

MODELIŲ TRANSFORMACIJOMIS PAREMTAS INFORMACINIŲ SISTEMŲ MODERNIZAVIMAS

Olegas Vasilecas¹, Simas Žikaras², Aidis Šmaižys²

¹Informacinių sistemų katedra, Vilniaus Gedimino technikos universitetas,
Saulėtekio al. 11, LT-10223, Vilnius, olegas@fm.vgtu.lt

²Informatikos katedra, Klaipėdos universitetas, Herkaus Manto g. 84, LT-92294, Klaipėda
simaszikaras@gmail.com, aidas@oil.lt

Anotacija. Sparčiai keičiantis verslo aplinkai, informacinės sistemos vis greičiau sensta ir nebeatitinka verslo poreikių. Liktinių informacinių sistemų pritaikymui prie besikeičiančios verslo aplinkos vis dažniau nebeužtenka vien nedidelių sistemų pakeitimų, jie pasidaro sudėtingi ir sunkiai įgyvendinami. Siekiant įdiegti naują funkcionalumą sukuriant kuo mažesnę riziką verslui, ši problema sprendžiama modernizuojant informacines sistemas, taip palengvinant jose esančios verslo informacijos supratimą ir pakartotinį panaudojimą. Šiame straipsnyje nagrinėjama informacijos išgavimo iš liktinių informacinių sistemų problema, pristatomas modelių transformacijomis paremtas informacinių sistemų modernizavimo metodas, skirtas programinio kodo transformavimui į abstrakčios sintaksės medžio (*angl.* Abstract Syntax Tree Metamodel, ASTM) ir žinių radimo (*angl.* Knowledge Discovery Metamodel, KDM) metamodelius. Pateikiama SQL kalbai pritaikyta sukurto metodo eksperimentinė validacija.

Pagrindiniai žodžiai: architektūra grindžiamas modernizavimas, atgalinė inžinerija, ASTM, KDM, Gra2Mol,

Įvadas

Informacinių sistemų (IS) modernizavimas — tai nauja kryptis IS inžinerijos srityje. Nedideliems IS pakeitimams priežiūros metu nebetenkinant keliamų reikalavimų, atsiranda poreikis naudoti sudėtingesnius ir daugiau galimybių suteikiančius IS atnaujinimo metodus. Liktinėms IS būdingas monolitiškumas, nelankstumas, aiškumo trūkumas, sudėtingas aptarnavimas dėl tinkamų specialistų trūkumo, pasenusi vartotojo sąsaja, didelė IS eksploatacijos kaina. Kai verslo poreikiai pradeda augti greičiau, nei spėjama atlikti būtinus darbus, sistemos funkcionalumas ima nebetenkinti ir vis didėja atotrūkis tarp verslo poreikių ir IS galimybių, problemų sprendimui gali būti pasitelktas IS modernizavimas.

Šio straipsnio tyrimo objektas yra IS modernizavimas naudojant modelių transformacijas. Darbo tikslas yra sukurti metodą, skirtą pakartotinai panaudoti liktinėje IS esančią verslo logiką transformuojant programinį kodą į ASTM ir KDM modelius.

Straipsnio struktūra tokia: pirmoje dalyje pateikta susijusių su IS modernizavimo tema darbų apžvalga, antroje pristatomas naujas IS modernizavimo metodas, kurio etapai (programinio kodo transformavimas į ASTM bei ASTM transformavimas į KDM) detalizuojami trečiame bei ketvirtame skyriuose, o penktame skyriuje pateikiama metodo eksperimentinė validacija.

1. Tyrimų IS modernizavimo srityje apžvalga

Vis plačiau taikant modelių transformacijas tiesioginėje IS inžinerijoje, kyla poreikis pritaikyti modelių transformacijas ir liktinių IS modernizavimui. Liktinė IS – tai informacinė sistema, kuri priešinasi modifikavimui ar evoliucijai, kurie reikalingi tam, kad būtų tenkinami nuolat kintantys verslo reikalavimai (Bakehouse and Wakefield, 2005).

