A User Driven Method for Database Reverse Engineering

Aziz Barbar

I3S Laboratory, University of Nice
Bâtiment Euclide, Les Algorithmes
2000, route des Lucioles
06903 Sophia-Antipolis – France
Tel : +(33) 4 92 94 27 45, Fax: +(33) 4 92 94 28 96
E-mail : barbar@i3s.unice.fr

Abstract

In this thesis we describe the UQoRE method which supports database reverse engineering by using a data mining technique. Generally, Reverse Engineering methods work by using information extracted from data dictionaries, database extensions, application programs and expert users. The main differences between all these methods rely on the assumptions made on the a-priori knowledge available about the database (schema and constraints on attributes) as well as the user competence. Most of them are based on the attribute name consistency.

This paper presents a method based on user queries. Queries are stored in a "Query Base" and our system mines this new source of knowledge in order to discover hidden links and similarity between database elements.

1. Introduction

The evolution of databases has always been an active research area as indicated by the quantity of studies done in the past. Database reverse engineering (DRE) has become an important research field these last few years [BATI92, CHIA94] even though the pioneering work started at the very beginning of the 83s [CASA83].

Various reasons can motivate a Database Reverse Engineering Process (DREP) and imply various approaches. It can be necessary for moving the database implementation from a DataBase Management System (DBMS) to another one. More frequently, it consists in modifying the underlying logical data model, for instance changing a hierarchical database into a relational database or a relational database into an object oriented database. In our approach, we investigate the latter since object oriented area is closer to real complex data than relational paradigms. For example, most molecular biology database are implemented in relational DBMSs. But because relational databases are limited in their modeling capabilities of sophisticated scientific structures, maintenance tasks are tedious. DRE methods tend to design specifications by understanding existing database semantics. A DRE aims at producing a new description of the stored data by studying the database implemented in a specific DBMS. Various information sources can be used.

Our strategy is based on the hypothesis that essential semantic knowledge resides in queries expressed by expert users on the original database. We propose to enhance classical DRE methods by taking into account the semantics about data and structures which is implicitly included in these queries. This semantics is considered as knowledge and extracted by a data mining tool.

The rest of the paper is organized as follows. In section 2 we give an overview over different DRE methods. In section 3 we outline the UQoRE method we are designing. In section 4, we describe the main steps in our method and the similarity problem in the reverse engineering process via a distance extraction measure. Finally, section 5 contains the concluding remarks and our perspectives for future works.

2. DRE methods

In this section, we will present a general overview about existing DRE methods and the differences between them.

2.1. The Database Reverse Engineering Process (DREP)

Information systems are accepted now as a source of competitive advantage for organizations. They have to incorporate fast changes resulting from the evolutions that characterize enterprises today. Generally, in large organizations, the amount of saved and manipulated data is becoming more and more important; the information manipulated by automated procedures, have been designed and structured by several generations of analysts and database administrators. As a consequence, we are in a situation where we find a lot of redundant information, undeclared dependencies and incoherent sources [BLAH98]. Giving order to data becomes a necessity. Instead of designing a new system from the beginning, a reverse engineering process allows to design a new database schema by using the existing system and the old conception efforts.

2.2. Data models in DREP

DRE initial methods were concerned with the conventional files e.g. COBOL [CASA83, HAIN91]. The next step consisted of studying network and hierarchical databases [BATI92, HAIN95, WINA91]. Presently, they are rather concerned with the reverse engineering of relational databases. Most of the DREP works [AKOK98, ANDE94, CHIA94, FONK92, JOHA94, NAVA87, SIGN94, WATT96] consist in finding a conceptual schema after analyzing the relational database. These works often involve discovering an Entity-Relationship schema (*ER/EER/ECR*), but other algorithms are focused on object-oriented conceptual models [PREM94, TARI96].

The basic differences between all these proposed methods, come from the a-priori hypotheses on the database source state. These hypotheses are on the schema initial form, the naming of attributes, the source programs and the availability of all the information and knowledge on the launching of the process.

Unfortunately, these assumptions are not always realistic and more semantics should be extracted to meet real database cases.

