

Jedan pristup za vizuelizaciju CWM-bazirane šeme relacione baze podataka

Igor Tomić
Tehnička škola
Gradiška, BiH
tomicigor@hotmail.com

Dražan Brđanin, Slavko Marić
Elektrotehnički fakultet
Banja Luka, BiH
bdrazen@etfbl.net, ms@etfbl.net

Sadržaj—U ovom radu prikazan je jedan pristup za vizuelizaciju šeme relacione baze podataka, zasnovan na ekstrakciji elemenata šeme predstavljene u skladu sa CWM specifikacijom i automatskom generisanju odgovarajućeg UML dijagrama klasa koji reprezentuje odnosnu šemu. Automatski generator ciljnog dijagrama klasa implementiran je kao Eclipse-Topcased plugin pod nazivom VisualCWM. Implementirani generator prvo generiše XML stablo na osnovu CWM modela, a zatim iz XML stabla ekstrahuje karakteristične koncepte i generiše odgovarajuće elemente ciljnog dijagrama klasa u skladu sa XMI specifikacijom. Nakon toga, vizuelizacija programski generisanog dijagrama klasa vrši se pomoću ugrađene Topcased funkcionalnosti. Pristup je ilustrovan na konkretnom primjeru.

Ključne riječi - CWM; šema relacione baze podataka; UML; dijagram klasa; Eclipse; Topcased; plugin

I. UVOD

CWM (eng. *Common Warehouse Metamodel*) [1] je OMG (*Object Management Group*) specifikacija za modelovanje i razmjenu metapodataka o relacionim, nerelacionim, višedimenzionalnim i drugim objektima u skladištima podataka. CWM predstavlja jedinstven okvir za reprezentaciju metapodataka te procesa i operacija za manipulaciju metapodacima. Jedinstvenim interfejsima za pristup i upravljanje metapodacima CWM specifikacija omogućava laku razmjenu metapodataka između različitih platformi i alata za skladištenje podataka te samih skladišta podataka u distribuiranim heterogenim okruženjima. Trenutna verzija CWM specifikacije 1.1.

Predmet ovog rada je vizuelizacija šeme relacione baze podataka koja je predstavljena u skladu sa CWM specifikacijom, jer postojeći CASE alati imaju veoma skromne

ili uopšte nemaju mogućnost vizuelizacije, kao što je to slučaj sa Eclipse-Topcased razvojnim okruženjem.

Rad je organizovan na sljedeći način. Nakon uvodnog dijela, u drugom dijelu navedene su osnovne karakteristike CWM-bazirane specifikacije šeme relacione baze podataka. U trećem dijelu opisane su mogućnosti vizuelizacije šeme relacione baze podataka UML dijagramom klasa. Princip rada automatskog generatora dijagrama klasa opisan je u četvrtom dijelu, a u petom dijelu dat je primjer vizuelizacije jedne CWM-bazirane šeme relacione baze podataka. Na kraju su dati zaključci i pravci za nastavak istraživanja.