Pastaraisiais metais suaktyvėjo tyrimai IS modernizavimo srityje. Yra sukurta keletas sprendimų šia tema. Juos galima suskirstyti į dvi grupes: pirmoji akcentuoja programinio kodo analizės problemą, antroji – darbą su modeliais, apibūdinančiais liktines sistemas. Iš pirmosios grupės pažymėtina Gra2Mol (angl. Grammar to Model Language) (Izquierdo and Molina, 2009) programavimo kalba, skirta programavimo kalbų gramatikoms ir metamodeliams – modeliams, išreiškiantiems modeliavimo kalbos abstrakčią sintaksę, susieti. Ši kalba leidžia aprašyti taisykles, kurios apibrėžia programavimo kalbai būdingas struktūras ir susieja jas su pasirinktos specifikacijos metamodeliu. Taip sukuriamas įrankis programavimo kalbos transformavimui į modelį, atitinkantį pasirinktą specifikaciją. Šiai metodų grupei taip pat priskiriami įvairūs automatiniai teksto analizės įrankiai (parser) (Fisher et al., 2007), skirti programinio kodo ar duomenų apdorojimui siekiant išgauti struktūrizuotą informaciją.

Antrojoje sprendimų grupėje šiuo metu svarbiausias tyrimų objektas yra pačios modelių specifikacijos, skirtos liktinių IS analizei. Liktinių IS modernizavimo tyrimus aktyviai vykdo Object Management Group (OMG) Architecture Driven Modernization (ADM) Task Force grupė. Modernizavimo tikslams jau yra sukurtos tokios specifikacijos, kaip Abstract Syntax Tree Metamodel (ASTM) (OMG ADM Task Force, 2011), skirtas programinio sintaksės medžio struktūrizavimui, Knowledge Discovery Metamodel (KDM) (OMG ADM Task Force, 2011a), leidžiantis aprašyti veikiančią IS ir jos aplinką ir Software Metrics Metamodel (SMM) (OMG ADM Task Force, 2010), skirtas programų sistemos metrikų aprašymui. KDM laikomas pagrindiniu metamodeliu, skirtu darbui IS modernizavimo srityje (OMG ADM Task Force, 2011a). Artimiausiu metu ruošiama dar keletas specifikacijų, apimančių kitas ADM tyrimų sritis: Analysis Package, Testing Package, Visualization Package, Refactoring package, Transformation Package. Modeliais paremtą IS modernizavimą akcentuoja Eclipse Modeling Project. Šio projekto pasiūlytas karkasas MoDisco yra skirtas modeliais paremtų IS modernizavimo priemonių kūrimui. MoDisco pagrindas yra atradimo priemonės (discoverer), skirtos transformacijų būdu išgauti modelius iš kitų modelių, programinio kodo ar fizinių programų sistemos elementų (Bruneliere, 2010).

Kitame skyriuje aprašomas metodas, apimantis tiek pirmosios, tiek ir antrosios grupės IS modernizavimo metodų savybes: programinio kodo analizę ir modelių transformacijas.

2. Metodo aprašymas

Siekiant iš programinio kodo išgauti informaciją apie IS sandarą, kaip bazinis modelis pasirinktas KDM modelis, atitinkantis OMG specifikaciją. KDM modelis pasirinktas dėl jo aktualumo bei pagrindinės paskirties – informacijai apie IS ir jos veikimo aplinką išreikšti.

Sukurtas metodas, skirtas KDM modelio iš programinio kodo išgavimui. Sukurtą metodą sudaro du etapai:

