Reengineering of relational Databases to Object Oriented Database

Mr.R.S.Mawale[1], Prof.A.V.Deorankar[2], Prof. P.U. Malve[3]

[1] M.Tech. Scholor, (CSE), Govt. College of Engineering, Amravati, Maharashtra, India
[2] Associate Professor, Dept. of CSE, Govt. College of Engineering, Amravati, Maharashtra, India
[3] Lecturer, Government Polytechnic, Murtizapur , Dept. of Computer Engineering, Maharashtra, India

Abstract:

Many information systems use relational database systems for efficient sharing, storage, and retrieval of large quantities of data. On the other hand, object-oriented programming has been gaining wide acceptance in the programming community as a paradigm for developing complex applications that are easy to extend and maintain. This paper discusses development of an integrated environment which maps a relational schema to an object-oriented schema without the need to modify the existing relational schema and providing a platform for migrating data from relational database to object oriented database.

Keywords: Schema Mapping, RDBMS, OODBMS, Reengineering

1. Introduction

Relational database management systems (RDBMS) provide a variety of tools and services for data management. There are many tools that interface with RDBMSs to enable end-users to carry out reporting, querying, and other data analysis activities easily. On the other hand, object-oriented programming has been gaining wide acceptance in the programming community as a paradigm for developing complex applications that are easy to extend and maintain. Developers typically implement object-oriented applications using object-oriented programming languages such as C# and Java. These applications use a schema made up of object classes and relationships between those object classes. Each object has a set of attributes. The value of an attribute could be another object itself, thus giving rise to complex objects. The main problem arises when the data corresponding to such objects are persistent in a relational database. The problem is due to incompatibility between relations and objects. At first, a solution for this problem may seem to be to use an ODBMS instead of RDBMS. If an ODBMS is managing the persistent data, the objects do not lose their structure after the application stores them in the object database. Moving to an ODBMS might mean throwing away all of the old ("legacy") data and applications. Most users of databases will not accept such a solution. They wish to be able to run their existing applications on existing databases and have access to the same data from object-oriented programs, too. Therefore, we need special techniques to convert the data that is residing in a relational database to a format that is suitable for access and manipulation by object-oriented applications.

2. Literature review

The problem of migrating data is present in almost every application development process, such as data warehousing and application integration. The algorithm implements process of migrating data involves firstly the mapping between the structures of the source and target databases and secondly the migration of the data from the source to the target [3].

The problem is how to effectively migrate existing RDBs, as a source, into OODB/ORDB/XML, as targets, and what is the best way to enrich and maintain RDBs' semantics and constraints in order to meet the characteristics of the three targets? Existing work does not appear to provide a complete solution for more than one target database. We tackle this question by proposing a solution for migrating an RDB into the three targets based on available standards [4].

One general approach to migrate to object technology is to divide the process into two phases, where the first one transforms the relational into an object-oriented schema and the second one migrates the data into the object-oriented database system discussed in [5].

The goal of re-engineering is to mechanically reuse past development efforts particular relational databases (RDBs).in order to reduce maintenance expense and improve software flexibility [4]. The object behavior of eight Java programs including four real-world Java object oriented database management systems and a counterpart of four real-world Java programs are analyzed in [6]. A technique for transferring query optimization techniques, developed for relational databases, into object Databases. This technique for ODMG database schemas defined in ODL and object queries expressed in OQL. The object schema is represented using a logical representation (Datalog) [7].

3. Analysis of problem

Moving to an ODBMS might mean throwing away all of the old ("legacy") data and applications. Most users of databases will not accept such a solution. They wish to be able to run their existing databases and have access to the

same data from object-oriented program.

Thus we need to implement a system that builds an understanding of a given conventional database by taking these characteristics as input and produces the corresponding object-oriented database as output. Finally, we handle the migration of data from the conventional database to the constructed object-oriented database.

The Primary objectives of proposed work are as follows.

1. Study the automatability of the relational-to-OO schema mapping process.
2. Define an interactive process for mapping an existing relational schema to an object-oriented schema.
3. Develop an interactive system to validate proposed work

4. System architecture

The architecture contains two major components needed for fulfilling our aim. The first component deals with mapping the relational schema to an object-oriented schema. The second component deals with the mapping between the relational data and objects.

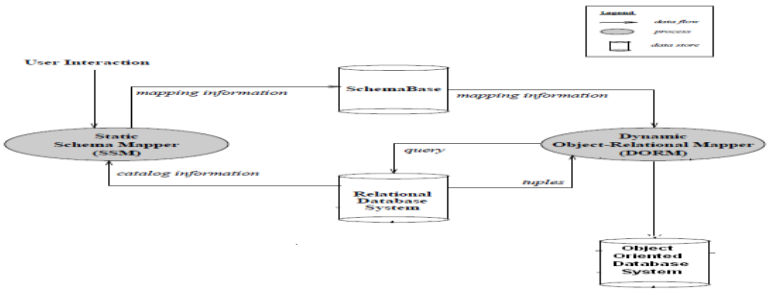

Figure 1. System Architecture

Schema Mapping

The static schema mapping process is a two-phase process. In the first phase, the relational schema is adjusted and transformed into another virtual relational schema that has some specific properties. In the second phase, object-oriented structures are extracted from the virtual relational schema.

