REENGINEERING OF RELATIONAL DATABASES TO OBJECT-ORIENTED DATABASE

Vinay Goyal[1], Amit Jain[2]

[1]*Director, Haryana Institute of Technology, Bahadurgarh (Haryana), INDIA*
[2]*Research Scholar (CS), Teerthanker Mahaveer University, Moradabad (UP), INDIA*

Abstract

Several information systems use relational database systems for effective dealing, storage, and accessing large measures of data. On the contrary, object-oriented programming has earned wide acceptance in the programming community as a epitome for developing typical applications that are easy to expand and uphold. This paper deal with development of an unified environment which relates a relational schema to an object-oriented schema without the requirement to manipulate the existing relational schema and providing a scenario for transferring data from relational database to object- oriented database.

Keywords: *Schema Mapping, RDBMS, OODBMS, Reengineering*

----------***----------

1. INTRODUCTION

A relational database management system (RDBMS) is a term used to describe an entire suite of programs for both managing a relational database and communicating with that relational database engine. Sometimes Software Development Kit (SDK) front-end tools and complete management kits are included with relational database packages.

On the other hand, object-oriented programming (OOP) drove the creation of C++, it is necessary to understand its foundational principles. OOP is a powerful way to approach the job of programming. Programming methodologies have changed dramatically since the invention of the computer, primarily to accommodate the increasing complexity of programs.

Programmers typically apply object-oriented applications with object-oriented programming languages such as C# as well as Java. These applications use a architecture made up of object, classes and relationships between those object & classes. Each object has a set of features. The significance of an attribute could be a new object itself, thus augmenting to complex objects.

The key trouble ascends when the data analogous to such objects are determined in a relational database. The difficulty is due to mismatch amongst relations and objects. At first, a elucidation for this problem may seem to be to use an ODBMS instead of RDBMS. If an ODBMS is handling the determined 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.

Several users of databases will not assume such a explanation. Some will desire to execute their existing applications on existing databases and have access to the similar data from object-oriented programs, too. Thus, we need distinct procedure to transform the data that is inherent in a relational database to a design that is proper for access and handling by object-oriented applications.

2. EXISTING SCENARIO

The issue of migrating data is existent in virtually each application development process, such as data warehousing and application incorporation. The process of migrating data includes the mapping amongst the structures of the source and target databases as well as the movement of the data from the source to the target [3].

The ISSUE is how to efficiently migrate current RDBs, as a source, into OODB/ORDB/XML, as targets, and what is the finest way to improve and sustain RDBs' semantics and restraints in order to meet the features of the three targets? Present effort does not seem to offer a comprehensive solution for more than one target database. We hold this problem by suggesting a solution for migrating an RDB into the three targets established on existing ideals [4].

One common methodology to migrate to object technology is to split the progression 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 [5]. The objective of re-engineering is to instinctively reprocess past expansion efforts particular to relational databases (RDBs) in order to moderate maintenance overhead and expand software flexibility [4].

There exists a technique for transforming query optimization developed for relational databases into object databases. This technique for ODMG database schemas defined in ODL and object queries expressed in OQL is represented using a logical representation [7].

3. ANALYSIS OF PROBLEM

Moving to an ODBMS might mean throwing away all of the old ("legacy") data and applications. Several users of databases will not consent such a elucidation. There is a desire to run their prevailing databases and have access to the similar data from object-oriented program. Thus we require to execute a system that provides an acceptance of a given conventional database by captivating these features as input and yields the agreeing object-oriented database as output. Finally, we handle the migration of data from the conventional database to the constructed object -oriented database.

The main goals of suggested effort are as follows:
1. Analyzing the computerization of the relational to Object-Oriented schema mapping process.
2. Illustrating an mutual procedure for mapping an prevailing relational schema to an object -oriented schema.
3. To make an interactive system to validate proposed work.

4. SCHEMA DESIGN

The design encompasses of two chief constituents needed for accomplishing our objective. The first constituent relates to mapping the relational schema to an object-oriented schema. The second constituent relates to the mapping amongst the relational data in addition to objects.