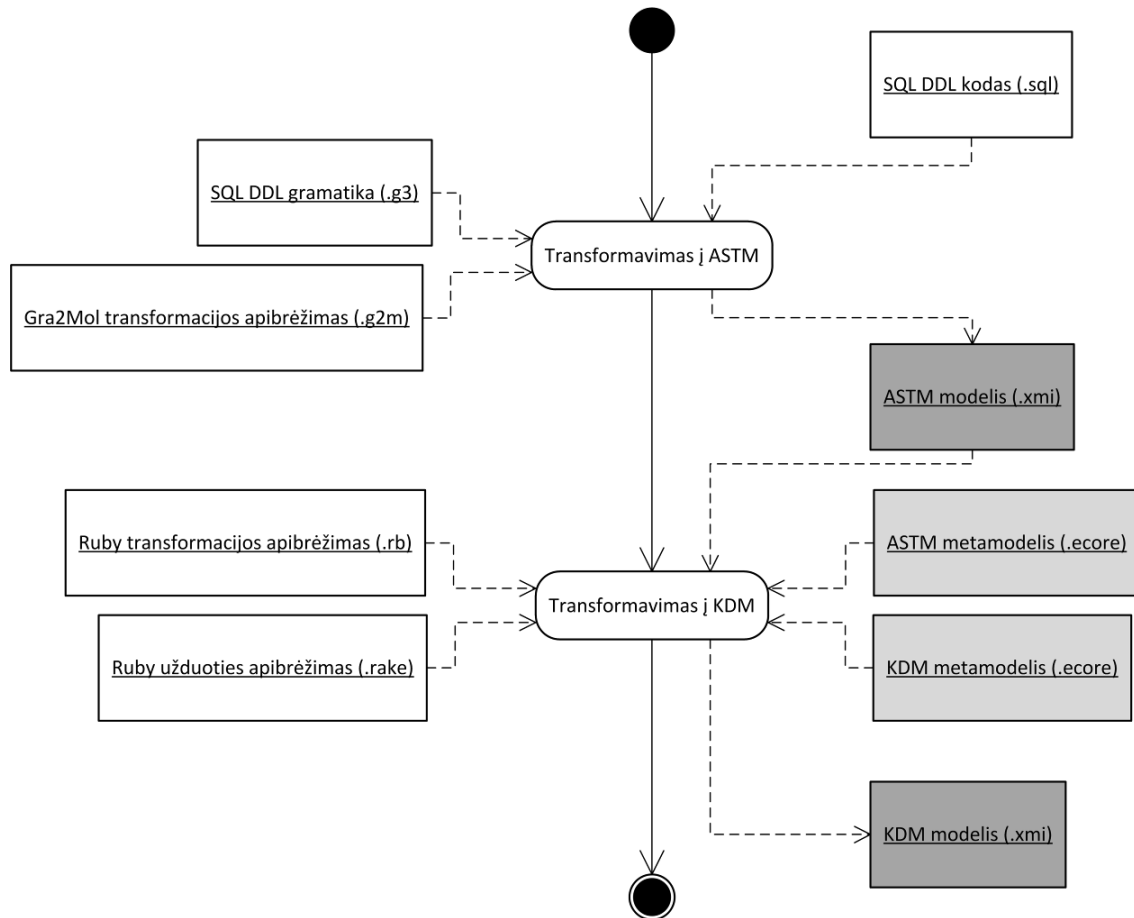
1. Programinio kodo analizė bei jo transformavimas į ASTM modelį;
2. ASTM modelio transformavimas į KDM modelį.

Metodo principas remiasi kodo transformavimo į modelį bei modelių transformavimo užduočių sujungimu, pasiūlytu Izquierdo ir Molina (Izquierdo and Molina, 2009). Tokiu būdu šio metodo negalima priskirti tik vienai modernizavimo metodų grupei, kadangi šiame metode taikomas naujas požiūris, sujungiantis minėtų modernizavimo metodų grupių savybes – informacijos išgavimas iš programinio kodo bei modernizavimui skirtų metodų analizė. Šiame metode siūloma iš programinio kodo pirmiausia išgauti ASTM modelį, atitinkantį OMG specifikaciją. Poreikis išgauti ASTM modelį atsiranda todėl, kad transformuoti programinį kodą tiesiogiai į KDM yra sudėtinga. Tiesioginė transformacija nebūtų universali, kadangi pritaikymas skirtingoms programavimo kalboms užimtų daug laiko ir turėtų apimti visą transformavimo procesą. Išskaidymas į du etapus leidžia padalinti transformavimo uždavinį į skirtingų tipų užduotis – transformavimą į ASTM modelį, skirtą programiniam kodui išreikšti, taip atsiribojant nuo programavimo kalbos, bei išgauto modelio transformavimą į kitos specifikacijos modelį, šiuo atveju – į KDM. Tarpinis ASTM modelis suteikia galimybę pasirinkti kitą proceso kelią, siekiant išgauti ne KDM, o kitos specifikacijos, pvz., Intermediate Object Model (IOM) (Newcomb, 2005), modelį. KDM modelis taip pat gali būti laikomas tarpiniu tiek dėl savo paskirties – metaduomenų apskaitinimui tarp sistemų, tiek dėl galimybės panaudoti šį modelį tolesnėms transformacijoms, siekiant išgauti, pavyzdžiui, vieningos objektinio modeliavimo kalbos (angl. Unified Modeling Language – UML) specifikacijos modelius.

