

ЈЕДАН ПРИСТУП МОДЕЛОМ ВОЂЕНОГ ИНЖЕЊЕРИНГА ИНФОРМАЦИОНИХ СИСТЕМА**A MODEL-DRIVEN APPROACH TO INFORMATION SYSTEM ENGINEERING**

Вељко Бубњевић, Факултет техничких наука, Нови Сад

**Студијски програм – СОФТВЕРСКО
ИНЖЕЊЕРСТВО И ИНФОРМАЦИОНЕ
ТЕХНОЛОГИЈЕ**

Кратак садржај – Циљ овог истраживања је да се, применом инжењеринга вођеног моделима, формално дефинишу и практично примене поступци трансформација који омогућавају поступни прелазак са концептуалног на имплементациони ниво. Полазећи од концептуалног модела информационог система заснованог на концепту типа форме, предложени приступ омогућава извођење релационе шеме базе података и њеног имплементационог описа за изабрану SQL (Structured Query Language) платформу. Дефинисан је методолошки оквир који обухвата проширење наменског језика FTDSL (Form Type Domain-Specific Language) развојем графичке синтаксе и имплементацију Model-to-Model и Model-to-Text трансформација, чиме се обезбеђује интегрисан и аутоматизован процес инжењеринга информационих система заснован на моделима.

Кључне речи : трансформације шема база података, концептуално моделовање, концепт типа форме, MDE/CASE алати

Abstract – The aim of this research is to formally define and practically apply transformation procedures based on model-driven engineering (MDE), enabling a gradual transition from the conceptual to the implementation level. Starting from the conceptual model of an information system (IS) based on the form type concept, the proposed approach enables the derivation of a relational database schema and its implementation description for the selected SQL (Structured Query Language) platform. A methodological framework has been defined, encompassing the extension of the domain-specific language FTDSL (Form Type Domain-Specific Language) through the development of its graphical syntax and the implementation of Model-to-Model and Model-to-Text transformations, thereby providing an integrated and automated model-driven information system engineering process.

Keywords: database schema transformation, conceptual modeling, form type concept, MDE/CASE tools

НАПОМЕНА: Овај рад проистекао је из мастер рада чији ментор је др Славица Кордић, ванр. проф.

1. УВОД

Традиционални начин изградње IS полазио је од прикупљања корисничких захтева, најчешће у неформалном облику, који су касније претварани у формалне моделе система. Такви модели су неретко садржали детаље везане за конкретну имплементациону платформу, док је програмски код настајао углавном ручним програмирањем. Савремени приступи, засновани на моделима полазе од идеје да се развој софтвера заснива на низу формалних трансформација модела различитих нивоа апстракције, почев од рачунарски независних модела (*Computation-Independent Model* – CIM), све до модела зависних од платформе и извршивог кода.

У том правцу, на Факултету техничких наука Универзитета у Новом Саду развијају се методе за платформски независно моделовање IS и алгоритми за аутоматску трансформацију таквих модела у извршиве прототипове IS. У оквиру ових истраживања развијен је алат под називом IIS*Case (*Integrated Information Systems CASE*), који припада класи интегрисаних CASE (*Computer Aided Software Engineering*) производа. Његова основна улога је да подржи што већи број активности у процесу развоја IS.

Иако је у алату IIS*Case већ реализован реверзни инжењеринг на основу мета-модела, презентованих у [1][2], процес *forward* инжењеринга у оквиру MDE окружења до сада није био у потпуности подржан. Уместо тога, IIS*Case је користио GPL (*General Purpose Language*) Java као засебан приступ, што је стварало одређени јаз у односу на принципе развоја софтвера вођеног моделима. У овом раду је развијена нова, унапређена верзија алата IIS*Case, у којој је процес инжењеринга IS у потпуности реализован применом MDE приступа. Реализација је спроведена, применом Model-to-Model (M2M) и Model-to-Text (M2T) трансформација у области инжењеринга IS, као и проширењем постојећег наменског језика FTDSL развојем његове графичке синтаксе, чиме се обезбеђује комплетан развој IS у оквиру јединственог и интегрисаног MDE окружења.

