МАТЕМАТИКА И МАТЕМАТИЧЕСКО ОБРАЗОВАНИЕ, 1999
MATHEMATICS AND EDUCATION IN MATHEMATICS, 1999
Proceedings of Twenty Eighth Spring Conference of
the Union of Bulgarian Mathematicians
Montana, April 5–8, 1999

AN APPROACH TO KNOWLEDGE INTEGRATION AND BUILDING INTELLIGENT INTERFACE FOR RELATIONAL DATABASE *

Kaloian Tomov Kaloianov, Kalinka Mihailova Kaloianova

This paper presents the three basic components of intelligent relational database interface.
The first of them is the logical object-oriented relational language. The language is a way for data and knowledge integration and it provides simplicity and powerfulness in data queries and manipulations.
The second is the menu-based query interface. There are no difficulties in formulating any query for any users. The basic component here is the knowledge of the links between objects in database. This knowledge also provides opportunities for extracting additional information.
The third is the usage of the stereotype of users that has been dynamically formed in process of work with the system. This is a way for solving some of ambiguities in query evaluation.

1. INTRODUCTION.

Integration of the Artificial intelligence and Database Systems has been the topic of a number of investigations. The results obtained in them are of great importance for introducing of new generation databases that respond the need for knowledge engineering, diverse object description, and database management facilities.

The quality of data definition and data manipulation languages realised in DBMS has always been the subject of extensive scientific research. Fourth generation languages such as SQL, QBE, etc. have great extensive capacity but their use is possible only after special training. The end users of any information systems commonly work with previously designed forms and reports. Doubtless this is a restriction of one's possibility to work with data in a natural way and this decreases the functionality of the systems.

On the other hand the expert systems must posses significant power for dealing with large collections of facts and rules [6]. It is remarkable that the above discussed integration presents such an opportunity.

The existing databases do not give any uniform formal way for knowledge presentation. This requires complicated, and sometimes inadequate tools added to the languages (cf. SQL and QBE).

*This parer is partially supported by project 247/1998-Sofia University

2. LOGICAL OBJECT-ORIENTED RELATIONAL LANGUAGE ILMA+

The integration of logical and relational databases can solve many of the problems arising in the usage of large logical bases. This is a result of the separation of methods for data retrieving in traditional logical base and relational database. In the same time the declarative languages are more expressive and adaptable for query formulation and execution.

The standard form of query in ILMA+ is the following:

```
W(A) :- B1,B2,...,BN;
```

Here W is a procedure with a list of parameters A.

This is a typical Prolog clause [1]. Let us consider the procedural interpretation of Prolog and let W be the performing procedure with the list of parameters A. Then in order to execute W it is necessary to execute successfully the procedures B1,B2,...,BN. ILMA+ is very similar to the relational language Alpha due to Codd [7]. The form of query is simple but not less expressive and powerful than this in Alpha.

We have implemented the following standard predicates for data definition and manipulation.

- creating tables

```
CREATE(<table_name>,
            <attribute1> <type1>,
            <attribute2> <type2>,...,
            <attributeN> <typeN>)
```

- removing tables

```
DROP(<table_name>)
```

- listing rows

```
SHOW(<list_of_attributes>)
```

- inserting rows

```
INSERT(<table_name>, <value1>,
     <value2>,...,<valueN>)
```

- deleting rows

```
DELETE(<table_name>)
```

- updating existing rows

```
UPDATE(<table_name>,<attribute1> <value1>,
<attribute2> <value2>,...,<attributeN> <valueN>)
```

The conditions for data restriction can include get(table) predicate for explicit access to a database, comparison of scalar expressions by means of simple comparison and arithmetic operations, and aggregate functions such as SUM, MIN, MAX, etc.

Object – oriented systems are based on three concepts: object, message, and class. The idea of extension of logical languages like Prolog with object-oriented capabilities is not new. A typical implementation of this concept can be found in [3,4].

In our implementation the class definition has the form:

```
CLASS(<name>, <class>, <arguments>) :-
```

```
<method>; [<method>; ... ]
<arguments> ::= argument1: type1
                [, argument2: type2, ...]
<method>   ::= <name>(<arguments>)
                  WITH <clause>
```

and class instances predicate:

```
INSTANCE(<object>,<class>)
```

It is evident that no multiple inheritances are provided. All methods are virtual in terms of object-oriented programming.

The basic problem in languages like Prolog is their little effectiveness. For this reason ILMA+ is equipped with tools for using indexing system with B-trees and it works with sets of rows instead of the typical for databases method – using of cursors.

3. MENU-BASED INTERFACE

The functional properties of a database system mostly depend on the interface. One of the main goals in the research in the area of human – computer communication is to develop intelligent interface [5].