Metodo eksperimentiniam patikrinimui pasirinkta SQL programavimo kalba, kadangi šioje kalboje vyrauja gana nesudėtingos programinio kodo struktūros, todėl pakanka gana paprasto gramatikos aprašo. Metodo, pritaikyto SQL kalbai, veiklos diagrama (angl. activity diagram) pateikta 1 paveiksle. Diagramoje balta spalva pažymėti metodo parametriniai objektai, šviesiai pilka spalva pavaizduoti viešai prieinami objektai, taip pat naudojami kaip metodo parametrai, o tamsiai pilka spalva pažymėti proceso išeigos objektai, t.y. gaunami modeliai. Pastebėtina, kad didžioji dalis transformavimo į ASTM parametrinių objektų (gramatika, transformacijos apibrėžimas), kiekvienai programavimo kalbai gali būti pakartotinai panaudoti skirtingiems programinio kodo fragmentams analizuoti. Transformavimas į KDM pasižymi dar didesniu universalumu, kadangi naudojant pakankamai išsamų Ruby transformacijos apibrėžimą, tokias transformacijas galima taikyti iš skirtingų programavimo kalbų gautiems ASTM. Išgautas KDM modelis gali būti panaudotas tolesnėms transformacijoms, pavyzdžiui siekiant išgauti BPMN (angl. Business Process Modeling Notation) ar UML modelius, IS elementų analizei ar skirtingų IS suderinamumui užtikrinti. Detalesnis metodo etapų aprašymas pateiktas kituose skyriuose.

3. Programinio kodo transformavimas į ASTM

Darbe panaudota viešai prieinama Microsoft SQL Server duomenų bazė AdventureWorks. Iš šios duomenų bazės eksperimentui pasirinktos dvi lentelės Contact ir StoreContact. Vienos iš jų, StoreContact, sukūrimo kodas pateiktas 1 lentelėje.



1 pav. SQL programinio kodo transformavimo į KDM modelį metodo veiklos diagrama.

1 lentelė. Reliacinės lentelės StoreContact sukūrimo kodas ir SQL DDL gramatikos kodas.

StoreContact sukūrimo kodas	SQL DDL gramatikos g3 formatu kodo fragmentas
<pre>CREATE TABLE StoreContact(CustomerID INT NOT NULL, ContactID INT NOT NULL, ContactTypeID INT NOT NULL, ModifiedDate datetime NOT NULL);</pre>	<pre>create_table : 'CREATE TABLE' ID '(' column (' column)* (' constraint)* ')' '; ; column: ID TYPE '(' NUMBER ')'? ISNULL? ;</pre>

SQL gramatikos specifikacijai aprašyti panaudota Gra2Mol autorių pasiūlyta (Izquierdo and Molina, 2009) PL/SQL Data Definition Language (DDL) gramatikos specifikacija g3 formatu, kuri adaptuota SQL kalbai. Šios specifikacijos fragmentas, apibrėžiantis taisyklės lentelių struktūrai bei laukų pavadinimams ir tipams nustatyti, pateiktas 1 lentelėje. Sukurtas SQL transformacijos į ASTM apibrėžimas Gra2Mol kalba, g2m formatu. Literatūroje galima rasti jau aprašytų transformacijų iš kitų programavimo kalbų – Java, Oracle Forms XML, ANTLR (Izquierdo and Molina, 2009), tačiau SQL kalbai sukurtos išsamios transformacijos apibrėžimo nėra. Šiame metode naudotas transformacijos iš SQL į ASTM modelį aprašymas sukurtas remiantis Gra2Mol

pateikiamu neišsamiau PL/SQL transformacijos į ASTM apibrėžimu. Transformacijos metu gautas ASTM modelio kodo fragmentas pateiktas 2 lentelėje.

