Object Oriented Data Model for Constraint Management

Sahebrao N Shinde
Dept. of Computer Science
K.T.H.M College
Nashik, India

Chitra Desai
Professor and Head,MCA
Marathwada Institute of Technology
Aurangabad, India

Abstract
Object-Oriented Database started becoming popular because of their very powerful features like persistence, support of transactions, simple querying of bulk data, concurrent access, resilience, security. OODBMS were meant for more complex data handling. The inherent complex nature of OODBMS led towards identification and analysis of integrity constraints. In order to better understand the nature of integrity constraints in OODBMS, 31 constraints are studied in light of RDBMS with examples using Oracle. Based on the study of identified constraints we have proposed Object Oriented Data Model which will form an interface between object oriented DBMS and object oriented Programming Language; here in case java programming language. The interface has effectively handled various integrity constraints issues available in the existing OOODBMs thereby giving a solution of constraint management at various levels.

Index Terms
Object oriented data model, integrity constraints, constraint management

1. INTRODUCTION
Every human being today has become dependent on computers for their day to day activities. This is because of the very basic feature of computers that is to store the data and retrieve it for future use. Thus computers have proved to be useful for managing information from a very small piece of data to the complex data seen in large databases. To effectively handle the data various database systems have evolved. The evolution of various database systems generations was the resultant of in depth understanding of the data properties and their representations. The transition from one generation to the next has always been necessitated by the ever-increasing complexity of database applications and the cost of implementing, maintaining, and extending these applications. The first generation was file systems, such as ISAM and VSAM. The second generation was hierarchical database systems, such as IMS and System 2000. The third generation was CODASYL database systems, such as IDS, TOTAL, ADABAS, and IDMS etc. The fourth generation technique is the relational database which evolved overcoming various short comings of its previous generations. However, relational database system also was unable to incorporate the complex data in applications like computer aided design, engineering, software engineering and manufacturing systems and applications that run on them; knowledge based systems; multimedia systems which deal with images, voice and textual documents and programming language systems [1]. In 1980s most of the research went around analyzing the various short comings of the conventional database systems and gave momentum to overlay the importance of the fifth generation of database technology – object oriented database.
A data model is a logical organization of the real-world objects (entities), constraints on them, and relationships among objects. A data model that captures object –oriented concepts is an object oriented data model. An object oriented database is a collection of objects whose behavior and state, and the relationships are defined in accordance with an object oriented data model. An object oriented database system is a database system which allows the definition and manipulation of an object oriented database [1].
The need for object oriented database evolved as a reason for the various short comings with the conventional database system. Here conventional database system refers to relational and past generation database system. Object-oriented database systems evolved from a need to satisfy the demand for a more appropriate representation and modeling of real world entities, so OODBs provide a much richer data model than conventional (relational) databases. The OODB paradigm is based on a number of basic concepts, namely object, identity, class, inheritance, overriding, and late binding [2], [3], [4], [5], [6]. Basically OODBMS were meant for more complex data handling. The inherent complex nature of OODBMS led towards identification and analysis of integrity constraints.
This is a short research paper based on the doctoral research work entitled, "Identification and Analysis of Integrity Constraints in Object Oriented Database System". This paper presents a proposed Object Oriented Data Model which forms an interface between object Oriented DBMS and Object Oriented Programming Language; here in case java programming language. The interface has effectively handled various integrity constraints issues available in the existing OOODBMs thereby giving a solution of constraint management at various levels.

2. INTEGRITY CONSTRAINT
A constraint is an assertion that indicates a restriction and must be satisfied by a correct design of system [7]. In the schema a constraint is a model element instance of the metaclass constraint. The main idea for constraints definition in OO languages consists defining them as operations that are called as constraint operations [7].While defining constraints the challenge is with the heterogeneous data.

Integrating heterogeneous data sources is a fundamental problem in databases, which has been studied extensively in the last two decades. Recently, due to the need to integrate data sources on the web, much of the research on data integration. Data integration is the problem of combining the data residing at different sources, and providing the user with a unified view of these data [8].

A typical error occurs while entering data. The way to make sure that database modifications do not allow inappropriate tuples in relations is to write application programs so every insertion, deletion and update command has associated with it the check necessary to assure correctness. The correctness requirements are frequently complex and they are always repetitive; application programs must make the same tests after every modification [9].

Constraint allows defining certain data requirements that the data in the database needs to meet. If the user tries to insert data that doesn't meet the requirements there has to be some kind of constraint. An important kind of constraint on a database schema is that values for certain attributes must make sense [10]. SQL provides a variety of techniques for expressing integrity constraints as part of the database schema. The key constraints

are where an attribute or set of attributes is declared to be a key for a relation.

