An Approach for Response Generation of Restricted Bulgarian Natural Language Queries

Silyan Arsov

Abstract: *The paper presents our researches in formation of methodology for accomplishment database management system with a restricted Bulgarian natural language interface. It is proposed a method for formulation of the query constructions in a restricted Bulgarian natural language, which is based on the elegant theory of the relational algebra. Within the framework of researches it is introduced system tools for formulation of the user's query. It is proposed an algorithm for response generation, which includes semantic analysis of a user's query and method for direct access to data.*

Key words**:** *Query, Response Generation, Question-Answering, Restricted Natural Language Interface to Database.*

INTRODUCTION

Several methods for response generation from the database by queries in a restricted natural language (RNL) exist.

In syntax-based systems as LUNAR [17] the user's question is parsed, and the resulting parse tree is directly mapped to an expression in some database query language. Syntax-based natural language interfaces to databases (NLIDB) usually interface to application-specific database systems that provide database query languages carefully designed to facilitate the mapping from the parse tree to the database query. It is usually difficult to devise mapping rules that will transform directly the parse tree into some expression in a real – life database query language (e.g. SQL).

In semantic grammar systems as IRUS [5], DELPHI [6], and PLANES [16] the question-answering is still done by parsing the input and mapping the parse tree to a database query. The difference in this case, is that the grammar' s do not necessarily correspond to syntactic concepts. There exists systems, as that described in [9], which combine semantically retrieval with other specific methods. Semantic grammars contain hard-wired knowledge about a specific knowledge domain, systems based on this approach are very difficult to port to other knowledge domain. A new semantic grammar has to be written whenever the NLIDB is configured for a new knowledge domain.

When intermediate representation languages are used, then NLIDB first transform the natural language question into an intermediate logical query, expressed in some internal meaning representation language. The intermediate logical query expresses the meaning of the user's question in terms of high-level world concepts which are independent of the database structure. The logical query is then translated to an expression in the database 's query language, and evaluated against the database. The many natural language front-ends as SQUIRREL [4], DATALOG [10], TEAM [12], EUFID [15], EXACT [18], use several intermediate meaning representation languages, not just one.

The specified methods are used for access to relational databases, in which the semantics of the relationships in database is not stored, it makes them inconvenient for direct access when using queries in a RNL. That's why the known systems for natural-language access translate the natural language query in a database language as SQL. It complicates the development of natural language interfaces of this type [8]. The natural language interfaces are considered as dependent on the relational data model.

In the paper, with purpose of a response generation from databases by queries in a RNL, it is proposed a different approach from the known ones thereby. It is used a developed further data model proposed in [2], so that the semantics of the relationships between entities and between entities and attributes are depicted in the conceptual scheme of the database. In defence of this approach it can be indicated [1], where the

data model is depicted as a heart of the interface design. The exploration work and the design of EXODUS system described in [7] are also a witness of the method proposed in [2]. It is used an appropriate internal representation of the database proposed in [3]; It is propounded an appropriate query organization and constructions based on the relational, aggregate and actualization operations.

DEFINITION OF THE PROBLEM

To propose an appropriate method and algorithm for response generation from databases by queries formulated in a restricted Bulgarian natural language.

To use for description of databases the model entity–relationship–attribute, developed further by the authors and proposed in [2].

To use a method for a three-dimensional separated internal representation of the data and relationships for storing the database in the memory as propounded in [3].

To use query constructions for execution of the operations specially-formulated by the authors for formulation of queries as shown in figure 1.

SOLUTION OF THE PROBLEM

For the purposes of researches, query constructions for execution of the basic operations on relations, the basic aggregate operations, the basic actualization operations and their combinations are formulated. The generation scheme of the different operations is shown in figure 1.

