STUDIA INFORMATICA 2010
Volume 31 Number 2A (89)

Katarzyna TUBIELEWICZ
InsERT S.A.
Lech TUZINKIEWICZ
Politechnika Wrocławska, Instytut Informatyki

DATABASE REVERSE ENGINEERING METHOD

Summary. In this paper the method of reverse engineering for database is presented. The proposed method takes into account relational schemas which are at least in the 1stNF. In this method is used ontology as a model of the considered domain. It allows to interpret the physical data models according to the ontology and reconstruct conceptual data models.

Keywords: database, database reverse transformation, ontology

METODA ODWROTNEJ TRANSFORMACJI BAZ DANYCH

Streszczenie. W artykule przedstawiono koncepcję odwrotnej transformacji relacyjnych baz danych do poziomu modelu konceptualnego. W proponowanej metodzie wykorzystuje się ontologie reprezentujące dziedziny, z których dane gromadzone są w analizowanych bazach danych.

Słowa kluczowe: baza danych, odwrotna transformacja modeli danych, ontologia

1. Introduction

Nowadays it is important to handle a great amount of data. The success of an organization depends on obtaining precise information about its processes, to effectively manage this data and use it. This data may be used to analyze and guide organization`s activities [18]. In many fields of life, our actions are connected with acquiring data which is stored in a database.

In order to create a database, there is a need to understand particular domain, possess the ability of modelling this domain, and acquire and gather data in accordance with existing standards. In database design process data models at different level of abstraction are created,

starting with conceptual data model and ending with physical representation of a database. Data and rules in particular domain can be changed. As a result there is a need to modify database and documentation in order to apply these changes. Modifications of data models can be applied to different levels of abstraction. In many cases modifications are applied to physical data model, what is reprehensible, because in this situation the modifications should be traced to the higher levels of abstraction in order to keep them as actual. A special case is a legacy database, which is an existing database with some data, for which there is not any documentation, yet there is one needed.

Obtaining conceptual model from physical representation of database can be performed using database reverse transformation process. To understand a term of reverse transformation there is a need to understand the area of reverse engineering. The reverse engineering is a process of analyzing a system in order to identify its components and the relations between them, and to create representation of this system in a different form or at the higher level of abstraction [9]. It corresponds to three principal aspects of a system: data, processes and control [8]. A data reverse engineering concentrates on the database that is in the organization. "It is a collection of methods and tools to help an organization determine the structure, function, and meaning of its data" [9].

Database Reverse Engineering (DBRE) handles the tasks of understanding legacy databases and extracting their design specifications [7]. In this process the conceptual model is derived from physical one. It can be done by transforming physical model to logical model and then to conceptual model – Fig. 1. The logical model can be omitted.

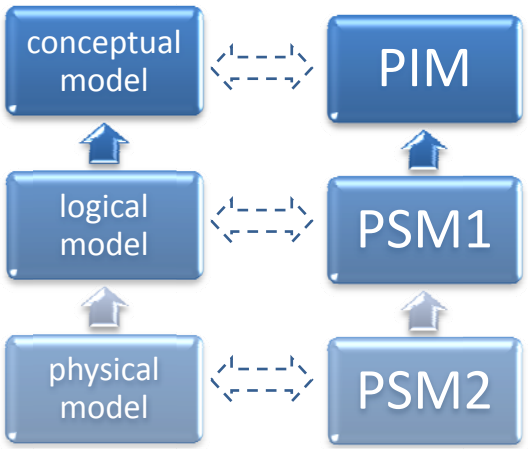


Fig. 1. Database reverse transformation in MDA approach

Rys. 1. Odwrotna transformacja baz danych (MDA)

The problem of obtaining conceptual data model from relational database requires identification of entities based on structure and data in database. In this process one can use ontology which represents considered domain. While performing an analysis of data in the database two cases can be considered. In the first case the problem domain is known. In this case it is needed to find appropriate ontology. This enables to eliminate entities which are not in scope of considered domain.

In this paper database reverse engineering method is presented. The method requires definition of a set of ontologies and databases at least in the first normal form (1stNF).

