ABHIYANTRIKI
An International Journal of Engineering & Technology
(A Peer Reviewed & Indexed Journal)
Vol. 2, No. 12 (December, 2015) http://www.aijet.in/ eISSN: 2394-627X

Approach of Object Oriented Technologies in Network DBMS

Bhavya R.*
PG Scholar Software Engineering
Deptt. of Information Science & Engineering
New Horizon College of Engineering
Bangalore, India

Lokesh M. R.
Assistant Professor
Deptt. of Information Science & Engineering
New Horizon College of Engineering Bangalore, India

Abstract

Object oriented technology with a state of the feature for processing information system and long surviving system. On the other side, the network database management system (NDBMS) is the product of IBM Corporation which supports all types of relationships and data is represented in the form of nodes and links. The NDBMS developed by Conference on DATA SYSTEM language (CODASYL) in 1971. It is a most efficient and active database management system but the implementation complexity has drastically reduced realizable features. Both are having complementing characteristics of each other. It is used to develop robust object database management system and it acts like establishment of compatibility amongst model elements of OOT and NDBMS. This will show the satisfied methods of two paradigms. Use this two concept to map the model elements viz. class to record types, inter-relationships such as association, composition and aggregation to set types, inheritance record types, this is develop the cost of introduce that subclasses are non-overlapping.

Keywords*: Relationship between Class, Attribute, Association, Generalization, Aggregation, Record Type, Set Type.*

****Author for correspondence*** bhavyagowda060@gmail.com

1. Introduction

Object oriented technology (OOT) as a long surviving technology with state of the art features for processing information system. Data and procedures are encapsulated together and more relevant than procedure and the NDBMS for the engine to move the OOT train. We have made an attempt to wimp the complexities of NDBMS by the features of OOT. The basic structure of OOT is class. Since it is a very natural and also closer to human perspective. It will survive for long and object is a cohesive computational entity set of related data and related services and state information. This property of OOT exhibits the naturalness of OOT [3]. The OOT is passive; it will require an object oriented language to implement it. Presently have many programming languages and data base available to implement OOT concepts like java++, C#, ORDBMS, OQL etc. but they have to failed to implement all the features of OOT, for example multiple inheritance. But it will fail to implement it properly and there is no proper

implementations of many to many association i.e. they have failed to completely implement well defined interrelationships of OOT [2, 3].

Now a days, the network database management system (NDBMS) is considered to be legacy system. It is a complexity and has lost its popularity. Currently the Relational DBMS and Object DBMS have replaced the NDBMS. The NDBMS is the product of IBM Corporation which supports all types of relationships {one to one, one to many, many to many} and data is represented in the form of nodes and links [1]. The NDBMS is also a set of features which is maintained by IDMS (INTEGRATED DATABASE MANAGEMENT SYSTEM) [4] software which implements the feature of the NDBMS. The NDBMS is a type of DBMS system that supports the network data model. Example: IBM's IDMS, mainly used on mainframe systems. A DBMS is said to be network DBMS when it organizes the data in a network structure. Even though the NDBMS is said to be out dated one, still it is using in many of the organizations like LIC, Indian Army Management System etc. Except its complexities the NDBMS have many good features. The NDBMS is robust & most efficient DBMS. It is built on good database design principles. The many-to-many association is not implemented in any of the DBMS directly using database keys except NDBMS. The areas or realm of NDBMS increases the reusability of record types. The major feature of NDBMS is its owner member relationship. If the member records need to be accessed, it should be accesses through the owner record. This is one of the reasons for which the NDBMS is not so popular. Since NDBMS is developed by CODASYL (Conference on Data System Language), the representation is similar to the COBOL representation [2, 3].

2. Problem Formulation

The OOT is developed with naturalness and sound mathematical logic. The definition of various view elements incorporate good date base design principles. It is an art of technology and legacy system. The OOT is believed to survive long years to come. The OOT features viz. class, object, state, association, generalization/specialization, aggregation, object methods, visibility, and signature are the state of the art features culminating to the state of the art band vegan. The derived features like multiple inheritance, polymorphism, resolution of synonyms & homonyms, use case, all UML diagrams are all enshrine the naturalness in business processing. Unfortunately, OOT is passive i.e. state of the features without active engine [2].

