Experimental analysis on changing data from relational to article-oriented databases

Richa Tomer*, Vishal Bhatnagr, Md. Asadullah, Md. Shabbir Hassan, Suphian Khan

Department of Computer Science, Shri Venkateshwara University, Gajraula Distt- Amroha (U.P), India

ABSTRACT

Enormous alterations has been going on in IT for last few years. Owing to the expeditious development in this field, the requirement for invention in this field is much higher. This necessitates prominent attempt of firms to react rapidly to market considerations in order to coordinate job and carry on trade more expeditiously. Firms has to orchestrate subsisting class of Modern fundamental technologies like World Extensive Web or Computerized Commerce. A distinctive assumption in many firms when employing a orchestrate process is That on one hand, a prominent body of data is grabbed in relational or even hierarchal or network databases, and on the different hand article oriented applications has to be developed. So that, a Modern article model is constructed that represents the present state of the firm's trade procedures. Nevertheless, the Modern article model and the subsisting relational database's model generally do not go well unitedly. In different words, a prominent semantic spread amongst both models should be coupled. The approach indicated in this paper is database migration. Essentially, this approach comprises two tasks. In the initial task, the relational database outline is organize. The outline is translated into a well formulated and putatively apprehensible article oriented outline, that the Modern applications may conform. Subsequently, the data are migrated into an article oriented DBMS.

An approach is indicated for a conventional resolution of data migration process. The outline transformation process is separated into three successive degrees. In the initial degree, the relational outline is translated into an SOT outline. This initial SOT outline is then redesigned preeminent in the decent article oriented outline. Eventually, in the third degree preeminent SOT outline is translated into an article oriented outline according to the ODMG dominant. The data migration process is generated automatically for each outline transformation degree. In order to enforce outline transformation, the idea of transformation rule is indicated. The transformation rules define elementary restructuring operations within this model. A introductory transformation rules has been indicated that may be continued. Eventually, a prototype has been implemented as a evidence of idea**.**

Keywords: E-commerce, WWW (World Wide Web), relational outline, life cycle, article technology.

Introduction

Usually process deals all fields of information arrangements. In this paper, in particular, three fields are of primary concern.

Relational Databases Arrangements

RD system represents the present dominant in technology for realizing database applications. The idea was indicated in the early seventies, and now commercial DBMS (database management arrangements) like DB2, oracle, Informix Sybase prevail the market of data persistence. A prominent body of computerized data is to be read in relational database now a days. One major bonus of RDBMS is the maturity they has achieved in pervasive research Attempts in the last years. This permits RDMS products to be employed for great performance and mission-decisive database applications.

**Correspondence*

Richa Tomer

Department of Computer Science, Shri Venkateshwara University, Gajraula Distt- Amroha (U.P), India.

Article Orientation

Today article oriented epitome prevails in modern software development. It has egressed as significant technology to ascertain software engineering properties. Nearly all the elements of the Modern information arrangements are developed within an article-oriented software engineering event cycle. In particular, these elements include database arrangements, user interfaces, operating arrangements and applications. The degree of article–oriented software engineering comprises, amongst others, analysis, design and effectuation.

Reengineering: The expeditious alterations in IT and society force firms to rapidly react to changing considerations on a global market. This raises the difficulty of reengineering information system capitalize these technologies. Usually, two cases propelling this may be distinguished. Firstly, alterations in the internal organization of firms should be reflected in information arrangements. Secondly, owing to the emerging fundamental technologies like E-commerce, the WWW or data-ware housing, firms has to conform and modify parts of their information arrangements.

The evolutions of database technology and software engineering has hardly acted upon each other in the last decades. As a result, RDBMS and article-orientation comprise primarily different epitomes. Subsisting relational databases and article-oriented applications cannot be incorporated in a unseamed manner, a difficulty that is known as impedance mismatch.