1.1. Related work

Over the time, many database reverse engineering methods were proposed. First methods considered conventional files (e.g. COBOL) were created, for instance in [5]. Next generation of DBRE techniques have taken under consideration network and hierarchical databases (e.g. in [17]). Other techniques have translated a relational database into an Entity-Relationship (ER)[1] model, as well as extensions – EER, ERC+ (e.g. [1]), or conceptual Model. In all of those methods there have been made some assumptions (in some – many strong ones, in others – weaker and as few as possible). The most popular assumptions are: known functional and/or inclusion dependencies (e.g. [17]), some kind of attribute name coherency (e.g. [6]), normal form of relational schema (e.g. [13]), known candidate keys (e.g. [6], [15]), interaction with user (e.g. [6]).

In later works, assumptions are weakened. New sources of information have been explored, e.g. the user queries in [4]. Relations in second normal form (2^{nd}NF) are considered in [18]. Relational schemas are transformed into another constructs, like Object-Oriented Schemas in [10].

There was even a framework proposed for the design and evaluation of reverse engineering methods for relational databases by Chiang et al. in [7]. The new approach to Database Reverse Engineering has recently proposed in which relational databases are transformed into ontology [2,14].

2. Ontology

The ontology is defined as an explicit specification of conceptualization ([2,11]). It consist of a set of concepts, a set of properties, an inheritance hierarchy of concepts, a set of axioms and a set of simple data types. Property can be understood like an attribute of concept as well as a relationship between concepts. Each property belongs to one concept (concept is a domain of property) and its range is a concept (if it represents a relationship) or simple data type (if it represents an attribute) [2]. An axiom concerns particular property. In this work ontology covers only static elements of a domain.

2.1. Ontology repository

In order to be able to use domain knowledge gathered in ontology, there is a need to have ontology repository. According to [12], there are two techniques for storing ontology: storing ontology in a flat file using file system or in database using database management system.

[1] Information about an Entity-Relationship Programming Languages can be found in [16].

Advantages and disadvantages of these kinds of storing are discussed in [3]. In this work second approach was chosen, due to capability of performing query executions and easiness of managing of repository content. In the database there is defined an ontological model [3]. The ontological model used in this work consists of concepts, properties, data types, constraints and inheritance as well as synonyms. In the Fig. 2 there is presented class diagram of the ontological model.