2. КОНЦЕПТУАЛНО МОДЕЛОВАЊЕ IS

Инжењеринг IS у алату IIS*Case започиње концептуалним моделовањем IS, где пројектант на основу захтева и докумената креира типове форми,

којима се описују правила употребе података у реалном систему. Тип форме се може схватити као општи облик пословних докумената и екранских форми путем којих корисник ступа у интеракцију са *IS*. Типови форми представљају основу за генерисање почетног скупа атрибута и идентификацију полазног скупа ограничења, које *IIS*Case* користи као улазне параметре у поступку формирања релационе шеме базе података.

2.1. Концепт типа форме

Генерализацијом анализе садржаја екранских и штампаних форми (докумената) долази се до појма типа форме [3]. Тип форме F је структура стабла чије чворове представљају типови компоненти. Формално, тип форме је именована петорка $F(O, \varphi, Man, S_s(F), C_F)$, над скупом обележја $W(F)$, где је:

- $O = \{N_i(Q_i, C_i, Rel(N_i)) \mid i = 1, \dots, m\}$ скуп типова компоненти,
- $\varphi \subseteq O \times O$ је релација која над скупом O дефинише структуру стабла,
- Man је функција обавезности: $Man: \varphi \rightarrow \{M, O\}$, која сваком $(N_i, N_j) \in \varphi$ придружује вредност M (обавезна) или O (опциона) веза,
- $S_s(F)$ је параметар чија вредност Y указује на то да тип форме F учествује, а вредност N да тип форме F не учествује у формирању шеме базе података,
- C_F скуп ограничења типа форме F , при чему скуп обележја $W(F)$ задовољава услове (1) и (2) :

$$\bigcup_{i=1}^m Q_i = W(F), \quad (1)$$

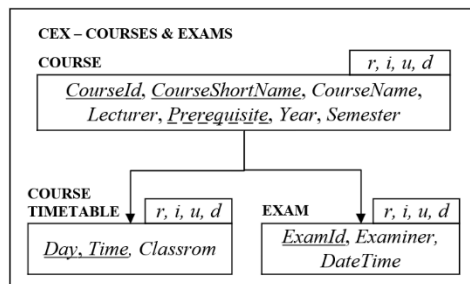
$$(\forall N_i, N_j \in O)(i \neq j \Leftrightarrow Q_i \cap Q_j = \emptyset). \quad (2)$$

2.2. Развој графичке синтаксе језика FTDSL

Да би се могао спровести процес *forward* инжењеринга алата *IIS*Case* користећи приступ развоју софтвера вођеног моделима, било је неопходно обезбедити подршку за концептуално моделовање *IS* употребом наменског језика *FTDSL*, као улазног модела у ланцу *M2M* трансформација. *FTDSL* је наменски језик намењен за концептуално моделовање *IS* путем типова форми, који омогућава формалну спецификацију структура и ограничења на нивоу *PIM (Platform-Independent Model)* модела. У оквиру овог истраживања развијена је графичка синтакса наменског језика *FTDSL*, оригинално представљеног у [2]. Док је *FTDSL* у почетној верзији био заснован на текстуалној нотацији, која је захтевала добро познавање синтаксних правила и структура самог језика, предложено проширење у овом раду уводи графички слој који омогућава интуитивније концептуално моделовање *IS*. Графичка синтакса омогућава визуелну спецификацију типова форми, компоненти и њихових односа, што приближава процес моделовања перцепцији крајњег корисника и олакшава комуникацију између аналитичара и доменских стручњака. Истовремено, задржана је формална повезаност са апстрактном синтаксом језика

FTDSL, што омогућава његову интеграцију у *MDE* оквир и *M2M* ланац трансформација.

На слици 1 приказан је тип форме *Courses & Exams*, док је на слици 2 приказан модел истог типа форме моделован коришћењем графичке синтаксе наменског језика *FTDSL* употребом *Eclipse Sirius* алата.