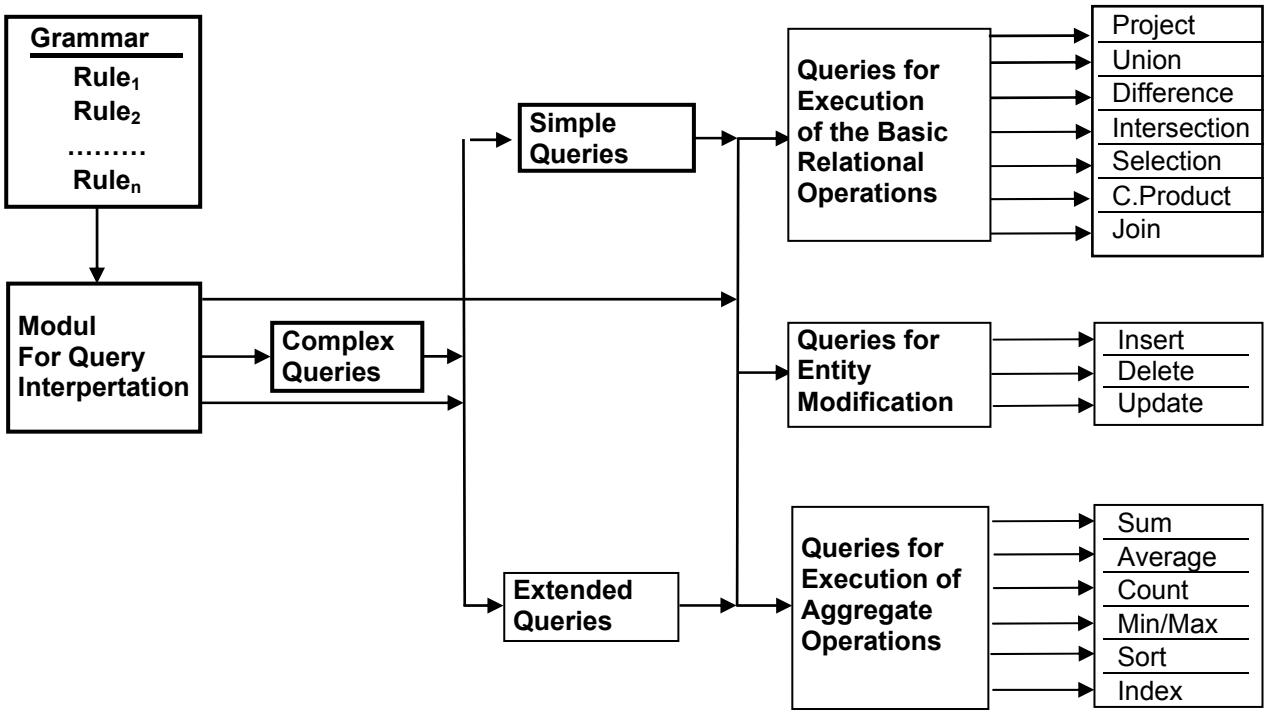

Figure 1. Generation scheme of restricted Bulgarian natural language query constructions

The operations and their combinations, which are included in different query constructions and their corresponding grammar rules of the language, shown in figure 1, are described below. The queries for execution of the basic relational operations execute one of the following operations: projection, union, difference, intersection, selection, cartesian product, and join operation. The queries, which execute operations for entity modifications, execute one of the following operations: insert, delete, and update operation. The query constructions, which execute the aggregate operations execute one

of the following operations: sum, average, count, minimum/maximum attribute value, sorting of data, and index creation operation. The simple query constructions are two types: a simple query with value aswer and a simple query with alternative answer. They can combine the relational, aggregate and modifications operations as shown in figure 1. The extended query constructions are as follows- an extended query with value answer and an extended query with alternative answer. The complex queries combine two or more