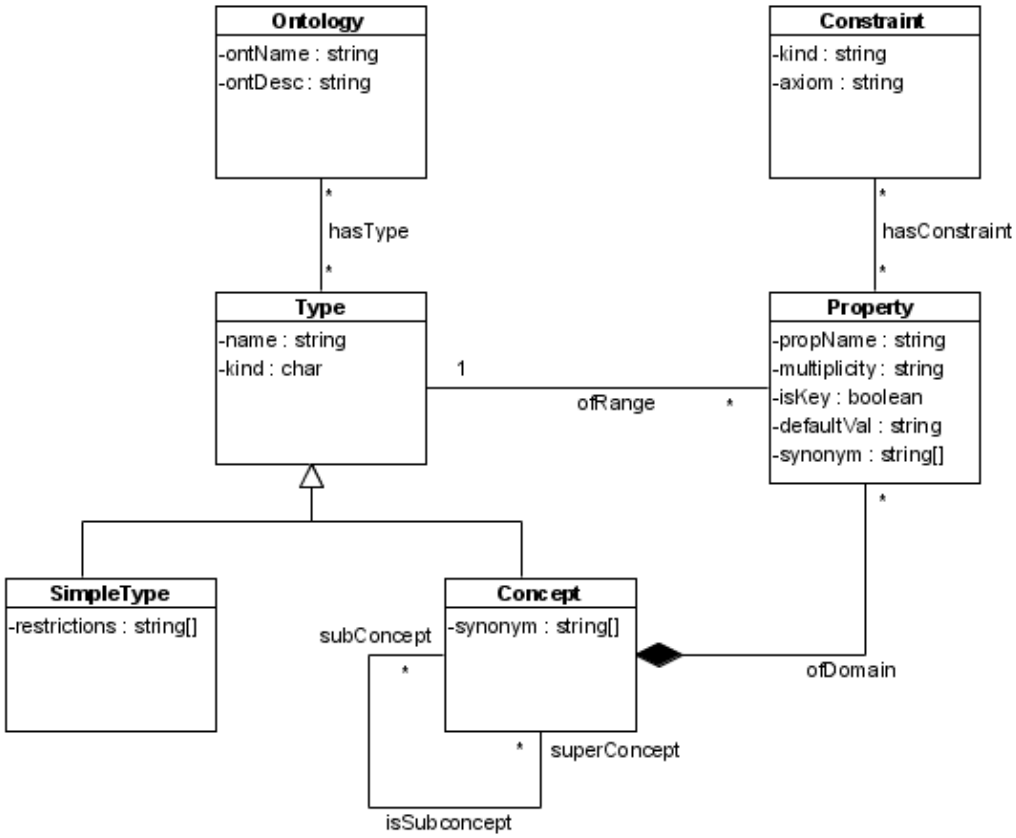


Fig. 2. The representation of the ontological model
Rys. 2. Model konceptualny modelu ontologicznego

3. Ontology-based Reverse Engineering Method

Ontology-based Reverse Engineering Method (OREMoD) of Database is a method of database reverse transformation. It allows to extract conceptual data model form relational databases. The discussed method is dedicated for relational schemas which are at least in 1stNF. It does not require any modifications of database schema. On the database there could have been applied some optimization techniques (for instance denormalization or partitioning). These techniques are taken into account in this method.

3.1. Description of the OREMoD method

The transformation process of the OREMoD method can be divided into four main phases, as it is illustrated in the Fig. 3. First initialization is conducted, which includes selection of a database and definition of compliance function parameters. The second phase determines the level of database compliance with selected ontology (or all ontologies from ontology repository). Subsequently, the analysis of database in accordance to ontology is performed. Finally, relational database is transformed to conceptual data model.