II. CWM-BAZIRANA ŠEMA RELACIONE BAZE PODATAKA

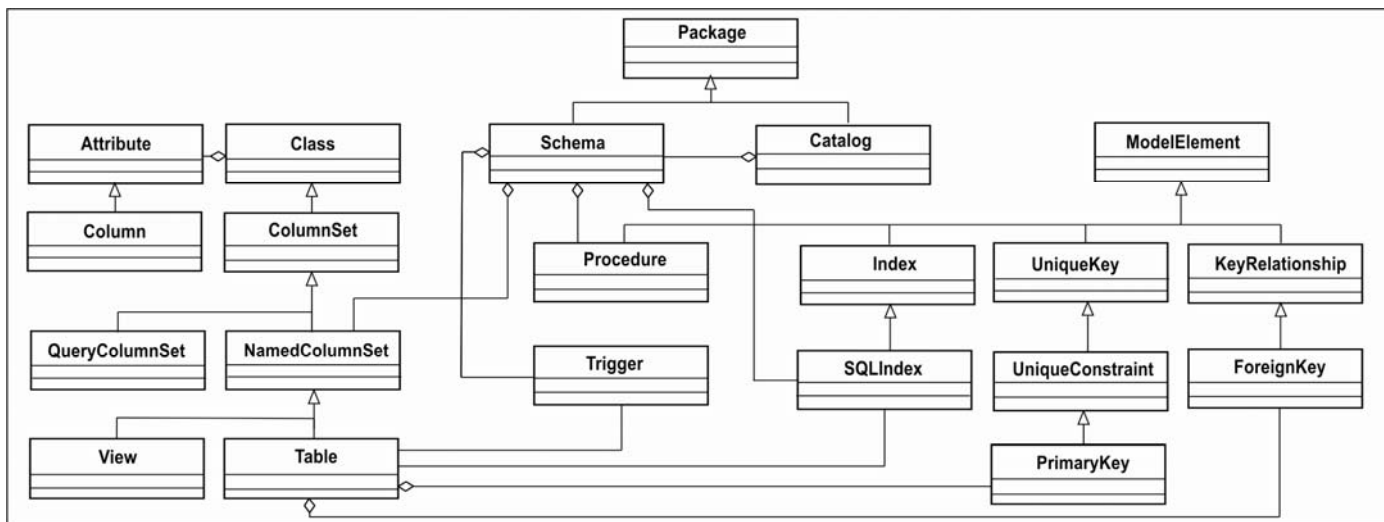
CWM specifikacija je bazirana na tri ključna industrijska standarda u domenu (meta)modelovanja: UML (*Unified Modeling Language*) za modelovanje [2], [3]; MOF (*Meta Object Facility*) za metamodelovanje [4]; i XMI (*XML Metadata Interchange*) za razmjenu metapodataka [5].

CWM kao kompletan metamodel za reprezentovanje koncepata iz oblasti baza podataka i poslovne analize, omogućava i sintaksu i semantiku potrebnu pri konstruisanju metapodataka koji opisuju sve komponente ISC-a (eng. *Information Supply Chain*) [6]. Sastoji se od nekoliko podmetamodela koji su usko povezani i od kojih svaki predstavlja određeni poddomen iz ISC okruženja. Osnovu CWM metamodela čini pet metamodela: *Object Model*, *Foundation*, *Resource*, *Analysis* i *Management*.

Metamodeli su hijerarhijski organizovani u vidu istoimenih paketa, koji kontrolišu složenost, promovišu razumijevanje i omogućavaju ponovo korištenje. Svaki metamodel ima podmetamodele, kao što je prikazano na sl. 1.

Management	Warehouse Proces			Warehouse Operation		
Analysis	Transformation	OLAP	Data Mining	Information Visualization	Buisness Nomenclature	
Resource	Object	Relational	Record	MultiDimensional	XML	
Foundation	Business Information	Data Types	Expressions	Key and Indexes	Software Deployment	Type Mapping
Object Model	Core		Behavioral	Relationships	Instance	

Slika 1. Struktura CWM metamodela [1]



Slika 2. Struktura relacionog paketa [1]

Relacioni paket služi za reprezentaciju šema relacionih baza podataka. Ovaj paket je usaglašen sa SQL99 standardom [7], uključujući i njegove objektno-orijentisane ekstenzije. Relacioni paket podržava razmjenu podataka između baza podataka većine proizvođača sistema za upravljanje relacionim bazama podataka. Iako je dovoljan da podrži razmjenu relacionih podataka, relacioni paket ipak ne omogućava uvijek razmjenu kompletnih opisa šema relacionih baza podataka, specifikovanih u različitim komercijalnim implementacijama. Razlog za to je što relacioni metamodel podržava samo logičke aspekte (tabele, pogledi, okidači, ...), ali ne i fizičke aspekte (npr. lokacije datoteka). Trenutno se radi na kreiranju proširenog paketa koji će podržavati i specifičnosti fizičke šeme.