2.3. Information Sources

Various sources of information are relevant for this semantics discovery task :
Physical schema, is obviously an important source of information although it may no longer be a good implementation of the conceptual schema. Successive updates have resulted in redundancy and incoherence.
Some DRE methods are based on the 3NF form assumption which can't be realistic, since real databases are de-normalized for access optimization first, and are updated by attribute insertion and suppression. [WATT96] proposed a method for de-optimizing databases before any other re-engineering operation. It is based on the assumption that attribute naming is consistent which is a strong condition.
Database extension, is an instance of the schema in which some constraints may be violated at a given time, querying this extension helps a DREP, ex. when looking for discovering domain value or inclusion dependencies between attributes that sometimes imply hierarchies between real world objects.
SQL statements, are used for accessing data and are embedded in application programs. SQL statement analysis is a valuable source of information for obtaining database semantics. Its interest has been mentioned only recently [PETI94] and in a limited extent. We consider that even if the application is not well defined, it can finally cope with the company needs via well-constructed SQL queries.
Expert user of the database, is important because he knows about the database modifications even if those updates were not written or added in the documentation, so, he can help in eliciting semantic ambiguities. A DREP must be interactive in order to ask the expert user as mentioned by [BLAH98].

Generally both the database schema and the data are analyzed [CHIA94, TARI96]. Programs and queries are used in methods which focus on the analysis of self-joins and equi-joins [PETI94]. Many of the past DREP are based on strong assumptions. Those hypotheses will lead to a simplified reverse-engineering process assuming that the database schema is in 3NF or that the attribute naming is consistent. In addition to that, we feel the need to explore a new source of data that can reveal a lot about database semantics : the user queries.

The weak points listed above, show the necessity of a new reverse engineering method that doesn't take into consideration the strong a-priori hypotheses. In our work, we propose to build a new method where all the assumptions are minimized and where a new source of data is explored to discover more hidden semantics in the underlying database model. This new method is called UQoRE and is the main goal of my thesis.

3. The UQoRE method

In the UQoRE (User Query oriented Reverse Engineering) method, we are interested in the reverse engineering of relational databases into an object-oriented model. Only the static aspect of the object model is taken into consideration here, since reverse engineering programs is beyond the scope of this paper. Our study aims at providing a tool for helping discovering object classes and links between them by extracting knowledge from all available sources of the initial relational database. The relationships doing the links between classes are based on shared attributes and are identified mainly by foreign keys. Discovering shared attributes is thus a main step.

All existing information sources have been exploited in the past DREP, but the problem of discovering the hidden semantics persists. This problem proves the need for a new source of information : we consider the user queries as a valuable knowledge that reveals the real application semantics. The UQoRE method consists in saving the queries that have been typed and composed by the user on his screen when working on the application. We consider that this user is an expert one, and that after a certain time of exploiting the application, he is able to understand the goal of this application and relatively the underlying semantics. We suppose that, even if the application has many defects, this user adapt his queries in order to extract the appropriate information by formulating the query filling on his screen some attributes in a precise way.

We propose to enhance previous DRE methods by taking into account the semantics about data and structures which is implicitly included in these queries. This set of queries is going to be saved in a database named “Query Base” and because of its dimensions, this database is going to be exploited by Data Mining techniques : Association rules, classification, distance measure, etc … to extract the hidden semantics of the database application and construct the object conceptual model starting with a relational database.

4. Preliminary results

In this paper, we will present the principal steps for our DREP. These steps are as follows :

- *Similarity extraction :* Attribute identification by name is not consistent. Every schema update may introduce new attribute names with no strict terminology respecting the integrity naming. Homonymous with no common semantics may be inserted and synonymous may have nothing to share. Reverse engineering methods e.g. [CHIA94, TARI96, WATT96] tend to establish similarity between attributes by strictly comparing their value sets. [PETI94] proposed to ignore attribute names and to exploit equi-joins for selecting similar attributes. We consider this criteria just as an indication, but the overall process cannot be based on it since it may introduce errors. For instance, we may find two primary keys involved in an equi-join, having identical value sets and implementing non-similar concepts.

Let us look at the following query which is used to find cars which names are flowers names :

```
SELECT  C.name
FROM  FLOWERS F, CARS C
WHERE  F.name = C.name ;
```

In this case, the relation F represents flowers (name is the primary key) and the relation C represents car models (name is the key attribute). F.name and C.name check equi-join and value sets criteria, but they are not similar.