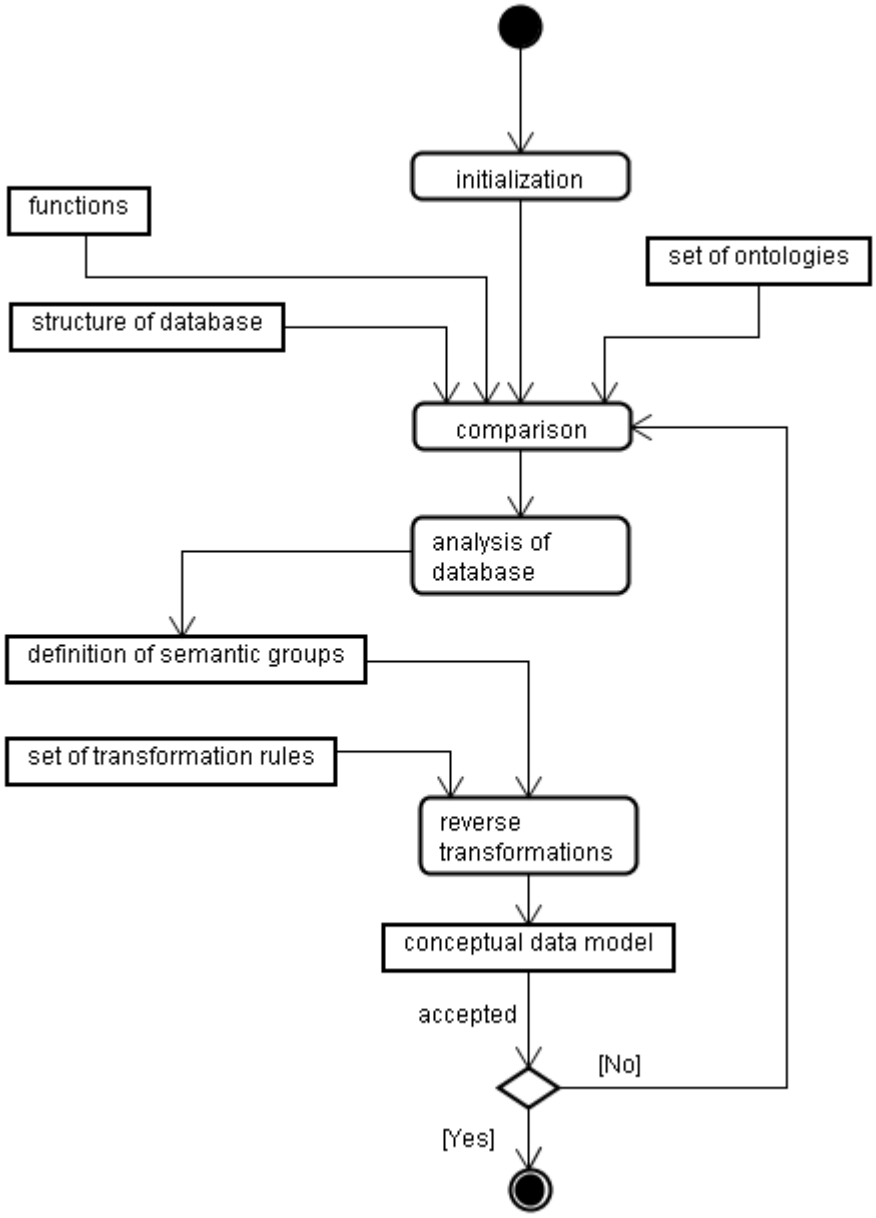


Fig. 3. Activity diagram of the transformation process
Rys. 3. Diagram aktywności procesu transformacji

3.2. Usage of ontologies

As it was mentioned before, in order to identify existing entities within a database, ontology is used as a domain model. In the first case, when the domain of data is unknown, there is a need to find compliant ontology. On the other hand, when the domain of data is known, there can be made a comparison with the declared ontology. This enables to eliminate the effects of inappropriately designed database (e.g. data redundancy), eliminate entities which are not in scope of considered domain and specify entities (which are incomplete) from considered domain.

To evaluate the database compliance with the ontology the following function is defined:

Comp: (DB, O) → ℝ, where

ℝ is a degree of compliance and its value is from a range of real numbers from 0 to 1 inclusive, DB denotes a database, O denotes ontology. The compliance function is parameterized by weights which denote influence of name of a column, type of a column, column constraints, name and schema of a table on final value of database compliance with the ontology.

The minimum value on the acceptance level is chosen as 0,5 (default value). The proposed ontology should be treated only as a suggestion and the final decision is taken by the user.

3.3. Analysis

In the process of analysis each of database elements is qualified to one of fourteen semantic groups, among other: simpleClass, associationClass, and aggregation. Each of the group represents specific elements at the level of conceptual data models, e.g. classes and relation between them.

There is defined the set of analysis rules, which allow to classify database objects into semantic groups. The set includes 52 rules. If there is not any possibility of unambiguous qualification of database objects to semantic group then the experienced user should decide to which group an object belongs. The user`s decisions are final. After analysis, each of database objects must belong to one of the semantic group.

An example of an ambiguous interpretation of database schema is presented in Fig 4. The considered example concerns the table in which rows are identified by artificial key and there are defined three referential constraints (to tables: Students, Courses and Employees). In this case this table can be represented at conceptual data level as an association class with {bag} constraint or a class and associations with remaining three classes. In the Fig. 5, there are presented possible mappings of TakesExams table into elements of conceptual data model.

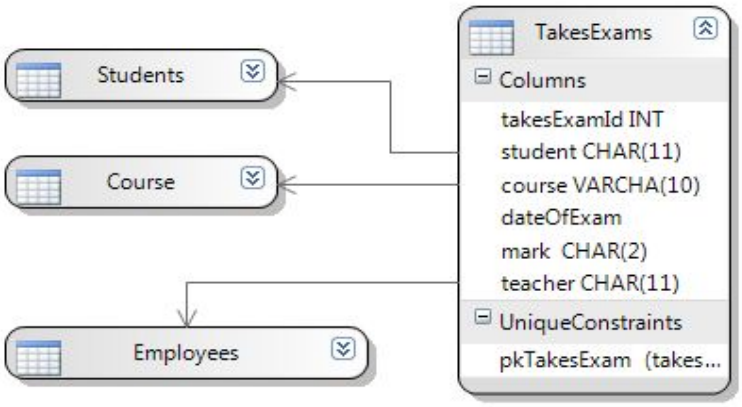
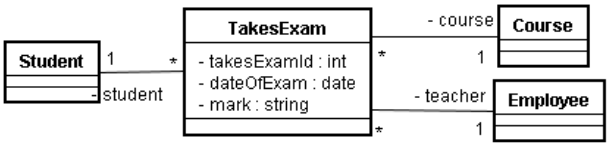
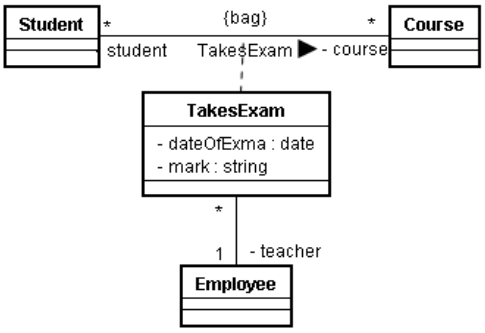


