International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:11, 2007

New Approach for the Modeling and the Implementation of the Object-Relational Databases

Amel Grissa-Touzi, and Minyar Sassi

***Abstract*—**Conception is the primordial part in the realization of a computer system. Several tools have been used to help inventors to describe their software. These tools knew a big success in the relational databases domain since they permit to generate SQL script modeling the database from an Entity/Association model. However, with the evolution of the computer domain, the relational databases proved their limits and object-relational model became used more and more. Tools of present conception don't support all new concepts introduced by this model and the syntax of the SQL3 language. We propose in this paper a tool of help to the conception and implementation of object-relational databases called «NAVIG-TOOLS" that allows the user to generate script modeling its database in SQL3 language. This tool bases itself on the Entity/Association and navigational model for modeling the object-relational databases.

***Keywords*—**Abstract Data Table, Navigational model, Object-relational databases, References.

I. Introduction

AT the end years 90, a new generation of databases management systems called object relational was born. It intends to remedy the weak points of the present systems and to permit a power of modeling importing (semantically rich) and offer all other functionalities of database management system. This system is based on the extended relational model to the main concepts of object-oriented model (extensibility of the set of type, complex structures, object identity, inheritance and methods) [1]-[2]-[3].

Currently, all big constructors of databases management systems are conscious of limits of the relational model and propose systems of this type. They even participated in the development of the norm of the new version of SQL, SQL3. As example of databases management systems, we can mention Informix, Postgres, oracle [1]-[2].

Several computer solutions have been proposed trying of enrich their tools to can modelize the object-oriented concept as the inheritance, classes etc. Among these tools, we illustrate the Win'Design[4], PowerAMC[4], Rational Rose[4]-[5], Designer [4]and JDevelopper [4]-[6] etc. But these tools remain limited to some concepts always. Indeed, these tools leave always is the E/A diagram in the case of PowerAMC, Designer, either of the class diagram in the case of Rational Rose for modeling the object-relational databases or object-oriented databases.

In 2000, SOUTOU introduced in [7] a model called navigational model that permits of modeling all new object-relational database concepts and more precisely the new concepts introduced with the SQL3 language. SOUTOU has specify in [1] rules of passage of this model toward the UML class diagram and toward the language SQL3. He showed the possible choice number that can have an object-relational database inventor to pass navigational model toward the implementation of the basis with SQL3 in addition. This model will be primordial in the conception of the object-relational databases.

We present in this paper a tool of conception and modeling of object-relational databases, based on the navigational model, called NAVIG_TOOLS. This tool gives to the inventor the possibility to draw the E/A model, to help him to define his navigational model of his object-relational database and to deduct the corresponding script of it in language SQL3, while using rules of passage proposed by SOUTOU [7]. We have tried to adopt the same interfacings of PowerAMC, so that users who familiarize already find again quickly.

The remainder of paper is organized in six sections. Section II presents object-relational databases technologies. Section III presents conception of object-relational databases. Section IV describes the navigational model. Section V is devoted to the description of our tool called NAVIG-TOOLS. The balance of this work and its future perspectives are discussed in Section VI.

II. Object-Relational Technologies

The object-relational technology was born in 1992, it is therefore new enough on the market of database management systems. Currently, all big publishers of database management systems are conscious of limits of the relational model. They saw therefore in the object-relational model a manner to enrich this model. Informix was the first publisher of management systems to raise the challenge of the object-relational as purchasing the business illustrated, that developed a commercial version of the Postgres system, an

Manuscript received Mai 20, 2005.

A Grissa-Touzi M. Sassi, with the Department of Electrical Engineering of National School of Engineers of Tunis, Tunisia (e-mail: amel.touzi@ enit.rnu.tn, sassi_minyar@yahoo.fr).

extended relational management system. Others big publishers of management systems, as IBM and Oracle, followed this evolution [1]-[2]-[7]-[8].

A. Object-relational model principle

The object-relational model is founded on the idea of extension of the relational model with object concepts.