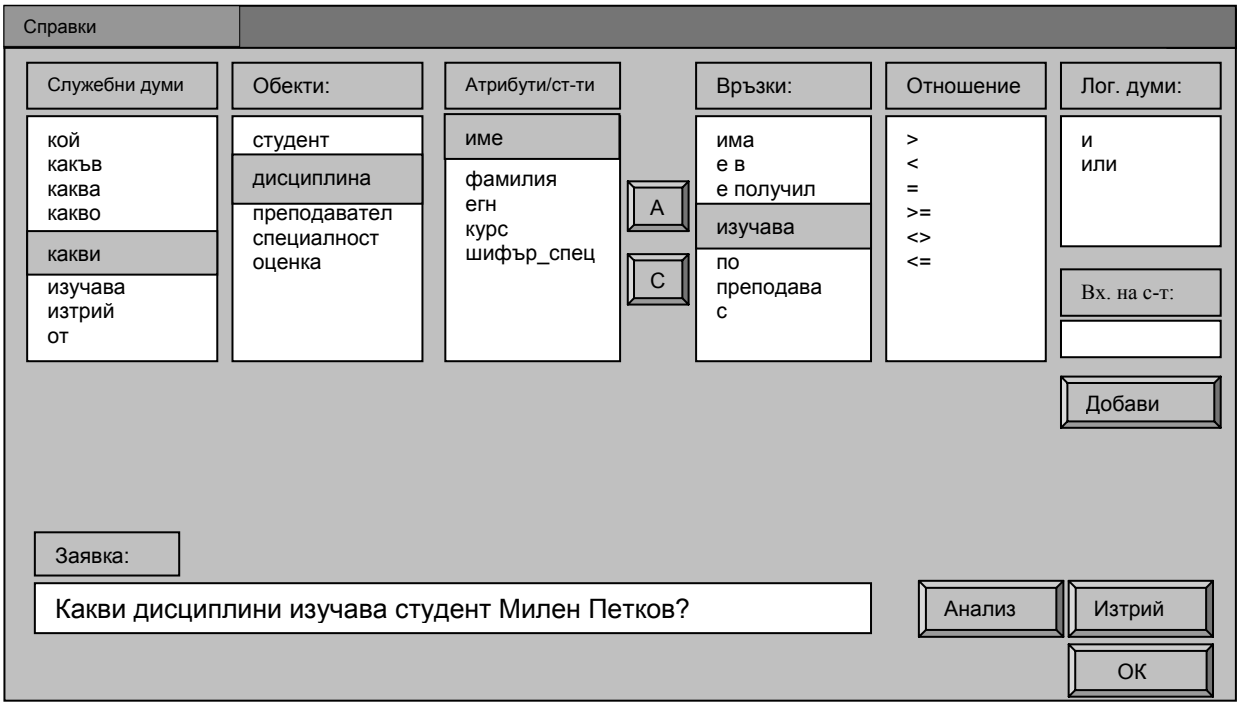

Figure 2. Menu based graphical dialogue panel for query formulation

simple or extended queries joined with "and" or "or" conjunction words.

An interpreter of queries in a RNL is constructed with purpose to be possible for the system to reply adequately to the formulated queries.

Table 1. Non-terminal symbols

Semantic forms	Codes
Key words	QUEST
Entities	ENTITY
Attributes	ATTRIB
Relationships	RELSH
Ratio	RATIO
Logical words	LOGREL
Attr. Values	ATTRVAL

Due to the input query formulation is executed by choosing words from the list boxes at the menu based graphical dialogue panel presented in figure 2, then there is no risk of making a lexical error. The symbols, which participate in the query, are separated in a symbol table. The symbol table is created at the time when the query is formulated.

Depending on their semantics, the non-terminal symbols are divided as shown in table 1.

After the user formulates a query from the menu based graphical dialogue panel, shown in figure 2, the system management is handed over to the sub-system for analysis, which executes the query analysis and generates a response.

The module of query analysis checks whether the query corresponds to one of the grammar rules.

When it is found a rule, which corresponds to the query then the management is handed over to the corresponding procedure for response generation.