Another form is referential integrity called foreign-key constraints, which are the requirement that a value in an attribute or attributes of one relation must also appear as a value in an attribute or attributes of another relation [11]. If constraints are enforced data will be integrated (consistent and accurate) i.e. the reason constraints are called data integrity constraints. Integrity constraints refer to the expression of integrity validity and do not include the enforcement or the maintenance [10]. The term integrity covers consistency and validity. The important kind of constraint in a database is a declaration that certain attribute or set of attributes forms a key for relation [9].

3. FACTORS AFFECTING INTEGRITY CONSTRAINTS

In last 25 years huge set of work has been carried out in the area of Object Oriented Database constraints. The problem which frequently occurs in the design of object oriented database is the integration of integrity constraints into the database system. Most of the past solutions to the integration problem are system specific since the set of supported constraints varies from system to system. The consistency of an object oriented database depends upon the state of participating objects. Certain rules are therefore required to restrict possible state of the database modeled by so called integrity constraints. The problem of integrity constraints in object oriented database comes from external constraints, i.e., that can neither be specified explicitly nor implicitly. To study the issues related to integrity constrains factors listed in Table 1 were considered.

Table 1: Factors affecting integrity constraints

Use of code for imposing simple restrictions	Indexing is not available	Optimization
No complex restrictions	Lack of parallelism	Lack of compression
Joins without optimization	Backups are not possible	Lack of flashback features
Row chaining	Fragmentation	Transactions
Indexing is not available	Security	View
Multiple user data control	Clustering	No data loss
No separate memory	Partitioning	Lack of triggers
Lack of different processes	Authorization	Data concurrency and thread
Lack of nested queries and compound queries	Configuring different block sizes in OODBMS	Lack of different methods of accessing the object data
Lack of parallelism	More burden on RAM	Data concurrency and consistency
Restrictions on the system resources		

On analyzing integrity constraint issues related to above factors to overcome the several limitations of integrity constraints with object oriented database a new object oriented data model has been proposed in the next section.

4. PROPOSED OO DATA MODEL

In Object Oriented databases the user interface for the system can be designed using following two approaches.

- By defining database language to be embedded in cost programming language.
- By extending object oriented programming languages with database repeated constructs.

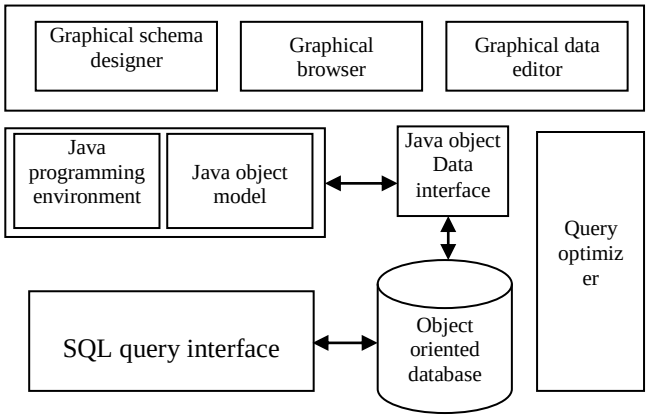

Fig. 1: Proposed OO Data Model

The Programmatic interface to an object oriented database must include:

- A data definition sublanguage
- A query and data manipulation sublanguage.
- A data control Sublanguage

After identification and analysis of the integrity constraints related to object oriented databases in the previous chapter, we have present an object oriented data model in which sublanguages are designed to reflect constraints inherent with object oriented database by seamlessly integrating object-oriented programming language and database language which is consistent with the data message passing paradigm of the system. Integrity constraints express information that is not directly derivable from the database data. If one or more integrity constraints are not satisfied, then the database may go in inconsistent state. In object oriented database when multiple objects interact, resulting into the state change, the possibility of inconsistency in the database cannot be avoided. The proposed object oriented data model must only tracks the inconsistency in the object oriented database due to integrity constraint, but also provides a repair mechanism for it. The repair mechanism used here based on insertion and deletion on tuples. The following are the features of the new proposed Object Oriented Data Model:

1. As the Proposed Object Oriented Database System is designed for general purpose use of object oriented database page server architecture instead of query server model. Query server model are more appropriate for scientific database.