Meantime, article Oriented DataBase Management System (OODBMS) has been indicated to support unseamed integration of article technology and data persistence, and various commercial products are available. Nevertheless, many establishments primarily abstain from utilizing OODBMS since subsisting products cannot contend with RDBMS with respect to maturity and dependability.

The approach indicated in this paper is database migration. Essentially, this approach comprises two tasks. In the initial task, the relational database outline is organize. The outline is translated into a well-formulated and putatively apprehensible article-oriented outline, that the Modern applications may conform. Subsequently, the data are migrated to OODBMS. There are various grounds why database migration worth advance investigation. Initial, database migration predicts good results than different approaches like article expressions for the upcoming grounds. Since the data are converted to objects only once, database migration primarily permits more tractability with respect to the reengineering the relational outline into a suitable article- oriented outline. Secondly, subsisting approaches for database migration do not exploit the complete potential of the article-oriented epitome, so that the preeminent article oriented outline still looks relational and retains subsisting advantages and disadvantages of the relational outline. Eventually, Attempts are taken to redress the present naiveness of commercial OODBMS for some mission-crucial database applications. They are well befitted for saving data of typical such as CAD arrangements and office automation arrangements. Lately, article-Relational DBMS (ORDBMS) has initiated to extend some article–oriented characteristics, and advance characteristics are likely to be addressed in the future. ORDBMS may be expected to extend present commercial RDBMS products. Hence, they extend the similar dependability as RDBMS. ORDBMS has the similar difficulty, as referred earlier, for OODBMS, viz. to change a subsisting relational outline in to Modern exploiting article-oriented characteristics, and to conform the database. Thus, the results present in this paper for OODBMS will also be important for the article-relational database arrangements

Relational dtabase concepts

Relational database arrangements represent the dominant technique for realizing database applications. The primary bases of relational databases were put in early seventies, mainly in the relational data model[Cod70]. Afterward, the Entity-relation model[Che76] was indicated for simple yet effective simulation of relational database schemas.

Data models

The process of designing database outline is generally decomposed into various degrees. In each degree, a different data model is employed. Typically three levels of designs are distinguished, as shown in Figure-1: conceptual design, logical design and physical design.

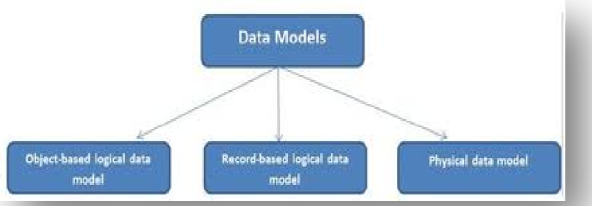
Data Models
Object-based logical data model
Record-based logical data model
Physical data model

Figure-1:Data Models

Logical design consists of representing the conceptual outline into a logical outline that may be handled by the DBMS. The logical outline is thus expressed by means of data manipulation language. Eventually, the physical outline describes the internal storage structure of the database.

The ER model

The Entity relation(ER) model was introduced by Chen[Che76], and describes data as entities, relationships and properties. An entity is a "thing" in the real world with an autonomous existence. Each entity has properties- the particular properties that describe it.

The relational data model is one of the traditional data models like the network model and hierarchical models. These has been rather successful in developing the database technology for many traditional data base applications Relational database design follows a waterfall–oriented approach. Firstly, a Conceptual outline is made, generally by means of ER model. Such a conceptual outline describes the universe of disclosure as a set of entities that are characterized by a number of properties. In addition, relation amongst entities may also be described. In the second stage of relational database design, logical design, a relational outline is made in terms of a data resolution language that may be handled by a DBMS. Sometimes, specific constricts in the ER outline such as generalization relationships, may arouse various alternative constructs in the relational outline. Although parts of the specified semantics in the ER outline may get lost in the relational outline, the latter may be enhanced by totalling restraints or expressions.

article–oriented concepts are marginally older than relational concepts. Nevertheless, it took a long time for article-orientation to accede the primary stream. In the late seventies, language Smalltalk-80 was introduced by Xerox[GR89]. Parallel to the development of the article-oriented programming languages, conventional languages were continued with concepts for realizing abstract data types and information hiding, in order to satisfy Modern necessities of software engineering. Examples of such languages are Ada and Modula. Nevertheless neat article oriented languages were regarded the best choice for integrating Modern software engineering necessities, supporting reprocess, maintaining software, sustaining the article oriented software engineering life cycle [ABD+89].