2 lentelė. ASTM ir KDM modeliai, kuriuose išreikšta reliacinė lentelė StoreContact.

ASTM modelio fragmentas, kuriame išreikšta reliacinė lentelė StoreContact	KDM modelio fragmentas, kuriame išreikšta reliacinė lentelė StoreContact
<pre> <?xml version="1.0" encoding="ISO-8859-1"?> <gastm:Project xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:gastm="http://modelum.es/modernization/gastm" xmlns:sastm="http://modelum.es/modernization/sastm"> <files language="SQL"> <...> <fragments xsi:type="sastm:RDBTableDefinition"> <identifierName nameString="StoreContact"/> <Column> <identifierName nameString="CustomerID"/> <type xsi:type="sastm:RDBNumber"/> </Column> <Column> <identifierName nameString="ContactID"/> <type xsi:type="sastm:RDBNumber"/> </Column> <Column> <identifierName nameString="ContactTypeID"/> <type xsi:type="sastm:RDBNumber"/> </Column> <Column> <identifierName nameString="ModifiedDate"/> <type xsi:type="sastm:RDBDate"/> </Column> </fragments> </files> </gastm:Project> </pre>	<pre> <kdm:Segment xmlns:code='http://www.eclipse.org/MoDisco/kdm/code' xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance' xsi:schemaLocation='http://www.eclipse.org/MoDisco/kdm/data ../metamodels/kdm.ecore#//data http://www.eclipse.org/MoDisco/kdm/kdm ../metamodels/kdm.ecore#//kdm http://www.eclipse.org/MoDisco/kdm/code ../metamodels/kdm.ecore#//code' xmi:version='2.0' xmlns:xmi='http://www.omg.org/XMI' xmlns:kdm='http://www.eclipse.org/MoDisco/kdm/kdm' xmlns:data='http://www.eclipse.org/MoDisco/kdm/data'> <model name='DDL Datatypes' xsi:type='code:CodeModel'> <codeElement name='SQL.Datatypes' xsi:type='code:LanguageUnit'> <codeElement xsi:type='code:BooleanType'/> <codeElement name='sql varchar' xsi:type='code:StringType'/> <codeElement name='sql int' xsi:type='code:IntegerType'/> <codeElement name='sql decimal' xsi:type='code:DecimalType'/> <codeElement name='sql date' xsi:type='code:DateType'/> </codeElement> </model> <model name='Relational Schema' xsi:type='data:DataModel'> <...> <dataElement name='StoreContact' xsi:type='data:RelationalTable'> <itemUnit name='CustomerID' type='#//@model.0/@codeElement.0/@codeElement.2' xsi:type='code:ItemUnit' /> <itemUnit name='ContactID' type='#//@model.0/@codeElement.0/@codeElement.2' xsi:type='code:ItemUnit' /> <itemUnit name='ContactTypeID' type='#//@model.0/@codeElement.0/@codeElement.2' xsi:type='code:ItemUnit' /> <itemUnit name='ModifiedDate' type='#//@model.0/@codeElement.0/@codeElement.4' xsi:type='code:ItemUnit' /> </dataElement> </dataElement> </model> </kdm:Segment> </pre>

Gauti rezultatai rodo, kad transformacija įvykdyta sėkmingai, tačiau ASTM modelio korektiškumą tiksliausiai galima įvertinti tik po transformacijos į KDM modelį, kadangi tokia transformacija galima tik iš teisingai suformuoto ASTM modelio. ASTM modelio transformacija į KDM modelį aprašyta kitame skyriuje.