On the other side, the NDBMS developed by CODASYL, which is most efficient i.e., it is active. The NDBMS though it is more efficient for very large databases. It has appreciable features like the record type, set type, realm and location modes, navigation, retention criteria, and authorization through schema and subschema definitions. The NDBMS is more complicated most of the OOT features have analogy with NDBMS features. Both OOT and NDBMS are complement to each other with respect to their facility and implementation. Therefore the blending of both together will form a near complete NDBMS [1, 3]. The NDBMS, though it is most efficient for large database, could not become popular on par with RDBMS because of three reasons: One, the navigation to access a record occurrence has become unduly complex, second is business process during those days could not envisaged very large databases, third, the NDBMS features were developed when there is no awareness for their utility in business circle. Therefore there is a need to super impose of OOT and NDBMS features one over the other to create. But some of the NDBMS features, though they are NDBMS feature cannot be

implemented in the super imposition [2]. The NDBMS is using CODASYL approach were performance and ability to represent complex data relationship. The complexities involved in NDBMS are, one is representation of many-to-many relationship, second is all member records are accessed through its owner record occurrence and so on [1, 10]. The NDBMS is using some addressing modes [1] are an aspect of the instruction set architecture in most central processing unit (CPU) designs. The various addressing modes that are defined in a given instruction set architecture define how machine language instructions in that architecture identify the operands of each instruction. Addressing modes specifies to calculate the effective memory address of an operand by using information held in registers or constants contained within a machine instruction. In computer programming, addressing modes are primarily of interest to compiler writers and to those who write code directly in language. We are using some addressing modes like absolute or direct modes, PC-relative modes; register indirect modes, sequential addressing modes, Immediate/literal modes, auto increment indirect mode, and random modes [1].

3. Literature Survey

CODASYL is an acronym for "Conference on Data Systems Languages-An organization founded in 1959 by the U.S. Department of Defense [10]. It evolved into a variety of volunteer committees and ultimately disbanded by the mid-1990s. CODASYL was widely known for its definition of COBOL, but it was also involved with the network database model and the data description language (DDL) for defining database schemas. The development of a standard programming language that could be used on many computers, effort led to the development of COBOL and other standards. CODASYL approaches were performance and ability to represent complex data relationships. CODASYL DBMS is designed for databases of all levels of complexity, ranging from simple hierarchies to sophisticated networks with multilevel relationships. CODASYL DBMS provides a reliable operating platform for application environments where stability, high availability, and throughput are essential [1, 10]. CODASYL DBMS provides options supporting very large memory addressing and hot standby for performance and availability. The subgroups of DBTG & DDL is a Network Data Model (NDM) was proposed by the Data Base Task Group (DBTG) of the Programming Language Committee (subsequently renamed the COBOL committee) Network Data Model of the "Conference on Data Systems Language". The Network Data Model is also known as the "CODASYL Data Model" or sometimes as the "DBTG Data Model". The DBTG final report was produced in 1971. The DBTG report contained proposals for three distinct database languages: one is a schema data description language; second is a subschema data description language; third is a data manipulation language [1, 10]. The Data Description Language (DDL) is a collection of statements for the description of data structure types. For the network data model, the main data structure types are Record types and Set type [5].

In October 1969 the DBTG published its first language specifications for the network database model which became generally known as the CODASYL Data Model. This specification in fact defined several separate languages:

a. A data definition language (DDL) to define the schema of the database, another DDL to create one or more sub-schemas defining application views of the database; and
b. A data manipulation language (DML) defining verbs for embedding in the COBOL programming language to request and update data in the database. Although the work

was focused on COBOL, the idea of a host-language independent database was starting to emerge, prompted by IBM's advocacy of PL/I as a COBOL replacement.