Specification of article behaviour

From the relational point of view, there is no reason to explicitly model composite objects in the relational outline. It is sufficient that composite objects may be extracted from the database, where a view may be described containing composite objects utilizing the question referred above. In a article-oriented design, various article specific operations may be specified, for composite objects. In an article-oriented design, nevertheless, article specific operations for points may ideally be implemented as methods of a class point.

The primary contribution of this paper has been to demonstrate that relational and article-oriented database design follow different design strategies and consequently result in structurally different database schemas. There is no ubiquitous representing scheme amongst relational constructs on the one side and article-oriented constructs on the different. In particular, this concerns the primary construct relations, tuples, classes and objects.

Relation and Classes: The examples demonstrated that not every relation in a relational outline corresponds to a class in a corresponding article-oriented outline. Conversely, not every class present in a article-oriented outline is derived from a corresponding relation in a relational outline.

Tuples and Objects: Not every tuple in a relational database is represented as an article-oriented database. Conversely, not every article is derived from one specific tuple. Grounds for this are the deficiency of conceptual simulating constructs in relational database design, and the possible specification of the article life cycles.

The contributions of this paper from the necessities for the migration approach. This paper demonstrate the result of these different design strategies, that is that converting a relational outline to a corresponding article-oriented outline which is a non-superficial task. Consequently, the migration algorithms should be effective enough to support such conversations.

Migration to article technology

The extensive acceptance of article technology in software engineering has propelled the combination article technology and data persistence. Currently, the attempt to smoothly incorporate relational data and article technology is especially eminent when Modern article-oriented applications has to access subsisting relational databases. Relevant approaches are referred and the advantages and disadvantages of each of them are compared. Database migration, the subject of this paper, is one of the approaches. Usually, two different approaches may be distinguished. Firstly, the data is retained in relational databases. The simplest resolution, a entryway amongst a relational and an article-oriented database.

Database migration is the second approach, in that the RDBMs is completely substituted by an article-oriented one, and both outline and database should be migrated. The decision to employ a particular migration scheme for a certain database system depends on various pragmatic restraints.

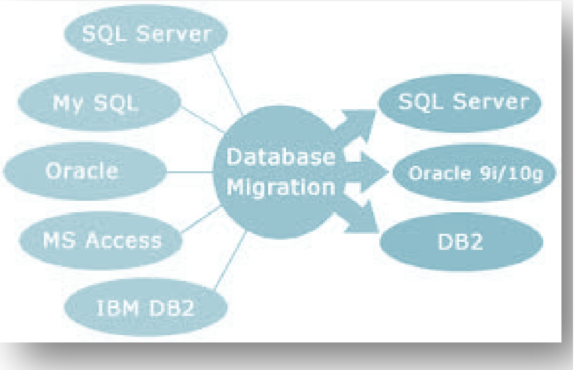

Figure-2:DB Migration

The transformation in principle, every table of the relational outline is mapped into a class and foreign fundamental restraints are mapped into reference properties. Consequently, every tuple of the relational database is represented as an article in the article view.

Relational rhyming

The transformation of article-oriented outline into a relational alone, when utilizing objects for applications and relations for persistence. Since both article- relational representing and relational–article representing share the task of specifying mappings amongst relational constructs and article-oriented constructs, various article-relational representing product also support the opposite direction. Research projects in this field are Penguin[KH93] and article Driver[Leb93].