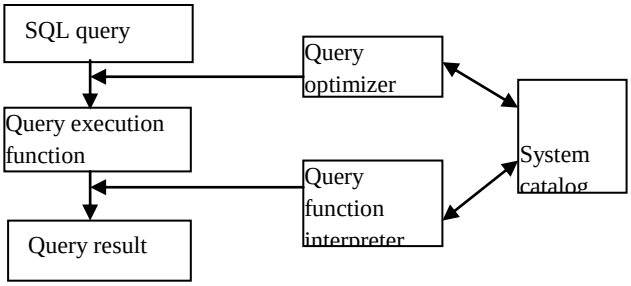

Fig. 2: Query Evaluation

2. It provides embedded SQL features by embedding SQL with Java programming language. Here we have adopted SQLJ features rather then using JDBC.SQLJ differs from JDBC in that SQL statements can be embedded directly into the java Programs and translational time semantic checks can be performed .To include SQLJ following Classes are imported :java.sql.*,sqlj.runtime.* and sqlj.runtime.ref.*.In addition to establishment the default connection to oracle, the oracle class from the oracle.sqlj.runtime.* package is required. Then we connect to the database. The default connection to the database. Once the default connection has been establish SQL statements can be enabled within the java program query:#SQL{< Sql.statements>}.
3. The graphical Schema designer allows visualizing a schema set at different levels of abstraction.
4. The graphical browser allows browsing and navigating the schema tree and performs searches.
5. The graphical Schema designer is integrated with the Graphical editor. If a file is edited in the editor. The changes are reflected in the schema browser.
6. The basic designed for the proposed Model is meant for checking integrity constraints and in order to achieve it affectively a debugger involving role designing is added.
7. Persistent is provided at the level of object rather than at class.
8. As the queries in OODBMS are complex a query evaluation plan is essential. Therefore the proposed model importantly includes the query optimized. The query evaluation takes place using the following process.
9. Indexes are essential components in database system to speed up the evaluation of quires. Because of the nested and hierarchical structure of objects, It is more complicated to apply indexing on OODBMS. Authorization problems occur while indexing on classes.

5. COMPARATIVE ANALYSIS

Table 2: comparative analysis of the proposed model

Sl. No	Features	Object store	ONTOS	Proposed OODBMS	
1	Page server architecture	√	√	√	
2	SQL like interface	---	√	√	
3	Graphical schema designer	√	√	√	
4	Graphical browser	√	√	√	
5	Graphical editor	---	√	√	
6	Debugger	√	---	√	√
7	Java interface	---	---	√	
8	Query optimization	√	---	√	
9	Dynamic add new class	---	√	√	
10	Persistence of level of objects rather than at class level	√	---	√	
11	Indexing	---	√	√	

6. CONCLUSION

The proposed model has several features to best suit the needs of commercial Object oriented database users. It makes use of page server architecture instead of query server architecture. Embedded SQL is used with java programming environment make the interface easy and overcoming the processor overhead. The graphical schema designer, browser and editor give easy to use environment. The effectiveness of integrity constraint is best routed through the inclusion of debugger in the system. Query optimization is the added feature. It also supports dynamic adding of new classes with the added facility of persistence at the level of objects rather than at the class level. The indexing is essentially included to speed up the process of evaluation of queries.

Thus the proposed model overcomes most of the problems addressed here related to integrity constraints of object oriented databases.

7. REFERENCES

[1] Won Kim “Introduction to Object-Oriented Databases”, The MIT press 1990.

[2] Atkinson, M., et. al., “The object-oriented database system manifesto,” in Proc.Int. Conf. On Deductive and Object-Oriented Databases, 1989.

[3] Bancilhon, F., “Object Oriented database systems,” in Proc. 7th ACM SIGART/SIGMOD Conf., 1988.

[4] Kim, W., “A foundation for object-oriented databases”, MCC Tech. Rep., N.ACA-ST-248-88, Aug. 1988.

[5] Stefik, M. and Bobrow, D.G, “Object-oriented programming: Themes and variations,” The AI Mag., Jan 1986.

[6] Zdonik, S.B. and Maier, D eds.," Readings in Object-Oriented Database Systems", San Mateo, CA: Morgan Kauffman, 1989.

[7] Antoni Olive "A Method for the definition of integrity constraints in object-oriented conceptual modeling languages", Elasevier Data and Knowledge Engineering 59 (2006) 559-575.

[8] Anrea Cali, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini "Data integration under integrity constraints" Elsevier Information Systems 29 (2004) 147-163.

[9] Kevin Loney "Oracle 8 DBA Handbook", Oracle Press Second reprint 1999.

[10] Belal Zaqaibeh, Essam Al Daoud "The Constraints of Object-Oriented Databases", Int. J. Problems Compt. Math., Vol. 1, No. 1, June 2008.

[11] Hector Garcia-Molina, Jeffry D. Ullman, Jennifer Widom "Database Systems: The Complete Book", Pearson Publication Six Impression 2011