The first CODASYL DBTG specifications are issued in 1971 largely in response to the need for programming language independence, the work was reorganized: development of the Data Description Language was continued by the Data Description Language Committee, while the COBOL DML was taken over by the COBOL language committee. With hindsight, this split had unfortunate consequences. The two groups never quite managed to synchronize their specifications, leaving vendors to patch up the differences. The inevitable consequence was a lack of interoperability among implementations [10]. Computer associates, which as of 2007 still and supports a version of IDMS [4] are Cullinet, originally known as Cullinet Database Systems, obtained the technology from B. F. Goodrich. Cullinet was eventually sold to Computer Associates, which as of 2007 still sells and supports a version of IDMS. ANSI and ISO adopted the CODASYL database specifications under the name Network Database Language (NDL), with work taking place within the same working group as SQL standardization. Some of the CODASYL committees continue their work today, but CODASYL itself no longer exists. The records of CODASYL were donated to the Charles Babbage Institute and a catalog may be found at its website. Interest in CODASYL gradually faded due to growing interest in relational databases beginning in the early 1980s [4].

4. Design of OOT with NDBMS Techniques

a. Mapping OOT Features to Compatible NDBMS

We consider the abstraction of subset of the model elements responsible for representing the structure part of the object. Further, these abstract models are mapped to NDBMS acceptable features of the model elements. The process of implementing the corresponding OO features in an object network approach [2].

b. Mapping Class to Record Type

The class is based on the entity definition, and record type is NDBMS also represents the entity. The entity normalized structure is object structure of record type. The data items of class are mapped to the attributes of the class. And record type is using good database design principles. It's a storage location modes are also specified for the speedy retrieval from the database [2].

c. Mapping Association to Set Type

The association is reach-ability between the related object classes in NDBMS. The association is represent the set type. The reach-ability made through these represents by set type. The reach-ability made through these sets occurrences. An association is implement is implemented directly as set types of (one: many) in which class on many side is represented as the owner and class on many side is represented as member of the set type [2].

d. Mapping Composition to Set Type

The Composition is also called as whole-part relation set type. The whole class is represents by owner record type and part class is represented by member record type. When whole type is activated then part is also activated. If in case whole is deleted then part is also deleted. To facilitate better access, the location mode of part record type is defined as via set type [2].

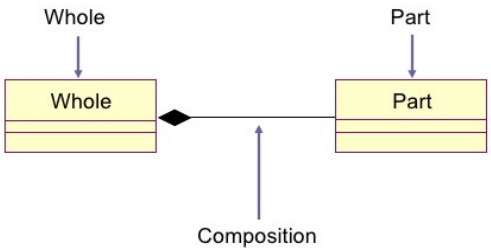

Fig. 1: Composition

e. Mapping Aggregation to Set Type

The aggregation is a form of association, has a type of relation similar to composition but part of the independent objects. This is an instant of many to many relationships. One set of owner by an aggregated record type and other set type of owned by part record type. Using array point to dump all the details of record set types, the semantics of array type is using easily [2].

f. Mapping Generalization/Specification

This is also called as inheritance concept. The inheritance tree is represents by single record type by grouping the attribute in the hierarchical object classes from root node to leaf nodes in higher level numbers in schema and subschema definitions. The overlapping attributes of the subclasses can be represented by the same level on par with redefines clause immediately prior to redefines clause. This server serves the purpose of representing super-class sub-class hierarchy but at the cost of increase in memory space. This technique is superior to single table mapping technique in object-relational mapping, where the inapplicable attributes values are padded with null [2].

g. Mapping Multiple Inheritance

The Multiple inheritance part is use to the owner member relationship between super and subclasses, thus each super class is connected with the set type. The merging of classes is single class that is transformed and represented in a schema as a single record type [2].