The Construction of article expressions is not simply the opposite task of the article-relational representing process, that may be executed automatically. In the case of article-relational representing, the article-oriented outline has been made by forward engineering. When creating article expressions the relational outline should undergo a reverse engineering process.

The algorithm for realizing the insertion operation is non-superficial. The index value of the tuple and the insertion point where the Modern element is inserted should be calculated. Then, all the elements behind the insertion point should be shifted one location to the right.

For combinational relational databases and article technology has been referred. In principle, two different strategies exist: article expressions over relational database and database migration. Although both strategies appear to very different, often pragmatic restraints determine that migration scheme to use. As regards, article expressions, various approaches has been indicated and various commercial instruments already exist. Although this scheme exhibits serious problems, At least for performance grounds, it represents the present trend of migration to article technology.

Semantic enrichment

Semantic enrichment is the task of gathering additional semantic information that is not explicitly available from the relational database system. This task is also known as reverse engineering, that stresses that the result of the semantic enrichment process may be represented through a conceptual outline. Whereas the process of database design is called forward engineering, reverse engineering may be regarded the opposite process, that is, there construction of conceptual outline out of an subsisting database. Reverse engineering is not only essential in the field of information migration, but of high importance for information arrangements reengineering usually.

There are various grounds for employing reverse engineering to databases, besides a subsequent database migration, as regarded in this context. During the design and maintenance of the logical database outline, some arena semantics might not be grabbed anymore. Thus without owning a conceptual view of the database it is difficult for users to realize the semantics of the database and retrieve data correctly. The need for conceptual schemas turns more essential for redesigning subsisting databases when Modern application necessities are considered. Another field of application is database integration, that is best executed at the conceptual degree[BLN86]. Eventually, the semantics of the database should be extracted when switching to another data model. This is required for both article expressions and data migration.

Outline transformation and data migration

Various approaches has been indicated for outlining transformation from relational outline to article oriented ones. Two approaches for database emigration are worth referred. In [AYCD98] an algebraic database migration approach is indicated for that a prototype also exists[AY98]. The focus is on optimizing the migration process and physical or organization of the database. Nevertheless, the outline transformation process does not support flexible transformations. In contrast to this, the approach in [Fah96] exhibits more flexible outline transformations, but the preeminent migration operations cannot be optimized. In particular, various transformation rules are indicated for both the relational and article-oriented model. Furthermore, the instance representing should be applied after each individual outline transformation operation separately, and is expressed on a rather information degree. As a result, the database migration process cannot be optimized. Some different approaches propose useful outline transformations in both, the relational and article-oriented[BP96] context. In relational database design, outline transformations has been employed for reverse engineering[HTJC93b] or quality improvement[BCN92], in order to reduce efficiencies such as de normalization or optimizations. In article-oriented design on different hand, outline transformations has been employed for outline purification[BP96].

The support of outline development is not a primary requirement of these data models. Subsisting algebras, like relational or article-oriented algebra are one manner to describe a conventional foundation. The algebras manner may be employed in different ways: as a conventional semantics of the data and a question language itself. Besides optimization, the necessities for the SOT data model and algebra differ from subsisting approaches. Therefore there are not suitable for outline transformation and data migration, and a Modern approach is presented in this paper.

The primary purpose of SOT data model is the support of outline restructuring and data representing. Outline restructuring is the act of modifying the SOT outline. Data representing alterations the database state, such that the preeminent database is consistent with the modified SOT outline. Outline restructuring and data representing are implemented by algebra. For simplicity, outline and data share the similar algebra. Various grounds exists why neither the relational nor the neither article-oriented nor any different subsisting data model fulfill the necessities of database migration. The primary reason is that the purposes of these data models are different from ours. Most subsisting algebras serve as theoretical foundation to describe the conventional semantics of data and of a question language.