Слика 1. Тип форме Courses & Exams

Слика 2. Модел типа форме Courses & Exams специфициран употребом графичке синтаксе језика FTDSL

3. ТРАНСФОРМАЦИЈА КОНЦЕПАТА ПРОШИРЕНОГ *IIS*Case* МЕТА-МОДЕЛА У КОНЦЕПТЕ ПРОШИРЕНОГ ГЕНЕРИЧКОГ МЕТА-МОДЕЛА

Након концептуалног моделовања, следећи корак представља пројектовање подшеме релационе базе података за изабрани апликативни систем. Подшема релационе базе података која задовољава услов *3NF* добија се поступком нормализације применом модификованог Бернштајновог алгоритма синтезе. Реализација модификованог Бернштајновог алгоритма синтезе спроведена је употребом ланца *M2M* трансформација. Као изворни модел коришћен је модел који је у складу са мета-моделом дефинисаним у оквиру апстрактне синтаксе језика *FTDSL*, док је као циљни модел коришћен модел који је у складу са Генеричким мета-моделом који је дефинисан у складу са теоријским основама релационог модела података [1]. Оба мета-модела су, у циљу подршке процесу *forward* инжењеринга, проширена новим концептима, при чему се први у наставку рада означава као Проширени *IIS*Case* мета-модел, а други као Проширени генерички мета-модел.

Како модификовани алгоритам синтезе обухвата више корака, који нису могли бити изражени потпуно декларативно, било је неопходно процес трансформације поделити на више целина, где сваки

излаз из претходне трансформације представља улаз у наредну. У ту сврху дефинисано је шест међусобно повезаних трансформација, које заједно омогућавају потпуну реализацију поступка *forward* инжењеринга: *FormConstraintsIdentification*, *Synthesis3NF*, *PKCandidatesIdentification*, *PKSelection*, *PKPropagation*, *InterRelationalConstraintsDefinition*.

3.1. Идентификација полазног скупа ограничења

FormConstraintsIdentification је егзогена *out-place* трансформација којом се трансформише модел који је у складу са IIS*Case мета-моделом у модел који је у складу са Проширеним генеричким мета-моделом. Ова трансформација уведена је како би се из концептуалног модела идентификовао полазни скуп ограничења који ће представљати улаз у алгоритам синтезе. Трансформацијом се идентификују следећа ограничења са типова форми: функционалне зависности, нефункционални односи, специјалне функционалне зависности са уграђеним ограничењима јединствености и уграђене зависности садржавања. Поступак идентификације полазног скупа ограничења у потпуности се заснива на формалним правилима дефинисаним у [4][5].

Пример 3.1. Посматра се тип форме са слике 2.1. Применом правила из *FormConstraintsIdentification* трансформације добија се следећи скуп функционалних зависности:

$F(F) = \{CourseId \rightarrow CourseShortName, CourseId \rightarrow CourseName, CourseId \rightarrow Lecturer, CourseId \rightarrow Prerequisite, CourseId \rightarrow Year, CourseId \rightarrow Semester, CourseId + Day + Time \rightarrow Classroom, CourseId + ExamId \rightarrow Examiner, CourseId + ExamId \rightarrow DateTime, CourseShortName \rightarrow CourseId, Prerequisite \rightarrow CourseId\}$, при чему је знаком $\rightarrow$ означена специјална функционална зависност. Применом правила за идентификацију нефункционалних односа добија се следећи скуп нефункционалних односа:

$$NF(F) = \{CourseId + Day + Time \rightarrow \theta_1, CourseId + ExamId \rightarrow \theta_2\}.$$

3.2. Алгоритам синтезе

Након идентификације полазног скупа ограничења следећи корак представља примена алгоритма синтезе. Алгоритам синтезе је специјално унапређен за практичне примене потреба алата IIS*Case, па се из тог разлога назива модификованим. Трансформација којом се спроводи алгоритам синтезе јесте *Synthesis3NF*. Улаз у ову трансформацију представљају два модела, један који је излаз из претходне трансформације, и други који је полазни концептуални модел у складу са IIS*Case проширеним мета-моделом. Потреба за концептуалним моделом у оквиру ове трансформације произилази из чињенице да се поједина ограничења, дефинисана унутар генерисане релационе шеме, могу закључити само на основу информација које су доступне у концептуалном моделу. Концептуални модел, представља неопходан извор семантичког контекста који омогућава правилну идентификацију, и формирање ограничења у релационој шеми базе података. По изласку из ове

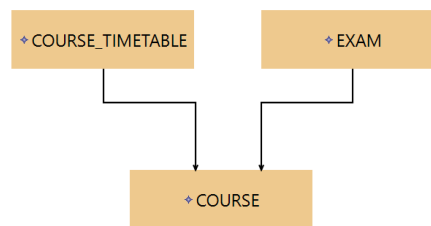
трансформације добија се скуп шема релација који задовољава услов *3NF*.