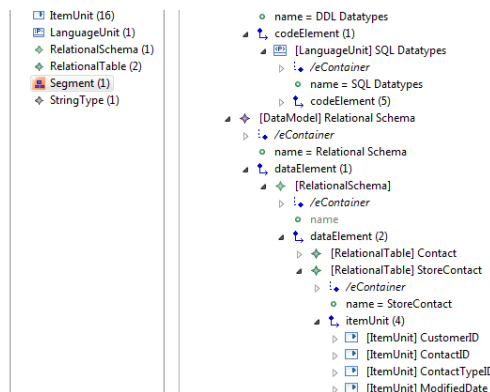
4. ASTM transformavimas į KDM

ASTM transformacijos į KDM realizacijai pasirinkti kiti transformavimo įrankiai, kadangi ši transformacija nebereikalauja programinio kodo analizės. Transformacijai naudojamas Izquierdo ir Molina (Izquierdo and Molina, 2009) pasiūlytas transformacijos

iš ASTM į KDM apibrėžimas, Ruby užduoties apibrėžimas ir viešai prieinamos ASTM metamodelio ir KDM metamodelio specifikacijos ecore formatu, apibrėžtos OMG (OMG ADM Task Force, 2011), (OMG ADM Task Force, 2011a). Transformavus ankstesniame etape išgautą ASTM modelį buvo išgautas KDM modelis, pateiktas 2 lentelėje. Modelį sudaro KDM segmentas, kuriame sukuriama du modeliai: kodo modelis, kuriame aprašomi transformacijos metu nustatyti duomenų tipai, bei duomenų modelis, kurį sudaro du duomenų elementai, atitinkantys pradinės lentelės Contact ir StoreContact. Tokia struktūra atspindi vieną iš KDM modelio privalumų – informacijos struktūrizavimą siekiant didesnio liktinės IS aiškumo. Atskleisti tokios struktūros naudą leidžia MoDisco Model Browser, kuriame pateikiami sugrupuoti KDM modelio elementai hierarchine tvarka, pvz. duomenų bazės schemas elementas, apimantis dvi lenteles, kurios atitinkamai apima jose esančių laukų ir jų savybių sąrašus. Gauti rezultatai rodo, kad išgautas KDM modelis yra validus, t.y. atitinka specifikaciją. Kartu tai patvirtina ir ASTM modelio validumą, kadangi tik iš tinkamai suformuoto ASTM modelio transformavimo būdu galima išgauti KDM modelį.

5. Eksperimentinė metodo validacija

Tyrimo tikslas – išgauti KDM modelį iš SQL DDL programinio kodo. Šiam tikslui pasiekti panaudota Gra2Mol programavimo kalba realizuota transformacija iš programinio kodo į ASTM modelį bei Ruby programavimo kalba realizuota ASTM transformacija į KDM. Eksperimentinės metodo validacijos metu iš programinio SQL DDL kodo, aprašančio dvi lenteles, vienos iš kurių, StoreContact, aprašas pateiktas 1 lentelėje, panaudojus SQL DDL gramatikos specifikaciją ir transformacijos aprašymą, kurio fragmentas pateiktas 1 lentelėje, buvo išgautas ASTM modelis, kurio dalis, aprašanti lentelę StoreContact, pateikta 2 lentelėje. ASTM modelis panaudotas kaip parametras kitoje transformacijoje siekiant išgauti KDM modelį, kurio fragmentas pateiktas 2 lentelėje. KDM modelio validacija atlikta naudojant MoDisco Model Browser įrankį. Išgauto KDM modelio vaizdas šio įrankio aplinkoje pateiktas 2 paveiksle. Metodo sėkmingą pritaikymą rodo gauti rezultatai – KDM modelis gali būti apdorojamas MoDisco Model Browser programine įranga, o tai rodo, kad tiek ASTM, tiek KDM modeliai atitinka savo specifikacijas.



2 pav. KDM modelio vaizdas MoDisco Model Browser įrankyje, kuriame išreikšta reliacinės lentelės StoreContact informacija KDM modelio struktūroje.