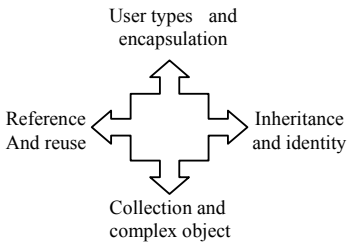

Fig.1 Relational model extension

An object-relational database is an extension of a relational database. It unites the key concepts of the object to facilitate the integration of the two models. An object-relational SGBD is a relational SGBD which will be added object functionalities as the object identity, the encapsulation, the inheritance of types and the support of complex objects. This is illustrated in Fig.1.

B. Essential Additional Concepts

We present in this section, additional concepts of object-relational databases:

- User data types: user definite data type composed of a data structure and operations encapsulates this structure.

The types system of the database management database becomes expandable and is not limited to the alphanumeric types of basis anymore. In this context, user must define its types and to adapt those to the management of information. It becomes possible to define the picture types, point, vector, etc. They are called Abstract Data Types (ADT).

- Collection templates: generic data type permitting to support attributes multi-valuated and to organize them according to a collection type.

Bosses of lists permit to define some new types on the basis of other type collections. They can use the different types of collections of the object-relational model, as table, list, set and dynamic tables.

- Reference of object: Particular data type permitting to memorize the invariant address of an object or a tuple. References are identifiers of object (OID) of the object-relational model. They represent the unique and invariant values permitting of chaining objects between them without passing by joints by values [6]-[7]-[8]-[9].

- Inheritance of type: Shape of inheritance implying the possibility to define an under-type of an existing type, this one inheriting the structure and the possible operations of the basis type.

- Inheritance of table: Shape of inheritance implying the possibility to define an under-table from existing tables, this one inheriting the structure and the possible operations of the basis table.

III. Object-Realational Databases Conception

Methods of classic conception of relational databases recommend to pass from E/A model (conceptual level) to a normalized relational model (logical level), then to link on an optimized relational model (internal level). Views are finally declared (external level) according to prerogatives of users of the data base. For the classic relational model, the passage between every level makes them while respecting of precise rules.

Conception process of an object-relational database, described in Fig.2 inspires itself of the process of conception of a relational database [7].

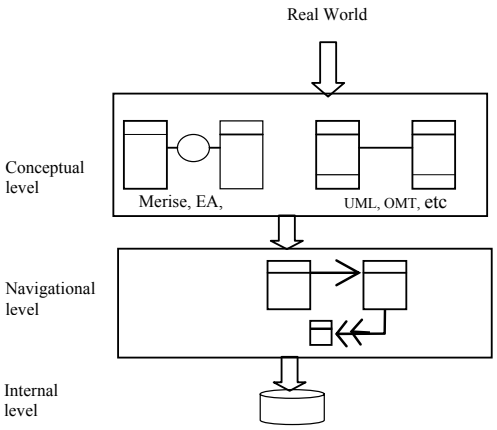

Fig.2 Modeling process

Process must follow the three steps:

- The modeling of the real world according to a semantic E/A or UML formalism (conceptual level).
- The transposition of the conceptual diagram in a navigational model (logical level).
- The transposition of the navigational diagram in types and in object-relational tables of the object-relational databases management systems (internal level).

Until now, it doesn't exist software allowing the inventor to draw his E/A model or UML or navigational. Otherwise the generation of the corresponding script in SQL3 language.

IV. Navigational Model

A. Presentation of the Navigational Model

Navigational term is used because this model includes drawers who facilitate the navigation between objects of the database. This model also proposes some notions as the types of wholes, structures and references. The navigational model plays a pivot role between the conceptual model and the physical model. It facilitates the passage of an object conceptual model to a model of object-relational database and limits the risks of incoherence. Indeed, object approaches permits to represent the complex data directly without necessary to normalize this information, but it is impossible to eliminate the problems caused by the non normalized objects.

The normalization is not a specific concept to the relational systems, it is susceptible to concern all the structures of data.

The navigational model can be considered like a theoretical support facilitating the passage of a conceptual model to an object-relational database model while limiting the risks of incoherence.

The abstract data types don't appear explicitly to the conceptual level. They are contained implicitly in the declaration of the entities (E/A) or of the classes (UML). Those types are clarified more in the navigational model.

B. Formalism of navigational diagram

The formalism of navigational diagram is very near to the formalism of Object Database Management Group (ODMG), to which has been added the notions of identifying, of structure and intra-classes ties. The navigational model formalism permits to represent the static and dynamic aspects of classes.