Fig. 4. Example of TakesExams table for which the problem of unambiguous mapping exists
Rys. 4. Przykład tabeli, dla której występuje problem niejednoznacznego odwzorowania

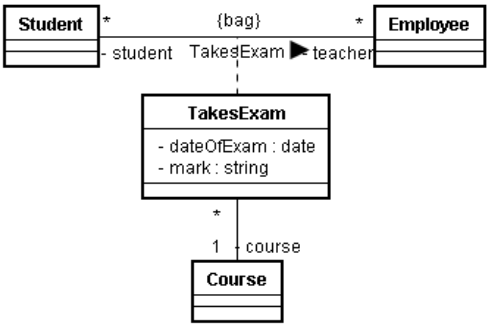
In this case the final decision must be made by the user.



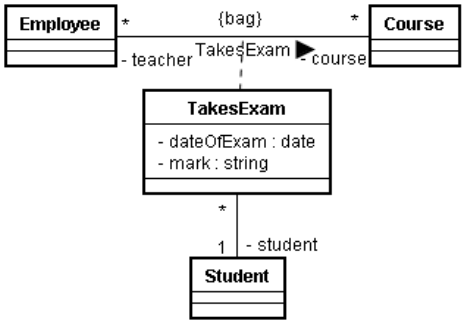
Variant 1. Class with three associations.



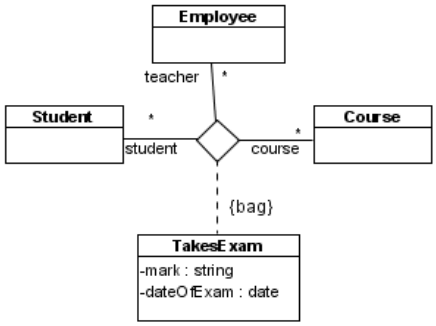
Variant 2. Association class between Student and Course classes.



Variant 3. Association class between Student and Employee classes.



Variant 4. Association class between Course and Course classes.



Variant 5. Association class between Student, Course and Employee classes.

Fig. 5. Examples of the different reverse transformations of database from Fig. 4
Rys. 5. Przykłady różnych odwrotnych transformacji bazy danych z rys. 4

3.4. Reverse transformation

In the method there is defined the set of transformation rules which determine how an element of semantic group is transformed into an element (or several elements) of a conceptual data model. The set includes 26 transformation rules.

Taking under consideration possible results of the reverse transformations proposed by the system (the examples in Fig. 5), the final decision must be made by the user. One of the acceptable solution from the set of possible variants is presented in Fig. 6.

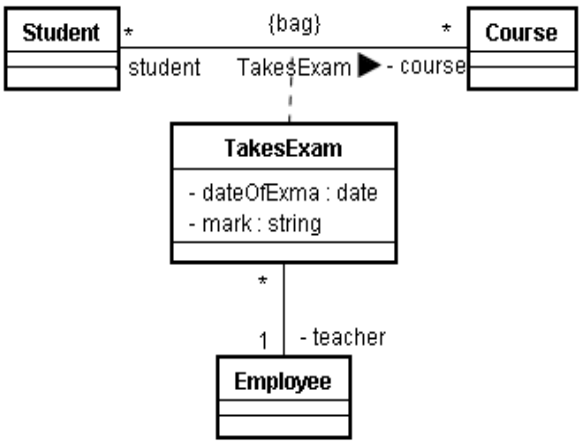


Fig. 6. The result of reverse transformation of the table TakesExams table which involves decision of the user

Rys. 6. Rezultat odwrotnej transformacji tabeli TakesExams z uwzględnieniem decyzji użytkownika

4. Experiment

The OREMoD method has been implemented in Microsoft Visual Studio .NET (C#), whereas the definitions of exemplary ontologies and analyzed databases have been implemented in MS SQL 2008.