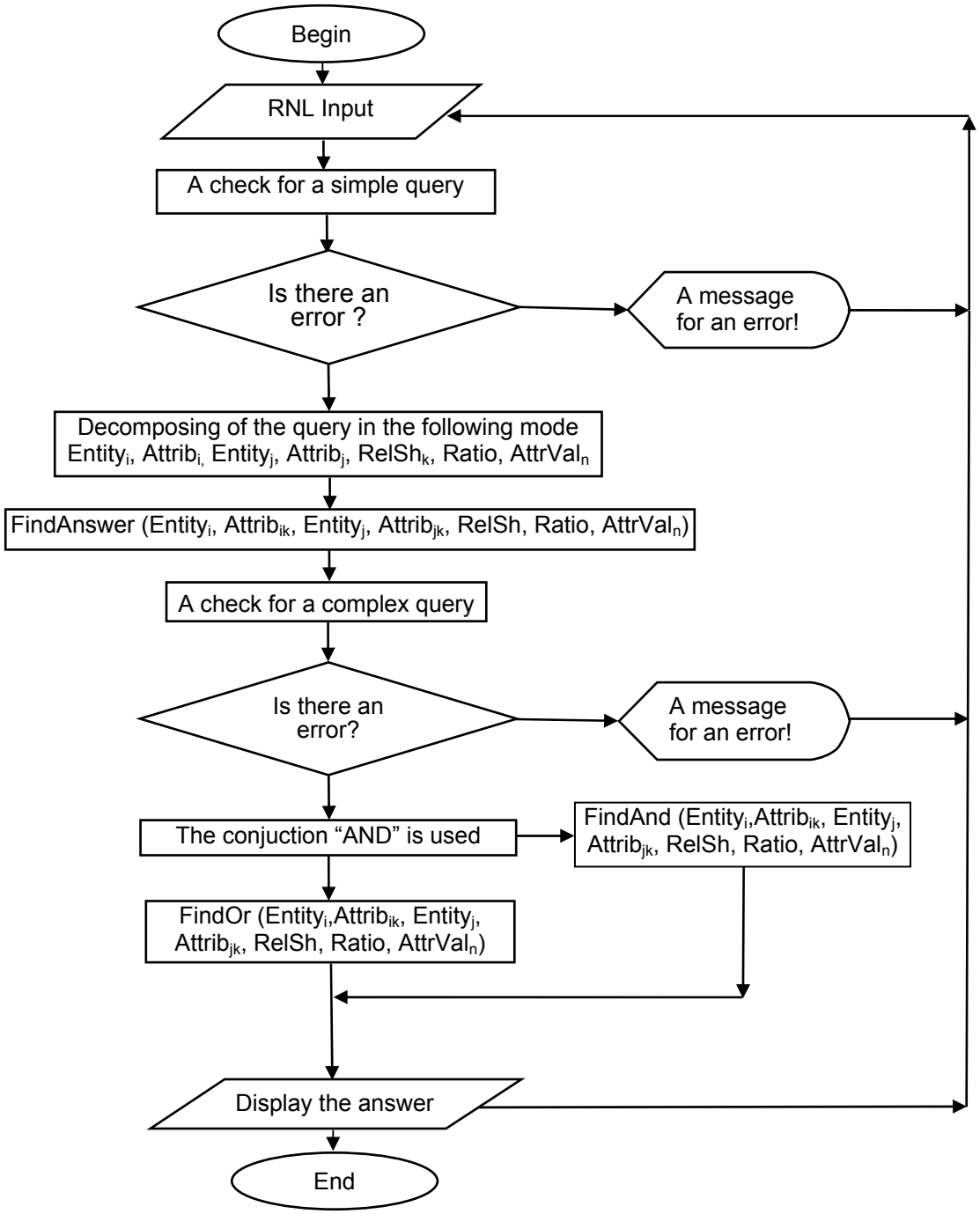

Figure 3. Summarized algorithm for response generation

When it is not found a rule, which corresponds to the query then a message for an error is displayed.

Table 2. Exemplary symbol table

№	Symbol	Код
1	"Какви"	QUEST
2	"дисциплини"	ENTITY
3	"изучава"	RELSH
4	"студент"	ENTITY
5	"име"	ATTRIB
6	"Милен Петков"	ATTRVAL

The summarized algorithm for response generation is presented in figure 3.