This phase in really crucial for the resulting process efficiency and will be accomplished by a distance computation. We define the similarity distance between two attributes X_1 and X_2 as a measure that represents the degree of similarity of their context in the queries. X_1 and X_2 are considered similar if they are used in similar contexts. Let us assume that :

- Q_1 and Q_2 are sets of queries using X_1 and X_2 in their WHERE clauses and $d(X_1, X_2)$ represents the distance between X_1 and X_2.
- X_1 and X_2 are considered as similar if $d(X_1, X_2)$ is close to 0. Thus :
  $d(X_1, X_2) = d(X_2, X_1)$
  $d(X_1, X_1) = 0$
- We define $d(X_1, X_2)$ as the distance between the associated sets Q_1 and Q_2.

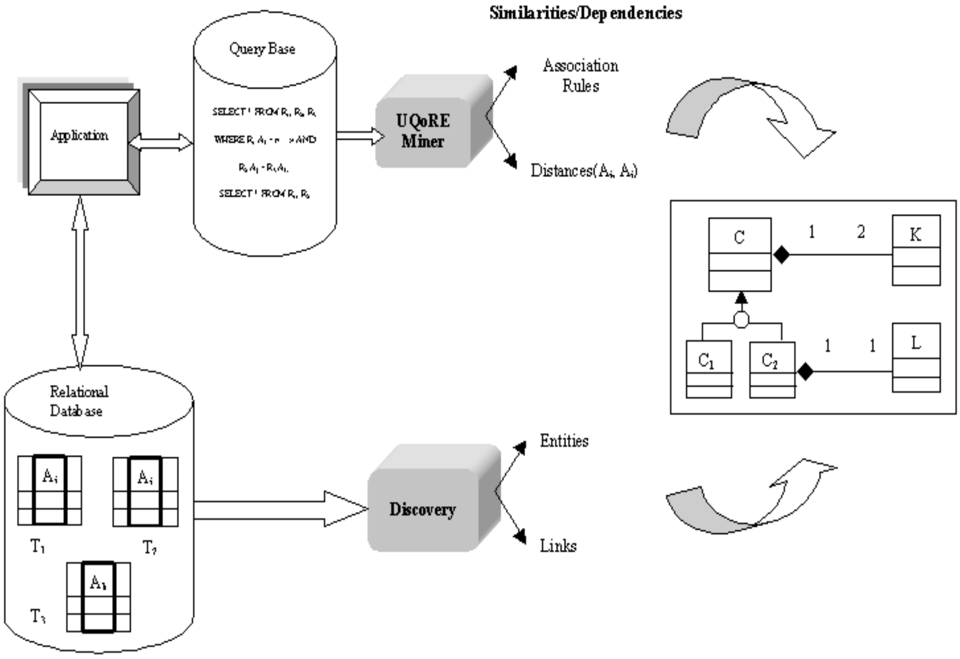
At the end of this step, the user will intervien to give a unique name for attributes having the same concept with different names so that the naming will become consistent. At the same time, attributes with the same names and having different concepts will be identified.

- *De-optimization* : this step includes normalizing and restructuring the schema. The normalization step migrates the initial schema to a 3NF schema. This step is neglected by the majority of reverse engineering methods which assume generally that they are provided with a 3NF database. We can only quote [WATT96] that talked about this problem and showed many restructuring and normalization procedures to recover the initial model. A realistic approach cannot be based on this assumption which obviously simplifies the process description. Indeed for achieving normalization, functional dependencies are detected and non 3NF relations are split. New attributes are appearing from splitting. The restructuring step consists in merging or separating some columns, and deleting others. The defflattering case have to be mentioned in this step too. A frequent question to solve in each of the steps above is how to detect similar and distinct attributes in order to decide merging or splitting candidate structures.

- *Discovery of primary/foreign keys :* we detect the generalization/specialization links (primary keys), and, other links between classes will be shown because of foreign keys. In this step, the value sets will let us discover the inclusion dependencies between the different components of the database schema.

- *Schema translation :* once the database schema is in 3NF, we can translate each relation to a class or association between classes depending on the primary key structure. The generalization/specialization links will be shown too.