-Definitions: Classes contains some attributes (properties) and signatures of methods (functions or procedures).
Attributes can be simples (string, number,…), struct, references (drawers), set (collection of structures or references) or of the previous type combinations.
The ties can exist inside a same classes (intra-class) or between two classes (inter-class).

-Choice of the graphic representation of the ties: in a navigational diagram, mono-valuated ties are represented by arrows and multi-valuated ties by arrows with duplicate tip.

Example:

The database organization has for goal to manage employees, departments of an enterprise.

An employee is described by identified Nas, a Name, an Adresse composed by Rue, Cp and City and a dependant described by last name and Sex. A department is described by identified number and name. A department is always managed by only one employee.

The E/A model of this database (presented in Fig.3) contains two classes: the Employee class described by a simple attributes Nas and name, a composed attribute adresse formed of three fields (Rue, CP, Ville) and a composed multi-valuated attribute described by (Prenom and Sex). The class Department described by simple attributes No, Nom and Responsable.

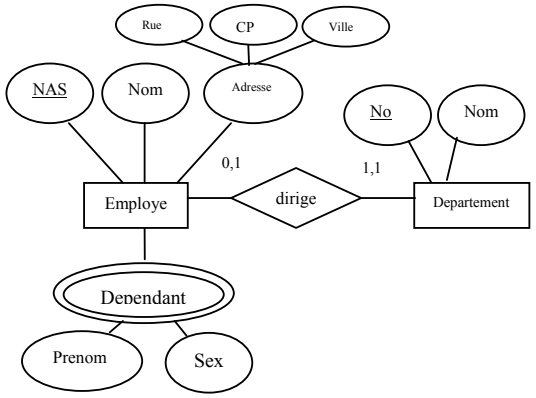

Fig.3 E/A model

The passage to the navigational model presented in Fig.4.

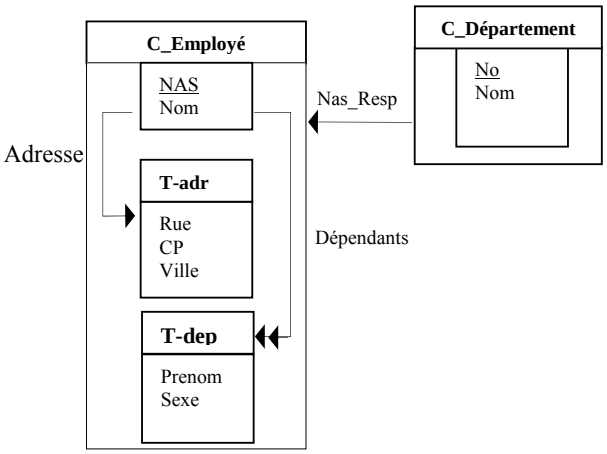

Fig.4 Navigational model

As the shows the diagram, we can modelize comfortably the composed attributes, multi-valuated attributes and drawers.

V. NAVIG-TOOLS

In order to help understanding the stratification concept, we have built the tool NAVIG-TOOLS. This tool is plate-form independent since it is developed with java. We chose Oracle9i like database management system.

NAVIG-TOOLS offer a convivial interface to the inventors of the object-relational databases that allow him:

-Modeling its conceptual Model in E/A and/or navigational model;

-Genering the corresponding SQL3 script;

For every ambiguous steps, NAVIG-TOOLS propose to the inventor several solutions. Indeed, it is due to the fact that to represent a tie one can have several solutions [7].
In our first version of the software we limited ourselves to the types offered by the version oracel 9i, as:

-Scalar Type (Number, Varchar, Date,..)
-Object Type
-Object Reference REF
-Collections (VARRAY and NESTED TABLE)

NAVIG-TOOLS present to the expert a main screen (Fig.5) similar to the standard usually used in the IDE (Integrated Development Environment) and very similar to Power AMC but to which one added new functionalities modeling the new symbols introduced by the navigational model and the new concepts introduced by the object-relational databases concepts and the SQL3 language.