Пример 3.2. Посматра се полазни скуп ограничења из примера 3.1. Применом алгоритма синтезе у оквиру *Synthesis3NF* трансформације добија се следећи скуп шема релација подшеме:

- *Course*({*CourseId*, *CourseName*, *CourseShortName*, *Lecturer*, *Year*, *Semester*, *Prerequisite*}, {*CourseId*, *CourseName*, *CourseShortName*}, {*Prerequisite*}),
- *Course_Timetable*({*Day*, *Classroom*, *CourseId*, *Time*}, {*Day+Time+CourseId*}) и
- *Exam*({*ExamId*, *CourseId*, *Examiner*, *DateTime*}, {*ExamId+CourseId*}).

У оквиру ове трансформације врши се генерисање и графа затварања [4][5]. Граф затварања представља погодну дијаграмску технику за представљање структуре шеме релационе базе података. Сваки чвор у графу затварања представља једну шему релације у шеми базе података, док свака грана између чворова означава да скуп обележја надређеног чвора функционално одређује, тј. затвара, скуп обележја подређеног чвора.

Пример 3.3. На слици 3 приказан је граф затварања у *Eclipse Sirius* едитору који је добијем *Synthesis3NF* трансформацијом на скуп ограничења из примера 3.1.



Слика 3. Генерисани граф затварања

3.3. Избор кандидата за примарни кључ

По изласку из алгоритма синтезе свака шема релације добија скуп еквивалентних кључева. Један од тих еквивалентних кључева пројектант треба да одабере за примарни кључ. У контексту модификованог алгоритма синтезе не може сваки еквивалентни кључ да испуњава услов да буде кандидат за примарни кључ. Ова тема детаљно је разрађена у [5]. Провера испуњености услова кандидата за примарни кључ спроведена је у оквиру трансформације *PKCandidatesIdentification*, која за сваки кључ утврђује да ли испуњава услов кандидата за примарни кључ и сходно томе га означава. Уколико ниједан кључ у појединој шеми релације не испуњава услов кандидата за примарни кључ, извршавање ланца трансформација се прекида и корисник се обавештава о томе. Након тога се извршава трансформација *PKSelection*, која насумично бира један од кандидата за примарни кључ и проглашава га за примарни. Обе трансформације представљају ендogene *in-place* трансформације, а као резултат се добија модел шеме базе података у којем свака шема релације има дефинисан примарни кључ.

3.4. Трансформација скупа шема релација са некоректним простирањем кључа

По изласку из претходне трансформације свака шема релација поседује примарни кључ. Међутим, приликом одабира примарних кључева могло је да дође до њиховог некоректног простирања, односно, да се деси да су различити еквивалентни кључеви једне шеме релације изабрани за стране кључеве. Трансформација *PrimaryKeysPropagation* има за циљ да изврши трансформацију скупа шема релација са некоректним простирањем кључа у скуп шема релација са коректним простирањем кључа, поштујући алгоритам трансформације у модификованом алгоритму синтезе.

3.5. Генерисање међурелационих ограничења

InterRelationalConstraintsDefinition трансформацијом се завршава процес генерисања подшеме релационе базе података. Ова трансформација генерише следеће типове ограничења:

- основна, инверзна и проширена ограничења референцијалног интегритета,
- ограничења основних и инверзних референцијалних интегритета заснованих на нетривијалним зависностима садржавања,
- међурелациона ограничења зависног скупа шема релација и
- проширена ограничења торке.