The overall process can be shown in the schema below :



UQoRE (User Query-oriented Reverse Engineering) Method

5. Conclusion

We have investigated 2 recent domains : the reverse engineering of relational databases and the data mining. We showed some of the recent methods related to the DREP and prove the need to exploit a new information source. Our main goal is not to fully automate the DREP, we know very well that this process must be interactive, but our contribution concerns a more realistic approach where the administrators or database designers intervention is limited.

We exposed the steps to be executed in our UQoRE method. The originality of UQoRE consists in mining a large database of collected queries from expert users in order to extract semantic relationship that may be expressed by them only. In one of our papers [BARB01] we talked about the attribute similarity extraction in UQoRE and our perspectives is to show by data mining techniques the semantic extraction in the second step of our method and to insert these results in the overall reverse engineering process.

References

[AKOK98] J. Akoka, I. Wattiau. *Une méthode de rétroconception de bases de données relationnelles*. Inforsid, Villeurbanne, Février 1998.

[ANDE94] M. Andersson. *Extracting an Entity Relationship Schema from a Relational Database through Reverse Engineering*. Proc. Of the 13th Conf. On ER Approach, Manchester, UK, Dec 1994.

[BARB01] A. Barbar, M. Collard. *A Distance-Based Approach for Database Re-engineering*. Proc. Of the ACS/IEEE International Conference on Computer Systems and Applications, Beirut, June 2001.

[BATI92] C. Batini, S. Ceri and S. Navathe. *Conceptual Database Design : an Entity-Relationship Approach*. Benjamin Cummings, 1992.

[BLAH98] M. Blaha. *On Reverse Engineering of Vendor Databases*. 5th Working Conference on Reverse Engineering, Honolulu, Hawaii, October 1998.

[CASA83] M.A. Casanova and J.E.A. de Sa. *Designing Entity-Relationship Schemas for Conventional Information Systems*. In Proc. Of the 3td Int. Conf. On the ER Approach to Software Engineering , pages 265-277, Anaheim, California , 1983. Elsevier Science Publishers.

[CHIA94] R.H.L Chiang, T-M Barron and V.C Storey. *Reverse Engineering of Relational Databases : Extraction of an EER Model from a Relational Database*. Data and Knowledge Engineering, 12, pp. 107-142 (1994).

[FONK92] M. Fonkam, W.A. Gray. *An Approach to Eliciting the semantics of Relational Databases*, Proc. Of the 4th Int. Conf. On Advance Information Systems Engineering-CaiSE'92, pp. 463-480, Springer-Verlag, 1992.

[HAIN95] J.L. Hainaut, V. Engelbert, J. Henrard, J.M. Hick, D. Roland. *Requirements for Information System Reverse Engineering Support*. Proc. Of the IEEE Working Conf. On Reverse Engineering, Toronto, Canada, IEEE Computer Society Press, July 1995.

[JOHA94] P. Johannesson. *A method for transforming Relational Schemas into Conceptual Schemas*. IEEE Int. Cong. On Data Engineering (ICDE), Los Alamitos, pp. 190-201 (1994).

[NAVA87] S. Navathe and A. Awong. *Abstracting Relational and Hierarchical Data with a Semantic Data Model*. In Proc. Of the 6th Int. Conf. On the ER Approach, pages 277-305, New-York, Nov. 1987.

[PETI94] J-M. Petit, J. Kouloumdjian, J-F. Boulicaut and F. Toumani. *Using Queries to Improve Database Reverse Engineering*. Int. Conf. On the Entity-Relationship Approach (ERA), Manchester, pp. 369-386 (1994).

[PREM94] W. Premerlani and M. Blaha. *An approach for Reverse Engineering of Relational Databases*. Communications of the ACM, 37(5), 42-49 (1994).

[SIGN94] O. Signore, M. Loffredo, M. Gregori and M. Cima. *Reconstruction of ER Schema from Database Applications : a Cognitive Approach*. Proc. of the 13th International Conference Approach, Manchester UK, december 94, pp. 387-402.

[TARI96] Z. Tari, J. Stokes and S. Hammodi. *The Reengineering of Relational Databases based on Key and Data Correlations*. IFIP 1996. Published by Chapman & Hall, pp. 183-214.

[WATT96] I. Wattiau and J. Akoka. *Reverse Engineering of Relational Database Physical Schemas*. Proc. 15th International Entity Relationship Conference, Cottbus, Allemagne, 1996.