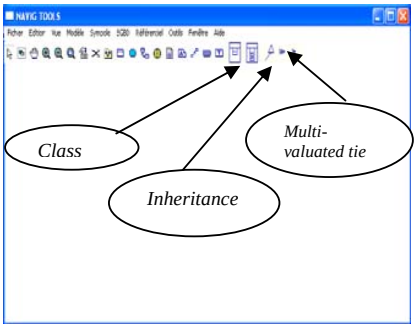

Fig.5 NAVIG-TOOLS Interface

For implanting his database, the inventor must follow the two following steps:

1) Conception of database

The inventor can follow one of the three following methods:

- To draw the E/A model of his database as the shows the Fig.3. NAVIG-TOOLS can deduct the navigational model (Fig. 6). If the user is not satisfied, he can modify this model.

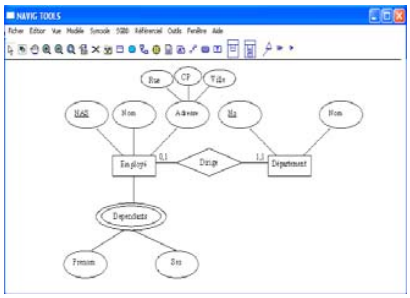

Fig.6 E/A Model

- To draw directly the navigational model of its object-relational database (Fig.7).

2) Generation of SQL3 script

The inventor chosen the database management system menu, the button to generate the SQL3 script will be generated automatically by NAVIG-TOOLS (Fig.8).

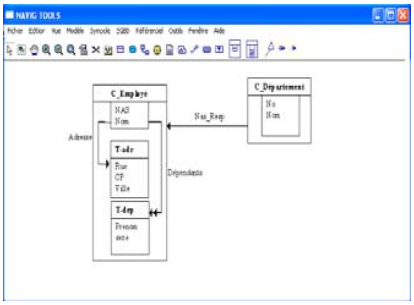

Fig.7 Navigational Model

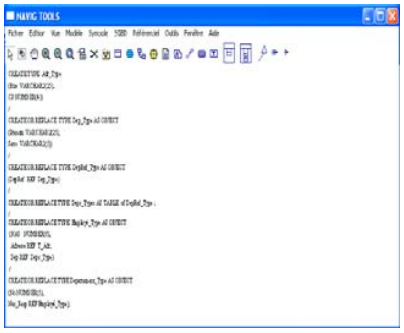

Fig.8 Generation of SQL3 Script

VI. Conclusion

In the next decade, the object-relational databases management systems will very certainly constitute one of the biggest technological progresses in databases domain. Mainly because they are construct on the bases of the relational technologies that gave its proofs since more of twenty years. It is for this reason, which we see that it is very important to think from now on one efficient tool for helping about the conception of object-reltional databases. As several of them exist for the relational databases and supporting the navigational model proposed by SOUTOU that showed its proofs. We proposed in this paper a tool of conception, called NAVIG-TOOLS that bring two main contributions to the conception and the implementation of the object-relational databases. The first improvement is that it offers to the user the possibility to draw the E/A and/or navigational models. The second is that it generated SQL3 script.

This work can be spread while proposing to test NAVIG-TOOLS on big database and to perfect our tools so that it can support the other types offered by the object-relational concepts.

References

[1] P. Daehne, and P. Leutert, "*Object and relational models*", High School specialized of Switzerland Occidental, November 2001.
[2] A. Gamache, "*Book on the databases* ", Department of data processing and software genius, University Laval.
[3] E. Meylan, "*Theoretical introduction to the object-relational model*", High School specialized of Switzerland Occidental, June 2001.
[4] C. SOUTOU, "*Of UML in SQL*" , Eyrolles,2002.
[5] T. Quatrani, "*Modélisation sur Rational Rose*", Eyrolles, 2000.
[6] G. Godin, "*Database management systems by example*", Montréal, Canada, 2003.
[7] C. SOUTOU,"*Object - relational under Oracle 8. Modelling with UML* " Eyrolles, 1999.
[8] R.G.G. Catell., "*Object-Oriented Data Management: Object-Oriented and Extended Relational Database Sytems*", Addison-Wesley Publishing, Inc., 1994.
[9] M. Makowski, "*A structured modeling technology*", European Journal of Operational Research, Vol. 166, Issue 3, 1 November 2005.