Our menu – based interface is similar to the relational language QBE. The most essential difference is that it is not necessary to give the links between tables. To know joint tables conditions is often a barrier for end users. Usually they are not familiar with the links between objects presented in the database. Furthermore as a result of the decomposition in process of database design many tables are obtained. This avoids duplication of data and some undesirable relations. But it is not necessary for the end user to know the process of decomposition or the view of programmers and designers of an information system. The user must have the possibility to manipulate data in a natural way according to his or her notion of information. In existing information systems there is the small number of forms and reports and user can work only with them.

The development presented here can be useful for programmers. It can save them the trouble to write a wide collection of programs for different queries. The system does not impose any restrictions on database design.

Of course there exist queries such that the interface can not deal with. This is the case when there exist several links between two tables. For illustration let us see the next simplified example.

Database “Library” consists of the tables:

```
BOOK(CODE#,TITLE,AUTHOR#,EDITION,YEAR)
AUTHOR(AUTHOR#,NAME,COUNTRY#)
READER(READER#,NAME,YEARS,JOB,COUTRY#)
READER_BOOK(READER#,CODE#)
COUNTRY(COUNTRY#,NAME,CONTINENT)
```

The tables BOOK and COUNTRY can be joined in two ways: through the AUTHOR, and through the READER_BOOK. An illustration of this is the queries: “Books from French authors”, and “Books got out from French students”. In first query we must set

only COUNTRY.NAME = 'France' and indicate that we want to receive all books. The system should choose from the two joint conditions

- `BOOK.AUTHOR# = AUTHOR.AUTHOR# and`
  `AUTHOR.COUNTRY# = COUNTRY.COUNTRY#`
- `BOOK.CODE# = READER_BOOK.CODE# and`
  `READER_BOOK.READER# = READER.READER# and`
  `READER.COUNTRY# = COUNTRY.COUNTRY#`

To do this it is possible to ask the user. Another way to solve the same problems is to support and to use the stereotype of the user (SU).

STEREOTYPE OF THE USER.

It is very important for the intelligent interface to know user's view toward data. With SU we note the behaviour typical for each end user. In our development we start from the assumption that many of the information systems were designed for more than one user and every user has specific view on data and manipulation procedures. The user tends to work with a relatively small subset of them.

Our system has knowledge how to join tables. We can connect this knowledge with the concrete user and in this way to decide some of problems in their utilisation.

Another issue is that the system can build SU gradually performing statistics of the user's actions. Later it is possible to create corresponding procedures for the most frequently performed operations.

In the example above these employees in the library that work with the readers will use often the link READER_BOOK for joining the tables READER and COUNTRY.

The key issue in this approach is that the system obtains some tools for self-adaptation according to the users and the needs of information in their usual activities.

REFERENCES

[1] D. Dochev, Dichev H., Markov Z., Agre G., "Programming in Prolog" Sofia, Science and art, 1989.

[2] M. Tsigarovska, M. Sc. Thesis , Sofia University "Sv. Kliment Ohridski", FMI, chair Computer science, 1993.

[3] C. V. Ramamoorthy, P. C. Sheu, "Logic-oriented Object Bases", IEEE CompSac, 1987, 218-225.

[4] C. Zaniolo, "Object-oriented Programming in Prolog", Proc. 4th. Symp. on Logic Programming, 1984, 265-270.

[5] Parsaye K., M. Chignell, S. Khoshafian, H. Wong, "Inteligent Data Bases – Object-Oriented, Deductive, Hipermedia Tehnologies".

[6] V. Sgurev, R. Pavlov, "Expert Systems" Sciense and Art, Sofia 1989

[7] C. J. Date, "An Introduction to Database Systems", 1977,1995 by Addison-Wesley Publishing Company

Kaloian Tomov Kaloianov
Faculty of Mathematics and Informatics
Sofia University
5 James Bourchier
1164 Sofia, Bulgaria
E-mail: kaloiank@fmi.uni-sofia.bg

Kalinka Mihailova Kaloianova
Scientific Research Institute of
Telecomunications
8 Haidushka Poliana
Sofia, Bulgaria
k_kaloianova@hotmail.com

ПОДХОД ЗА ИНТЕГРИРАНЕ НА ЗНАНИЯ И ИЗГРАЖДАНЕ НА ИНТЕЛИГЕНТЕН ИНТЕРФЕЙС В РЕЛАЦИОННИ БАЗИ ОТ ДАННИ

Калоян Томов Калоянов, Калинка Михайлова Калоянова

В настоящата работа са представени три основни компонента, върху които е изграден интелигентен интерфес в релационна база от данни.
На първо място логическият релационен език е основното средство за интегриране на данни и знания от една страна и за по-голяма простота и изразителна мощ при формулиране на заявки към базата от данни.
На второ място базираният на менюта интерфейс е доведен до значителна простота при използването от крайните потребители, като е снабден със знания за връзките между данните. Тази възможност е и удобно средство за извличане на допълнителна информация, свързана с обработваната в момента.
На трето място е изгражданият в процеса на работа на системата стереотип на потребителя и по такъв начин разрешаването на някои неоднозначности при организацията на интерфейса.