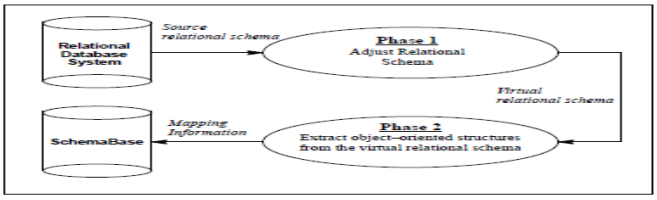

Figure 2.Two Phases of Static Schema Mapping

Phase I: Adjusting the Relational Schema

There are four specific aspects that are addressed during phase one. They are as follows:

Step 1. Eliminate 2NF relations and replace them with new 3NF virtual relations.

Step 2. Create virtual subclass relations for widow super class relations

Step 3. Create virtual superclass relations for orphan subclass relations.

Step 4. Eliminate multi-valued attributes and replace them with new 3NF virtual relations.

Phase II: Generation of the Object Schema

At the end of phase one of the schema mapping process, the relational schema has been adjusted to a form in which schema mapping rules can be applied uniformly.

Step1. Identifying Object Classes:

Those relations that correspond to object- classes must be identified.

Step2. Identifying Relationship:

There are three types of relationships that can be represented in an object model. They are associations, generalizations specializations, and aggregations.

Identifying each of these constructs constitutes a step in the mapping process.

Identifying Associations. Since the object model allows associations to be modeled as classes, we must either establish a simple association between two object classes or identify relationships where the associations are modeled as classes.

Identifying Inheritance. Inheritance structures capture the generalization and specialization relationships between object classes that have been identified so far.

Identifying Aggregation. The aggregation relationship models the composition of one object with other objects. such complex object must be identified. The difference between aggregation and association is that the former involves existence dependence of the sub-object on the whole object. For example, a door object, which is a part-of" of a car object, cannot exist if the car object does not exist. On the other hand, the enrollment of a student in a course is an association rather than an aggregation because the student and course objects can exist independent of each other.

Step 3. Establishing Cardinalities : Establishing the cardinalities of associations is important in order to facilitate the implementation of the object schema in a given programming language (e.g., C#). The different possible cardinalities are one-one, one-many, and many-many.

5. Conclusion

We will implement the architecture having two phases in the data mapping process; one mapping procedure specifies the data mapping between the original relational schema and the adjusted relational schema. The second mapping procedure specifies the data mapping between the adjusted relational schema and the object schema. The data mapping procedures have been specified using relational algebra for each new virtual relation that is created during phase one of the static schema mapping.

References.

[1] Ramanathan, C. Providing object-oriented access to a relational database. In Proceedings of the 32nd ACM annual southeast conference held in Tuscaloosa, Alabama, March, 1994.

[2] Andersson M. Extracting an entity-relationship schema from a relational database through reverse engineering. In Proceedings of the 13th international conference on entity relationship approach held in Manchester, UK, December, 1991

[3] Elton Manoku, Guido Bakema, A fact approach on Data Migration published in Academy of Communication and Information Technology HAN University of Applied Sciences, The Netherlands In June 2006

[4] Abdelsalam Maatuk, Akhtar Ali, and Nick Rossiter , A Framework for Relational Database Migration published in School of Computing, Engineering & Information Sciences Northumbria University, Newcastle upon Tyne, UK in 2001

[5] Andreas Behm, Andreas Geppert, Klaus R. Dittrich2 ,On the migration of relational schemas and data to object-oriented database published in Proceedings of the 5th International Conference on Re-Technologies in Information Systems, Klagenfurt, Austria, December 1997

[6] William J Premerlani, Michael R Blaha, An Approach for Reverse Engineering of Relational Databases published in May 1994 Communications of the ACM

[7] Chia-Tien Dan Lo, Morris Chang, Ophir Frieder and David Grossman, The Object Behavior of Java Object-Oriented Database Management Systems published in the Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC.02) IEEE 2002

[8] John Grant, Jarek Gryz, Jack Minker, and Louiqa Raschid, Logic-Based Query Optimization for Object Databases published in IEEE Transactions on knowledge and data engineering, vol. 12, no. 4, July/August 2000.

Authors Profile:

Mr.R.S.Mawale has received his B.E. degree from Govt. College of Engineering, Amravati, India. He is currently pursing his M.Tech in Computer Science & Engineering from Govt. College of Engineering, Amravati, India. He has published one paper in International Journal.

Prof A.V.Deorankar has received his M.E. degree in Electronics Engineering from Govt. College of Engineering, Amravati, India and pursing his PHD from Computer Science. He has 19 years experience in Teaching in various Government engineering Colleges in Maharashtra. He has published nine papers in national and seven papers in international journals. He has also patent of one paper. His area of research includes Computer network, Web Mining. Currently he is working as an Associate Professor at Govt. college of Engineering, Amravati, Maharashtra, India .

Prof.P.U.Malve has received His M.E.Degree in Information Technology from Sant Gadegebaba Amravati University in 2011. He has published Two papers in National and Three papers in International Journals. He has Five years experience in teaching various Government engineering Colleges & Polytechnics in Maharashtra. His area of research includes Computer Networks, Software Engineering, Network Security He is currently woking as a Lecturer in Department of Computer Engineering, at Government Polytechnic, Murtizapur, Maharashtra India