More precisely, the relational model lacks the individuality aspect and the support of typical structures. Individuality is simulated through fundamental properties whose uniqueness should be maintained by the user. As regard typical structures, aggregates and sets may be simulated by additional relationships, but list or array structures cannot be expressed instantly. On the different hand, the article-oriented model is too restrictive in article individuality and inheritance handling. Outline restructuring cannot easily be propagated to the data degree. The interface of an article or its class membership (generally) cannot be changed once it is made. As regards article individuality, there may be no external influence on creating Modern article identifiers as will be demonstrated afterward. Concerning cardinality, the “not null” and “candidate fundamental” restrictions, known from the relational data model, do not exist.

Conceptual model like the ER model, that are regarded. The migration frame job, the primary originality of the transformation process lies in the ability for flexible outline redesign. Input of the transformation process is a relational outline and a set of tables. On the article-oriented side, a outline expressed in the ODMG interface notation and data expressed in the ODMG article interchange format (OIF) is targeted. This manner the approach is not dependent on the concrete target DBMS.

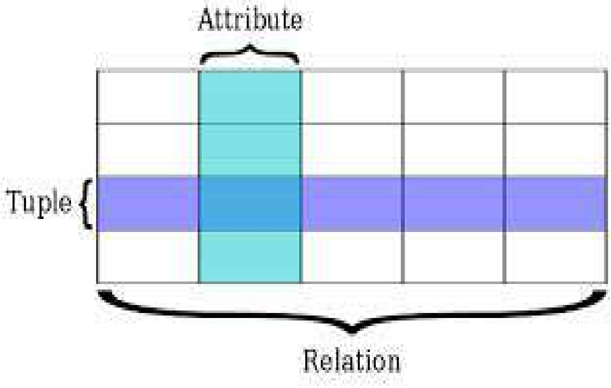

Figure-3:Data Transformation

The data migration process is generated after completing these three steps of outline transformation.

Algebra, described over the initial extensions

Eventually, in the third step SOT model is mapped to a class structure and the article-oriented database is made from the SOT extensions. The target class structure is expressed in the ODMG article resolution language (ODL) [CB00], and the database is made either by creating a migration application or by sorting the SOT extensions in a dump file in the ODMG objectt interchange format (OIF).

Conclusions

Subsisting approaches for migration do not exploit the complete potential of the article-oriented epitome so that the preeminent article-oriented outline still looks rather relational and retains the advantages and disadvantages of the relational outline. Therefore, one of the goals of this approach is to support outline transformation into an decent article-oriented outline s obtained by forward engineering, rigorously utilizing an article-oriented design method. In the initial part of the paper, the fundamental differences amongst relational and article-oriented database are referred. For the effectuation of the database migration process an intermediate data model is indicated that permits defining both, outline transformation and data migration. This data model contains all article-oriented simulating constructs and supports flexible outline transformations. Furthermore, algebra is indicated for conventional resolution of the data migration process.

References

1. Rajamanicakam Leelavathi, International Journal of Engineering And Computer Science 2013;2(8): 2403-2408.
2. Steven E. Minze, “A signaling protocol for Typical Multimedia services.” IEEE Journal on Selected Fields in communications, 1991;9(9):383-1394
3. Stanley L. Moyer, Howard E. Bussey, Paul E. Brierley. “A Connection Management Interface for Fabric Control Application,” in Proc. IEEE International Conference on Communications ICC’93, Geneva, Switzerland, May 23- 26, 1993.
4. Steven E. Minzer, Petros N. Mouchtaris, An article-oriented Resolution for Signaling in Heterogeneous Multimedia networks,” in Proc. Bell core article-oriented Arrangements and Technology Symposium, 1993..
5. Margret A. Ellis, Bjarne Stroustrup. The Annotated C++ Reference Manual, Addison-Wesley, Reading MA, 1990.
6. S.-C (Tom) Soon, P.E. Brierley, et. Al. “Utilizing State machines for article-oriented Simulating,” in Proc, Bellcore article-Oriented Arrangements and Technology Symposium, 1991.

Source of Support: NIL
Conflict of Interest: None