Išvados

Susijusių darbų analizė parodė, kad IS modernizavimo problemos pastaruoju metu skirstomas į du uždavinius: programinio kodo transformavimą į modelį bei modernizavimui skirtų modelių transformacijos į kitus modelius. KDM yra vienas naujausių OMG metamodelių specifikacijų IS modernizavime, skirtas išreikšti IS sandarą ir jų veikimo aplinką. KDM modelis pasirinktas kaip bazinis transformacijomis paremtame IS modernizavimo metode. Pasiūlytas metodas, paremtas kodo transformavimu į modelį ir modelių transformavimu užduočių sujungimu. Kodo transformacijai į ASTM modelį adaptuota SQL gramatika bei sukurtas Gra2Mol transformacijos iš SQL į ASTM apibrėžimas. Modelių transformacijai iš ASTM į KDM panaudotas Izquierdo ir Molina (Izquierdo and Molina, 2009) pasiūlytas transformacijos iš ASTM į KDM apibrėžimas, Ruby užduoties apibrėžimas ir OMG apibrėžtos ASTM (OMG ADM Task Force, 2011) bei KDM (OMG ADM Task Force, 2011a) metamodelių specifikacijos. Atlikus tyrimą, nustatyta, kad sukurtas metodas tinka SQL programinio kodo transformavimui į ASTM ir KDM modelius. Išgautas KDM modelis gali būti apdorojamas KDM skirtais įrankiais, o tai rodo, kad modelis atitinka specifikaciją.

Tolimesni darbai šio metodo tobulinimui gali apimti metodo pritaikymo SQL kalbai praplėtimas siekiant išreikšti ne tik duomenų struktūras, bet ir ryšius tarp jų. Taip pat galimi metodo pritaikymo kitoms programavimo kalboms tyrimai bei KDM modelio transformavimo į UML modelius galimybių analizė.

Literatūra

- Bakehouse, G., Wakefield, T. (2005). Legacy Information Systems. ACCA services.
- Bruneliere, H., Cabot, J., Jouault, F., Madiot, F. (2010). MoDisco: a generic and extensible framework for model driven reverse engineering. ASE '10 Proceedings of the IEEE/ACM International Conference on Automated Software Engineering.
- Fisher, G., Lusiardi, J., Wolff von Gudenberg, J. (2007). Abstract Syntax Trees – and their Role in Model Driven Development. International Conference on Software Engineering Advances (ICSEA 2007).
- Izquierdo, J. L. C., Molina, J. G. (2009). A domain specific language for extracting models in software modernization. Model Driven Architecture - Foundations and Applications, Volume 5562, 82-97.
- Newcomb, P. (2005). Modernization of the Naval Warfare Systems Command (NETWARCOM) Engineering Operational Sequencing System (EOSS). OMG's Second Annual Architecture-Driven Modernization Workshop.
http://www.omg.org/news/meetings/workshops/ADM_2005_Proceedings_FINAL/4-1_Newcomb.pdf
- OMG ADM Task Force (2011). Abstract Syntax Tree Metamodel.
<http://www.omg.org/spec/ASTM/1.0>
- OMG ADM Task Force (2011a). Knowledge Discovery Metamodel.
<http://www.omg.org/spec/KDM/1.3/Beta2>
- OMG ADM Task Force (2010). Software Metrics Metamodel.
<http://www.omg.org/spec/SMM/1.0/Beta2>

MODEL TRANSFORMATION BASED INFORMATION SYSTEM MODERNIZATION

Olegas Vasilecas, Simas Zikaras, Aidas Šmaižys

Summary

Information systems begin to date increasingly faster because of rapidly changing business environment. Usually, small changes are not sufficient to adapt complex legacy information systems to changing business needs. A new functionality should be installed with the requirement of putting business data in the smallest possible risk. Information systems modernization problems are being analyzed in this paper and a method for information system modernization is proposed. It involves programming code transformation into abstract syntax tree metamodel (ASTM) and model based transformation from ASTM into knowledge discovery model (KDM). The method is validated on example for SQL language.

Key words: architecture driven modernization, reverse engineering, ASTM, KDM, Gra2Mol.