Table 1

Values of compliance function for considered databeses and ontologies

Database name	Ontology name	Value of compliance
DrivingSchool1	Driving School	***0,8638***
DrivingSchool2	Driving School	***0,7857***
University1	Driving School	***0,2005***
University1	University	***0,8238***
University2	University	***0,7250***
DrivingSchool1	University	***0,1210***

Many experiments have been conducted which aim was assessment of usability of the proposed method. In one of the test there are taken into account two database schemas for the driving school (DrivingSchool1, DrivingSchool2) and two different database schemas for the university (University1, University2). For each domain has been created one ontology. In both cases the definition of database schemas were different. The schemas with identifier 1 were normalized and had natural primary key. In the second case schemas were denormalized and had artificial key.

The parameters of compliance function and the ranges of membership classes were the same for all databases. The result of the test is presented in the Table 1.

When the compliance function is calculated for the databases and the appropriate ontology (the same domain) then the value of compliance is higher and acceptable (here more than 0,7). On the other hand, when database and ontology are created for different, but still similar domains, the function detects similarity (terms and constructs) at the level of about 0.2.

5. Conclusions and Future Work

In this paper the method of reverse engineering for database is presented. The proposed method takes into account relational schemas which are at least in 1stNF. In this method is used ontology as a model of the considered domain. It allows to interpret the physical data models according to the ontology and reconstruct conceptual data models. When the related domain is known, ontology can be used to assess the quality of data in context of this domain.

The evaluation of database compliance with the ontology depends on the defined compliance function. The experiment results allow to state that the compliance function is well defined and allow performing valuable assessment. The compliance function is parameterized and enables to change defined parameters based on subjective expectations of the users of implemented method.

The important part of the method is the set of rules which are required to conduct the analysis. The rules of analysis and transformation physical data models were empirically developed. They may be modified depending on the results of future experiments.

Assessment of conceptual data models which are results of reverse transformation is a separate issue to be solved in the future work. It will be needed to develop a quality model for evaluation of received conceptual data models. Furthermore, a good idea is to take under consideration other types of databases (e.g. object-relational and object databases).

BIBLIOGRAPHY

1. Andersson M.: Extracting an Entity Relationship Schema from a Relational Database through Reverse Engineering. Proceedings of the13th International Conference on the Entity-Relationship Approach, Lecture Notes In Computer Science; Vol. 881. Springer-Verlag, London, UK, 1994.
2. Astrova I.: Reverse Engineering of Relational Databases to Ontologies. The Semantic Web: Research and Applications. Springer, Berlin/ Heidelberg, 2004.

3. Astrova I., Korda N., Kalja A.: Storing OWL Ontologies in SQL Relational Databases. Internationa Journal of Electrical, Computer and Systems Engineering (Fall 2007), s. 242÷247. Online on www.waset.org/ijecse/v1/v1-4-37.pdf.
4. Barbar A.: A User Driven Method for Database Reverse Engineering. Proceedings of the conference on Advanced Information Systems Engineering, CAiSE'01. The 8th Doctoral Consortium (Interlaken June 4-8, 2001).
5. Casanova M.A., de Sa J.E.A.: Designing Entity-Relationship Schemas for Conventional Information Systems. Proceedings of the 3rd Conference on the ER Approach to Software Engineerimg (1983), s. 265÷277.
6. Chiang R.H.L.: A knowledge-based system for performing reverse engineering of relational databases. Decision Support Systems, 13, 3-4 (March 1995), s. 295÷312.
7. Chiang R.H.L., Barron T.M., Storey V.C.: A framework for the design and evaluation of reverse engineering methods for relational databases. Data & Knowledge Engineering, 21, 1 (December 1996), s. 57÷77.
8. Chikofsky E.: The Necessity of Data Reverse Engineering. In Preface to Data Reverse Engineering: Slaying the Legacy Dragon. McGraw-Hill, New York, 1996.
9. Chikofsky E.J., Cross J.H.: Reverse Engineering and Design Recovery: A Taxonomy (January 1990), s. 13÷18.
10. Fahrner Ch., Vossen G.: Transforming Relational Database Schemas into Object-Oriented Schemas according to ODMG-93. Lecture Notes In Computer Science; Vol. 1013, Proceedings of the Fourth International Conference on Deductive and Object-Oriented Databases. Springer-Verlag, London, UK, 1995.
11. Gruber T.R.: A Translation Approach to Portable Ontology Specification. Knowladge Acquisition, 5 (1993), s. 199÷220.
12. Harrison R., Chan C.: Distributed Ontology Management System. In Proceedings 18th Annual Canadian Conference on Electrical and Computer Engineering. (Saskatoon, Canada 1-4 May 2005), s. 661÷664.
13. Johannesson P.: A Method for Transforming Relational Schemas Into Conceptual Schemas. Proceedings of the Tenth International Conference on Data Engineering (Washington, USA 1994), IEEE Computer Society, s. 190÷201.
14. Lubyte L., Tessaris S.: Extracting Ontologies from Relational Databases. Free University of Bozen-Bolzano, Bolzano, Italy, 2007.
15. Markowitz V.M., Makowsky J.A.: Identifying Extended Entity-Relationship Object Structures in Relational Schemas. IEEE Transactions on Software Engineering, 16, 8 (August 1990), s. 777÷790.