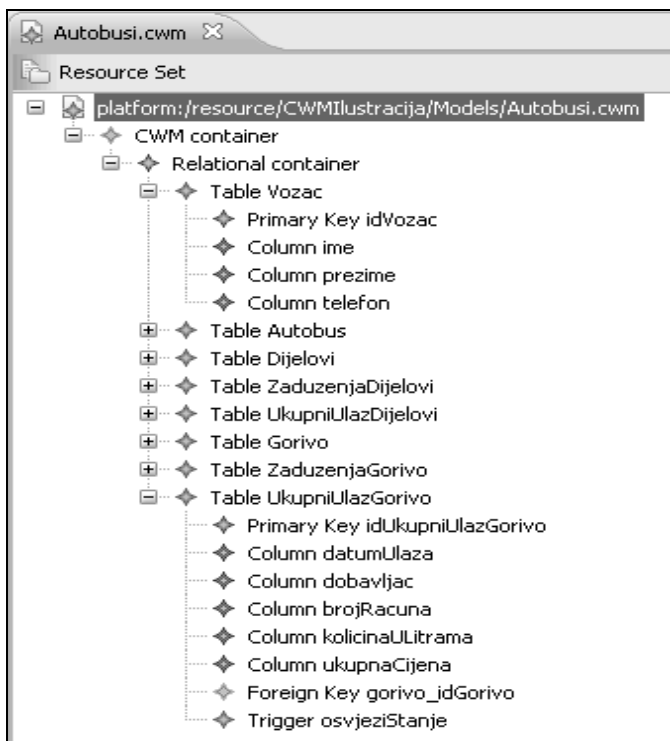
Relacioni paket, kao najveći paket u CWM-u, sadrži 24 vlastite klase i zahtijeva još 44 klase iz drugih paketa, što predstavlja približno jednu trećinu ukupnog broja klasa koje sadrži CWM (68/208). Struktura relacionog paketa prikazana je na sl. 2. Paketi od kojih zavisi relacioni paket su: Behavioral, Core, Instance, DataTypes i Keys and Indexes.

Implementacija CWM specifikacije u Eclipse-Topcased okruženju [8] omogućava reprezentaciju CWM bazirane šeme relacione baze podataka u vidu XML stabla čiji je root element Relational container. Unutar ovog elementa reprezentuju se tabele tagom Table, a dalje se, kao child elementi tabela, reprezentuju kolone (Column), primarni ključ (Primary Key), strani ključevi (Foreign Key), okidači (Triger), itd. Primjer reprezentacije CWM bazirane šeme relacione baze podataka prikazan je na sl. 3.

III. UML REPREZENTACIJA ŠEME RELACIONE BAZE PODATAKA

Dijagram klasa je strukturni UML dijagram koji opisuje statičku strukturu sistema prikazujući klase i njihove međusobne veze, i predstavlja jednu od najčešće korišćenih notacija u O-O softverskom inženjerstvu. S obzirom na intenzivan razvoj i primjenu UML-a u svim fazama životnog ciklusa softverskih sistema, dijagram klasa se često koristi i u projektovanju baza podataka, iako standardna notacija nije baš u potpunosti prilagođena za tu namjenu.

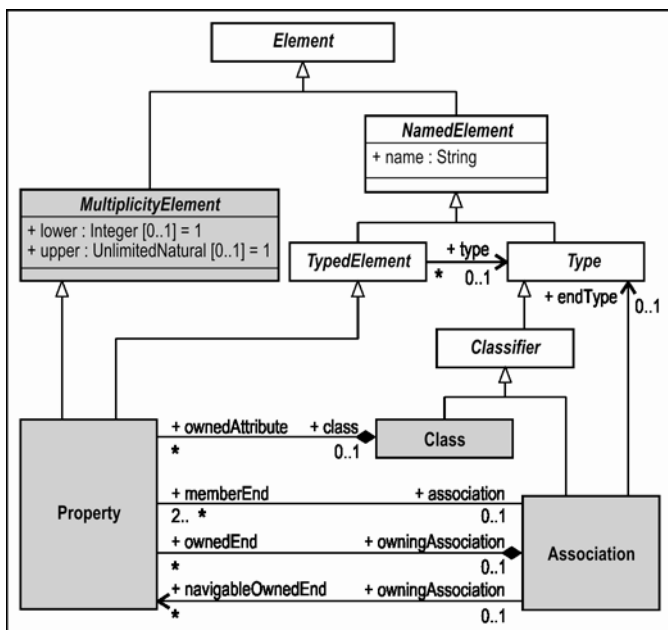
Za razvoj konceptualnog modela, koji prikazuje tipove entiteta (objekata) i njihove međusobne veze, moguće je koristiti standardni UML dijagram klasa. Međutim, šema relacione baze podataka je specifičan model koji ne može da se reprezentuje standardnom notacijom [9]. Zato se u praksi koristi specijalizovana notacija za modelovanje karakterističnih koncepata (tabele, ključevi, ograničenja, itd.). Mehanizmi za proširenje, odnosno specijalizaciju notacije su: stereotip (eng. *stereotype*), označena vrijednost (eng. *tagged value*) i ograničenje (eng. *constraint*). Skup proširenja standardne UML notacije za primjenu u nekom domenu naziva se profil. Najčešći korišteni mehanizam za proširenje UML notacije je stereotip. Stereotip omogućava da se, na osnovu nekog