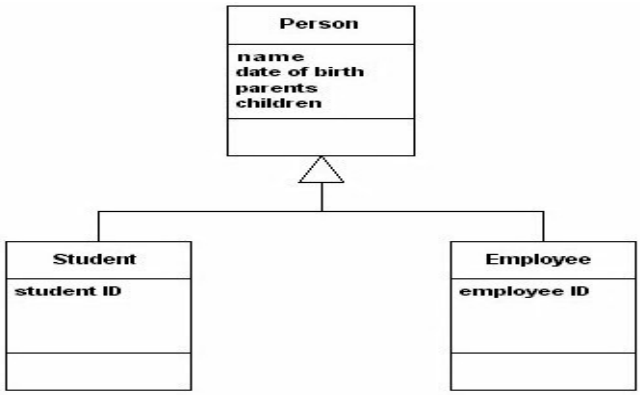

Fig. 2: Inheritance

h. Visibility

The host program is attached to a specific sub-schema. The record type and set type, which are constituent of subschema, are only visible to the host program. Apart from this we can restrict the visibility at record level, attribute level, set type level by using PRIVACY LOCKs defined in the subschema. The visibility of object class is same as record type. The public visibility is record type is incorporated by the presences of required portion of record type description as part of the subschema and private with privacy locks. The record type is restricted to adjacent record types through parameter passing [2].

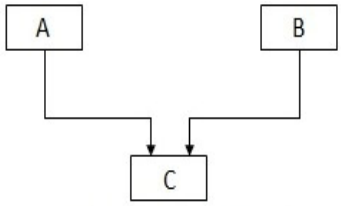

Fig. 2: Multiple Inheritances

5. Comparison of Technology between OOT and NDBMS Techniques

Parameters	OOT	NDBMS
State	Naturalness & Sound mathematical logic	Robust and most efficient DBMS
Design	Design the interrelationship	Design good database principles
Resemble	Provides Hardware engineering	Provides User work area for each user
Principle	Structure and design but have not build on good DB Design	Record type within area has thick and thin relationship between the areas.
Define	Design for semiotic diagrams	Design for schema & subschema diagrams

6. Conclusion

We have proposed DBMS applications through transforming features of OOT into the super highway of NDBMS and the techniques to map some criteria respectively like record type, set type, association, aggregation etc., as well as proposed mapping techniques to realize visibility through privacy locks.

7. Future Work

We would use RDBMS to implement the NDBMS and use multiple inheritances in a clear cut way as it may take multiple routes to the navigation of parent classes. The public visibility and protected visibility; there is a need of redundant records in every subschema and use of virtual attributes respectively.

References

[1] Wikipedia.org/wiki/NDBMS,CODASYL
[2] Shivanand M H, "An Ameliorated Methodology to Establish Compatibility between Structural Parts of Object Oriented Technology and Network DBMS", in proceeding of International Journal of Software Computing and Software Engineering, Vol .3 No .3, 2011.
[3] S. M. Handigund, Kshama S. B., Ranjana N "An Ameliorated Methodology to Design and Develop Work Process Diagram" in International Conference on Science and Technology (SAI-2013), London, UK.
[4] International Computer Ltd "IDMS Data Administration Language", References 1978.
[5] David Marie, Jacob Stein, Allen Otis, Alan Purdy. Development of an Object-Oriented DBMS". OOPSLa'86 as ACM SIGPLAN 21 Nov. 1986.
[6] Michael C. "Next Generation Object Database Standardization", Object Database Technology Working Group White Paper, Sept 2007.
[7] ETH Zurich's lecture "Object-Oriented Databases" by Michael Grossniklaus and Moira C. Norrie. http://www.globis.ethz.ch/education/oodb, Sept 2010.
[8] Object Management Group, "The Object Data Standard: ODMG 3.0, 1999. Morgan Kaufmann Publishers, San Francisco, California.
[9] R. G. G. Cattell, T. R. Rogrss, "Combining Object Oriented and Relational Models of Data", 1986 International Works shop on Object-Oriented Database Systems, Pacific Grove, Calif, and Sept 86.
[10] Robert W Tailor and Randall L F, "CODASYL Database Management System" Computing Surveys, Vol 8, No1, March 1976, ACM.