16. Markowitz H.M., Malhotra A., Tsalalikhin Y., Pazel D.P., Burns L.M.: An Entity-Ralationship Programing Language. IEEE Transactions on Software Engineering, 15, 9 (September 1989), s. 1120÷1130.
17. Navathe S., Awong A.: Abstracting Relational and Hierarchical Data With a Semantic Data Model. In Proceedings of the 6th International Conference on the Entity Relationship Approach (Amsterdam, The Netherlands 1987), North-Holland Publishing Co., s. 305÷333.
18. Ramanathan S., Hodges J.: Extraction of Object-Oriented Structures from Existing Relational Databases. ACM SIGMOD Record, 26, 1 (March 1997), s. 59÷64.

Recenzent: Dr inż. Dariusz Rafał Augustyn

Wpłynęło do Redakcji 31 stycznia 2010 r.

Omówienie

Projektowanie bazy danych wymaga zrozumienia dziedziny przedmiotowej, umiejętności jej zamodelowania, pozyskania oraz gromadzenia danych zgodnie z istniejącymi standardami. Dane gromadzone w bazie danych mają wartość rynkową, gdy reprezentują fakty znanej dziedziny i jednocześnie są aktualne. Projektowanie baz danych polega na modelowaniu danych na różnych poziomach abstrakcji i obejmuje poziom konceptualny, logiczny, fizyczny. Efektem końcowym jest baza danych utworzona na podstawie fizycznego modelu danych.

Rozpatrywany wycinek rzeczywistości jak również oczekiwania użytkowników w stosunku do realizowanych procesów na podstawie danych gromadzonych w bazie danych mogą podlegać zmianom. Zmieniające się wymagania, zarówno funkcjonalne jak i niefunkcjonalne, często wymagają modyfikacji modeli danych, ale niestety, w wielu przypadkach zmiany są dokonywane bezpośrednio na poziomie bazy danych bez aktualizacji dokumentacji. Szczególnym przypadkiem są tzw. bazy spodkowe, w których gromadzone są dane mające znaczenie biznesowe, ale najczęściej bez właściwej dokumentacji. Problem interpretacji tych danych wymaga identyfikacji bytów w danej bazie. W tym celu można wykorzystać ontologie reprezentujące pojęcia modelowej dziedziny. Zadanie polega na znalezieniu ontologii reprezentującej dziedzinę, z której pochodzą dane lub ocenie jakości (wartości) danych w kontekście wybranej ontologii. W artykule zaproponowano metodę odwrotnej transformacji modeli danych, która ma zastosowanie dla relacyjnych baz danych w celu odtworzenia modelu konceptualnego na podstawie istniejącej bazy danych.

Addresses

Katarzyna Tubielewicz: InsERT S.A., ul. Jerzmanowska 2, 54-519 Wrocław, Polska, Katarzyna.Tubielewicz@insert.com.pl
Lech Tuzinkiewicz: Politechnika Wrocławska, Instytut Informatyki, ul. Wybrzeże Wyspiańskiego 27, 50-327 Wrocław, Polska, lech.tuzinkiewicz@pwr.wroc.pl.