3.6. Мапирање концепта Проширеног генеричког мета-модела у концепт РСУБП мета-модела

Завршни корак у ланцу *M2M* трансформација представља трансформацију модела који је настао као резултат претходне трансформације у модел који је у складу са РСУБП (Релациони систем за управљање базама података) мета-моделом. РСУБП мета-модел обухвата појмове неопходне за моделовање шема база података на најнижем нивоу апстракције, односно на нивоу РСУБП-а. Овај мета-модел је дефинисан у [1]. Потреба за овим мапирањем произилази из чињенице да је РСУБП мета-модел концептуално и структурно ближи изабраној *SQL* платформи, чиме се омогућава једноставнија и ефикаснија реализација *M2T* трансформација у процесу генерисања извршивог *SQL* кода.

4. ГЕНЕРИСАЊЕ ИМПЛЕМЕНТАЦИОНОГ ОПИСА ШЕМЕ БАЗЕ ПОДАТАКА

Завршни корак поступка инжењеринга представља процес генерисања имплементационог описа подшеме базе података, који се спроводи применом *M2T* трансформација. Циљ овог корака је да се модел, који је у складу са РСУБП мета-моделом, трансформише у извршиви *SQL* код погодан за креирање релационе базе података у оквиру изабраног РСУБП-а. За реализацију генератора кода коришћен је *Eclipse Modeling Framework (EMF)* у комбинацији са *Acceleo* језиком, који представља уобичајено решење за *M2T* трансформације. Развијени генератор омогућава аутоматско генерисање имплементационог описа шеме базе података у облику *DDL (Data Definition Language)* скрипти за три платформе: *ANSI SQL*, *Oracle 11c* и *Oracle 23c*.

Генератор подржава генерисање процедуралних механизма за проверу очувања ограничења инверзних референцијалних интегритета. Кориснику је омогућен избор начина реализације ових ограничења — путем складиштених процедура (*stored procedures*) или погледа (*views*). Додатно, у оквиру овог решења подржао је и аутоматско генерисање процедуралних механизма који служе за проверу и одржавање проширених ограничења торке.

5. ЗАКЉУЧАК

Спроведеним истраживањем и реализацијом ланца *M2M* и *M2T* трансформација, у овом раду омогућен је развој *IS* применом *MDE* приступа. На овај начин, концептуални модели дефинисани језиком *FTDSL* постају полазна основа за аутоматско генерисање релационе шеме базе података и имплементационог описа. Кључни доприноси рада огледају се у:

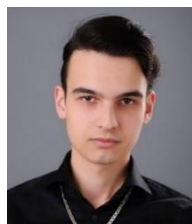
- развоју унапређене верзије алата *IIS*Case* у којој је реализован *MDE* процес инжењеринга *IS*,
- омогућавању интегрисаног концептуалног моделовања и генерисања релационе шеме базе података унутар јединственог *MDE* окружења,
- дефинисању и имплементацији ланца *M2M* трансформација који обезбеђује потпуно аутоматизован прелазак са концептуалног на имплементациони ниво.

Резултати овог истраживања представљају значајан корак ка успостављању потпуно интегрисаног *MDE* процеса алата *IIS*Case*, чиме се остварују могућности за даља истраживања, међу којима су генерисање прототипа *IS* заснованих на генерисаној шеми базе података.

6. ЛИТЕРАТУРА

- [1] Aleksić S, Metode transformacija šema baza podataka u obezbeđenju reinženjeringa informacionih sistema, doktorska disertacija, Univerzitet u Novom Sadu (2013).
- [2] Čeliković M, Pristup modelovanju specifikacija informacionog sistema putem namenskih jezika, doktorska disertacija, Univerzitet u Novom Sadu (2018).
- [3] Ristić S, Istraživanje problema konsolidacije podšema baza podataka, doktorska disertacija, Univerzitet u Novom Sadu (2003).
- [4] Luković I, Automatizovano generisanje podšeme relacione baze podataka putem formi, magistarski rad, Univerzitet u Beogradu (1993).
- [5] Mogin, Pavle, Ivan Luković, and Miro Govedarica. Principi projektovanja baza podataka. Fakultet tehničkih nauka (2004).

Кратка биографија:



Вељко Бубњевић рођен је у Новом Саду 2001. год. Дипломски рад на Факултету техничких наука из области Електротехнике и рачунарства одбранио је 2024. год. Године 2025. изабран је у звање сарадника у настави на Факултету техничких наука.
Контакт: bubnjevic@uns.ac.rs