Slika 3. Primjer CWM bazirane šeme relacione baze podataka

postojećeg koncepta kreira novi koji ima drugačije osobine. Na primjer [9], metaklasa *Class* sa stereotipom <<Table>> se tipično koristi za modelovanje tabele, metaklasa *Property* sa stereotipom <<PK>> za modelovanje primarnog ključa, a metaklasa *Dependency* sa stereotipom <<FK>> za modelovanje stranog ključa, odnosno ograničenja po referencijalnom integritetu između dvije tabele, itd.

U trenutnoj fazi implementacije, a u nedostatku adekvatnog UML profila za modelovanje šeme relacione baze podataka u Eclipse-Topcased okruženju, za grafičku reprezentaciju šeme koristi se standardna UML notacija. Na sl. 4 prikazan je dio UML metamodela [2] koji se trenutno koristi za reprezentaciju šeme. Tabela se reprezentuje klasom, a kolone korespondentnim atributima klase. Ograničenje po referencijalnom integritetu (migracija ključeva) reprezentuje se usmjerenom asocijacijom između odgovarajućih klasa, pri čemu naziv asocijacije odgovara nazivu primarnog ključa koji migrira. Kardinalnost mapiranja je '1:*', gdje je multiplikativnost '1' na strani klase (tabele) čiji primarni ključ migrira.



Slika 4. Dio UML metamodela [2] za reprezentaciju šeme relacione baze podataka

IV. GENERATOR DIJAGRAMA KLASA

Automatski generator dijagrama klasa, koji omogućava vizuelizaciju CWM-bazirane šeme relacione baze podataka, implementiran je kao Eclipse-Topcased *plugin* pod nazivom **VisualCWM**.

Funkcionalnost Topcased platforme [10] i implementiranog generatora zasnovana je na EMF-baziranoj [11] implementaciji UML 2.1 specifikacije, koja omogućava vizuelno UML modelovanje, programsko generisanje dijagrama i automatsku vizuelizaciju programski generisanih UML dijagrama. Svaki UML dijagram u Topcased modelu reprezentuje se sa dva istoimena fajla različitih ekstenzija. Fajl sa ekstenzijom `.uml` sadrži XMI reprezentaciju dijagrama, a fajl sa `.umldi` ekstenzijom opisuju njegovu vizuelizaciju.

Implementacija se temelji na kombinaciji DOM XML parsera za analizu ulazne CWM šeme i primjeni UML2 *factory* [12] za generisanje ciljnog dijagrama. Generator na osnovu ulaznog `.cwm` fajla generiše odgovarajuće XML stablo i ekstrahuje karakteristične koncepte ulazne CWM šeme, a zatim generiše korespondentne koncepte ciljnog dijagrama klasa u skladu sa XMI specifikacijom i kreira ciljni `.uml` fajl, kao što je prezentovano u [13]. Automatska vizuelizacija programski generisanog dijagrama (kreiranje `.umldi` fajla) potom se vrši pomoću ugrađene Topcased funkcionalnosti. Pojednostavljeni algoritam, koji je prikazan na sl. 5, opisuje postupak generisanja ciljnog dijagrama klasa na osnovu CWM bazirane šeme relacione baze podataka.