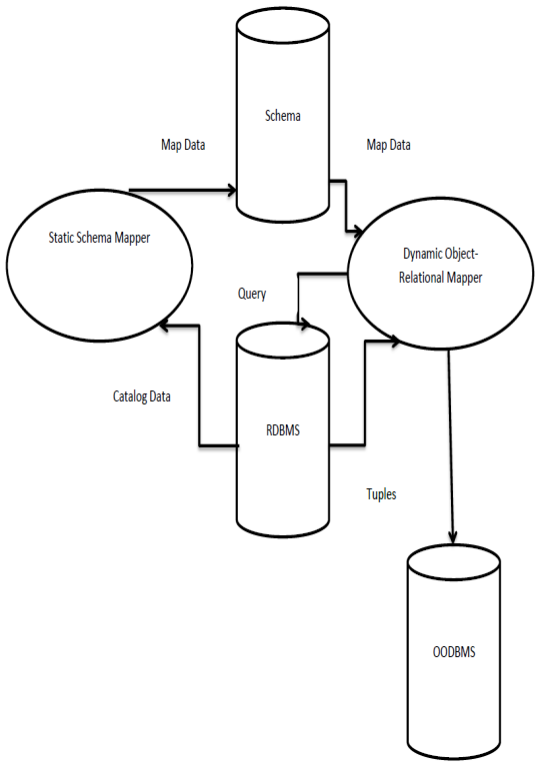

Fig 1 System Architecture

4.1 Schema Mapping

The static schema mapping process is a two-phase process. In the initial phase, the relational schema is accustomed and transmuted into alternate virtual relational schema that has particular properties.

In the subsequent phase, object-oriented structures are mined from the virtual relational schema.

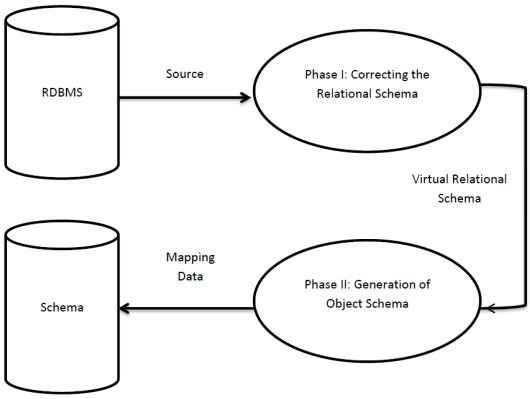

Fig 2 Two Phases of Static Schema Mapping

4.1.1 Phase I: Correcting the Relational Schema

There are four characteristics that are taken into account during phase one. They are as follows:
Step 1: {Exclude 2NF relations and substitute them with novel 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: Exclude multi-valued attributes and substitute them with novel 3NF virtual relations.

4.1.2 Phase II: Generation of the Object Schema

At the end of phase one of the schema mapping process, the relational schema has been accustomed to a find out, which schema mapping guidelines can be applied consistently.

Step1. Recognizing Object Classes:

Those relations that relate to object- classes must be recognized.

Step2. Recognizing Relationship:

There are three kinds of associations that can be characterized in an object model. They are associations, generalizations-specializations, and aggregations. Recognizing every constructs establishes a stage in the mapping process.

Recognizing Associations:

Since the object model permits associations to be modeled as classes, we must either form a simple association between two object classes or categorize relationships where the associations are displayed as classes.

Recognizing Inheritance:

Inheritance structures capture the generalization and specialization associations between object classes that have been recognized so far.

Recognizing Aggregation:

The aggregation association models the configuration of one object with other objects. The dissimilarity amongst aggregation and association is that the earlier one encompasses 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: Forming Cardinalities

Creating the cardinalities of associations is significant in order to expedite the execution of the object schema in a specified programming language (e.g., C#). The diverse probable cardinalities are one-one, one-many, and many-many.

CONCLUSIONS

Thus there exist an architecture having two phases in the data mapping process; one mapping procedure postulates the data mapping between the original relational schema and the accustomed relational schema. The second mapping process postulates the data mapping between the accustomed relational schema and the object schema. The data mapping measures have been stated using relational algebra for each new simulated relation that is formed 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.

BIOGRAPHIES:

Dr. Vinay Goyal, Internationally recognized expert in Software Engineering and Academician and has more than 15 years experience, published numerous papers in this field. His current research interests include Soft computing Techniques, Data Compression, Software Re-Engineering and Mathematical Modeling. He is presently, designated as Director, Haryana Institute of Technology, Bahadurgarh (Haryana) India. He has two books in his credit.

Amit Jain is a Doctoral Researcher at Department of Computer Application at Teerthanker Mahaveer University, Moradabad (Uttar Pradesh) India. He has around 11 years of teaching experience. He is currently working on Legacy Systems, Re-Engineering, Software Evolution and Software Migration. His research interest includes Service-oriented architecture, Service-oriented computing.