and its implementation.

Approval of the results

The results of this thesis were published in 3 scientific journals and 11 peer-reviewed conference proceedings and collections. The results were presented in 13 international and Lithuanian scientific conferences:

- International Conference of Information and Software Technologies ICIST 18, Kaunas, Lithuania, 2012;
- International Conference on Computer Systems and Technologies CompSysTech'12, Ruse, Bulgaria, 2012;
- International Conference on Computer Systems and Technologies CompSysTech'11, Vienna, Austria, 2011;
- International Conference on Computer Systems and Technologies CompSysTech'10, Sofia, Bulgaria, 2010;
- 6th International Conference “E-learning and the Knowledge Society”, Riga, Latvia, 2010;
- Informatics: 13th Lithuanian junior scientist conference “Science – Future of Lithuania”, Vilnius, 2010;
- Information technology, data analysis and modelling: 12th republic junior scientist conference “Fundamental research and innovations in juncture of sciences”, Klaipeda, Lithuania, 2009;
- International Conference on Computer Systems and Technologies CompSysTech'09, Ruse, Bulgaria, 2009;
- Informatics: 12th Lithuanian junior scientist conference “Science – Future of Lithuania”, Vilnius, Lithuania, 2009;
- 15th international conference on Information and Software Technologies (Information Technology), Kaunas, Lithuania, 2009 (3 presentations);
- Scientific-practical conference “Academic youth aims: economical, management and technological insights”, Klaipeda, Lithuania, 2009.

Scope of the work. The dissertation consists of four chapters.

Chapter 1 presents a systematic literature review of related studies in order to summarize the state-of-the art in this research field, identify any gaps in the current research and explore possible directions for the further research. In particular, it aims at answering what kinds of business knowledge are being extracted in this research field, what techniques are used as a basis for knowledge extraction, what types of software artefacts are being considered as input sources, and how extracted knowledge is being represented.

Chapter 2 formulates theoretical background for the business knowledge extraction method by introducing selected standard for the intermediate knowledge representation, defining well-formedness rules for this

representation, and by revising and applying static program analysis techniques to this representation.

Chapter 3 describes the proposed method for automated business knowledge extraction from existing software systems. It explains the steps of the method, describes the principles and rules for knowledge extraction and representation in the form of business vocabulary, rules and use cases. This chapter ends with the introduction to the architecture of the supporting tool framework.

Chapter 4 presents the case study on applying the method for knowledge extraction from existing enterprise content management system. In order to establish method effectiveness, the results are being evaluated in respect with the precision, recall, and accuracy measures. The evaluation shows that the proposed method is feasible and efficient enough to be further improved and applied in practice.

The main observations are summarised and concluded within the General Conclusions chapter.

General conclusions

1. In the first chapter we presented the systematic literature review of related methods for business knowledge extraction from existing software systems. Selected studies were published in peer-reviewed scientific papers in journals, conference proceeding and collections. From this review we can conclude that:
 - a) there is insufficient attention paid to extraction of business vocabulary and no concern is made for widely used business rule classification schemes;
 - b) the proposed methods rarely concern knowledge extraction as a complex problem which involves more than program code analysis, thus aggravating their application in knowledge extraction from software systems developed using multiple technologies;
 - c) the most common techniques for knowledge extraction are program slicing, patterns matching and transformations, although certain methods use the combination of them;
 - d) a minority of proposed methods were evaluated in industrial case studies on large enterprise software systems, considering various software artefacts and other available knowledge sources;
 - e) the Knowledge Discovery Meta-model (KDM), as an intermediate representation of knowledge about existing software systems, is not studied enough and there is a lack of algorithms for creating representations within the KDM and performing analysis on the models.

2. It has been established that the KDM supported stereotype-based extension mechanism is limited in producing rigorous and precise representations and in performing analysis on its models, as it does not allow to define various kinds of relationships between stereotypes.
3. It has been identified that the KDM does not provide formally defined well-formedness constraints on meta-model elements; therefore, discovered models cannot be validated for conformance to the KDM. For this reason, a partial set of well-formedness rules for Code and Data models has been provided within this thesis.
4. It has been observed that the KDM does not support meta-model elements for representing and evaluating information required for dataflow analysis. Therefore a number of extensions in form of derived properties of KDM elements were proposed and their derivation has been formally defined.
5. It has also been established that the KDM is insufficiently adjusted to represent extracted business knowledge. For this reason, an extension family of stereotypes corresponding to fundamental SBVR concepts was proposed. It is supposed that this extension would allow further transformation of extracted knowledge to more abstract formats.
6. The results of case study prove the feasibility and show the efficiency of the proposed method. The extracted vocabulary was of an average of 92% of precise and of an average of 94% of recall. The extracted set of candidates to business rules was of an average of 85% and of an average of 89% of recall. Both forms of knowledge representation preserve traceability links to their implementation allowing system analysts or maintainers more quickly identify software parts of interest.

Kęstutis NORMANTAS

VERSLO ŽINIŲ IŠGAVIMO IŠ EGZISTUOJANČIŲ PROGRAMŲ
SISTEMŲ TYRIMAS

Daktaro disertacijos santrauka
Technologijos mokslai, informatikos inžinerija (07T)

Kęstutis Normantas
RESEARCH ON BUSINESS KNOWLEDGE EXTRACTION
FROM EXISTING SOFTWARE SYSTEMS

Summary of Doctoral Dissertation
Technological Sciences, Informatics Engineering (07T)

2013 12 02. 1,5 sp.l. Tiražas 70 egz.
Vilniaus Gedimino technikos universiteto
Leidykla "Technika",
Saulėtekio al. 11, LT-10223 Vilnius,
<http://leidykla.vgtu.lt>
Spausdino UAB "Ciklonas",
J.Jasinskio g. 15, 01111 Vilnius