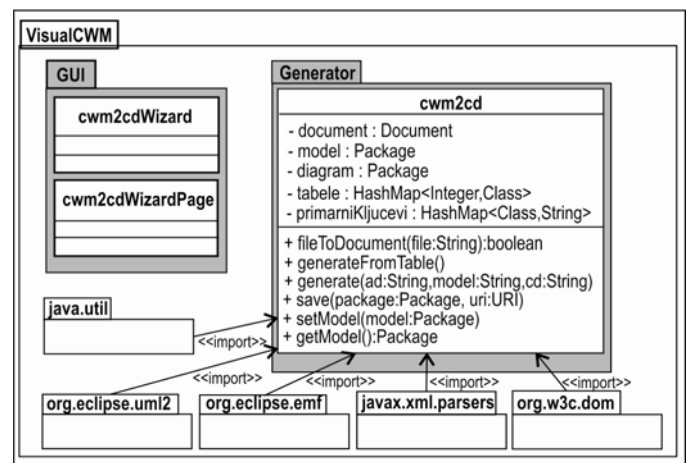
```

get all tables;
for each table
{
  get table_name;
  create class;
  get all table_elements;
  for each table_element
  {
    if (table_element == primaryKey)
    {
      get primaryKey_name;
      create class attribute;
    }
    else if (table_element == foreignKey)
    {
      get foreignKey_name;
      create class attribute;
      get foreignKey_AssociatedClass;
      create association between classes;
    }
    else if (table_element == column)
    {
      get column_name;
      create class attribute;
    }
  }
}

```

Slika 5. Algoritam generisanja dijagrama klasa

Arhitektura VisualCWM *plugin*-a prikazana je na sl. 6. GUI komponenta obezbeđuje integraciju *plugin*-a u Topcased platformu i selekciju polazne CWM-bazirane šeme. *Generator* komponenta, čiju osnovu čini `cwm2cd` klasa, implementira generator u skladu sa datim funkcionalnim zahtjevima.



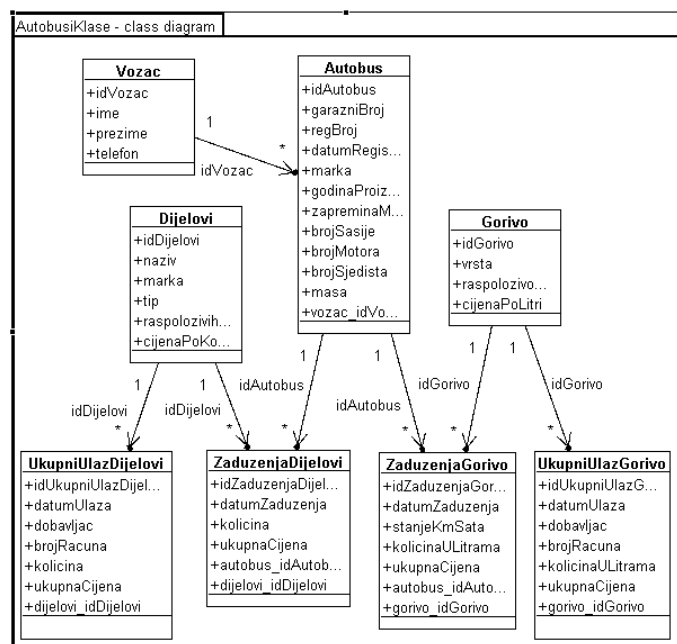
Slika 6. Arhitektura VisualCWM plugin-a

V. PRIMJER VIZUELIZACIJE

Primjena generatora ilustrovana je na konkretnom primjeru. Polazni osnov čini relacionala baza podataka čija je CWM-bazirana šema *Autobusi.cwm* data na sl. 3.

Rezultat primjene generatora na ulaznu CWM šemu je automatski generisana XMI reprezentacija (.uml fajl) ciljnog dijagrama klasa, čiji je rezultat vizuelizacije prikazan na sl. 7.

Prikazani dijagram klasa pokazuje da je implementirani generator automatski generisao sve očekivane klase sa svim pripadajućim atributima, koje reprezentuju odnosne tabele i njihove kolone u CWM šemi, kao i sve usmjerene asocijacije koje reprezentuju veze između tabela, odnosno ograničenja referencijalnog integriteta.



Slika 7. Rezultat vizuelizacije programski generisanog dijagrama klasa koji reprezentuje CWM baziranu šemu relacione baze podataka sa sl. 3

VI. ZAKLJUČAK

U ovom radu prikazan je jedan pristup za vizuelizaciju CWM-bazirane specifikacije šeme relacione baze podataka u Eclipse-Topcased razvojnom okruženju. Prezentovan je generator koji vrši ekstrakciju karakterističnih elemenata ulazne CWM šeme i automatski generiše korespondentne koncepte u UML dijagramu klasa kojim se vizuelizuje polazna šema. Generator je implementiran kao Eclipse-Topcased *plugin* pod nazivom VisualCWM.

U trenutnoj fazi implementacije, a u nedostatku adekvatnog UML profila za reprezentaciju šeme relacione baze podataka, implementirani generator koristi standardnu UML notaciju.

S obzirom na nedostatke standardne notacije, u nastavku istraživanja planirano je definisanje odgovarajućeg UML profila za reprezentaciju šeme relacione baze podataka i prilagođenje generatora.

LITERATURA

- [1] Object Management Group (OMG), Common Warehouse Metamodel Specification (CWM) v1.1. OMG, 2003.
- [2] Object Management Group (OMG), Unified Modeling Language (UML): Infrastructure, v2.3. OMG, 2010.
- [3] Object Management Group (OMG), Unified Modeling Language (UML): Superstructure, v2.3. OMG, 2010.
- [4] Object Management Group (OMG), Meta Object Facility (MOF), Core Specification v2.0. OMG, 2006.
- [5] Object Management Group (OMG), MOF2.0/XMI mapping, v2.1.1. OMG, 2007.
- [6] J. Poole, D. Chang, D. Tolbert, and D. Mellor, Common Warehouse Metamodel. New York: John Wiley & Sons, Inc, 2002.
- [7] ISO/IEC 9075-2:1999, Database Languages, SQL, Part 2: Foundation (SQL/Foundation). ISO, 1999.
- [8] Common Warehouse Metamodel in EMF, <http://www.dinkla.net/eclipse/cwm.html>.
- [9] S. Marić, i D. Brđanin, Relacione baze podataka. Banja Luka: Elektrotehnički fakultet, 2012.
- [10] Topcased Quick start editor tutorial, <http://www.topcased.org/documents>.
- [11] F. Budinsky, D. Steinberg, E. Merks, R. Ellersick, and T. Grose, Eclipse Modeling Framework. Boston: Pearson Education, 2003.
- [12] K. Hussey, Getting Started with UML2. New York: IBM Corp, 2006.
- [13] D. Brđjanin, S. Maric, and D. Gunjic, "ADBdesign: An approach to automated initial conceptual database design based on business activity diagrams", in: ADBIS 2010, LNCS, vol. 6295, B. Catania, M. Ivanović, and B. Thalheim, Eds. Berlin Heidelberg: Springer-Verlag, 2010, pp. 117-131.

ABSTRACT

This paper proposes an approach to visualization of relational database scheme, which is based on the extraction of the characteristic elements of corresponding CWM-based scheme specification and the automated generation of the UML class diagram representing the source scheme. An automatic generator of the target class diagram is implemented as an Eclipse-Topcased plugin named VisualCWM. Firstly, the implemented generator generates the appropriate XML tree based on the CWM model of the source scheme, and then extracts the characteristic concepts from the generated XML tree and generates the corresponding elements in the target class diagram according to XMI specification. Visualization of the generated class diagram is then performed by the integrated Topcased functionality. The proposed approach is illustrated by a concrete example.

AN APPROACH TO VISUALIZATION OF CWM-BASED RELATIONAL DATABASE SCHEME

Igor Tomic, Drazen Brđjanin, Slavko Maric