When the specified algorithm is executed, the query is structured in the symbol table, as it can be seen in table 2, which consists of the following elements: Entity$_i$, Relationship$_{i,j}$, Entity$_j$, Attribute$_{j,k}$, AttrVal$_{j,k,l}$.

It is followed by processing of the array and storing the relationships between the

entities.

When the participating relationship in the query is found, all relationships between the two attribute values of the related entities are processed for finding correspondence.

The unknown attribute value in the query corresponding to the given value of the known attribute is displayed as a query response.

We will discuss the exemplary query given in Bulgarian.

Какви дисциплини изучава студент име Милен Петков?

In what corses the student with name Milen Petkov participate?

After the analysis of the query, the symbol table is organized as it is shown in table 2.

Processing of the array with relationships for the relationship ("student" – "study" – "course") follows.

When such a relationship is found, all relationships between the values of the two attribute entities ("student"/"course") are processed. All values of the attribute "name" of the entity "course" are separated in a special array, which are related with the value Milen Petkov of the attribute "name" of the entity "student".

Actually the query response is in the described above array , which is a combination of the projection operation on the attribute student' s name and on the attribute name of course followed by the union operation.

The complex query are processed at the same algorithm as the query parts are processed separately and the collective response is displayed depending on the conjunction word AND or OR

For the designing of the described goals are used methods of the object-oriented design [11]. The described algorithm is implemented by using the tools of the object-oriented programming environment Visual C++.

CONCLUSION AND FUTURE WORK

It is developed a sub-system for query formulation in a restricted query language by choosing the words from a multiple- window menu. In this connection method for formulation of query constructions in a restricted Bulgarian language are proposed.

Algorithms for response generation to query with different types of constructions are propounded.

It is intended to investigate the system for the number of made errors by users in dependence on their qualification and the type and length of the formulated queries by them.

REFERENCES

[1] Akscyn, R., E. Yoder, and D. McCracken. The Data Model is the Heart of Interface Design. In proceedings of the SIGCHI conference on Human factors in computing systems pages 115-120, Washington, 1988.

[2] Arsov, S., and B.Rachev. Graphical Specification of Conceptual Database Schemes Using Modified Model "Entity-Relationship. In Proceedings of the International Coference on Computer Systems and Technologies, Sofia, Bulgaria, pages III.8-1 - III.8-13,22-23 June, 2003.

[3] Arsov, S., and B. Rachev. An Approach for Design of the Object-Oriented Databases with Restricted Natural Language Access. In Proceedings of the International Coference on Computer Systems and Technologies, Sofia, Bulgaria, pages II.13-1 - II.13-6, 19-20 June, 2003.

[4] Barros, F., and A. De Roeck. Resolving Anaphora in a Portable Natural Language Front End to a Database. In Proceedings of the 4th Conference on Applied Natural Language Processing, Stuttgart, Germany, pages 119-124,1994.

[5] Bates, M., R. Bobrow. Information retrieval using a transportable natural language interface. In Proceedings of the 6th annual international ACM SIGIR conference on Research and development in information retrieval. June, 1983.

[6] Bates, M., R. Bobrow, R. Ingria, D. Stallard. The Delphi natural language understanding system. In Proceedings of the fourth conference on Applied natural language processing, October, 1994.

[7] Carey, M., D. DeWitt, S. Vandenberg. A Data Model and Query Language for EXODUS. In proceedings of the 1988 ACM SIGMOD international conference on Management of data, Volume 17, Issue 3, July, 1988.

[8] Damerau, F.. Problems and Some Solutions in Customization of Natural Language Front Ends. ACM Transactions on Office Information Systems, Volume 3, Issue 2, pages165-184, April, 1985.

[9] Glockner, I., and A. Knoll. Natural Language Navigation in Multimedia Archives: An Integrated Approach. In Proceedings of International Conference of ACM-Multimedia '99, Orlando, FL, USA, october, 1999.

[10] Hafner, C., K. Godden. Portability of syntax and semantics in DATALOG. ACM Transactions on Information Systems (TOIS), Volume 3, Issue 2, pages 141-164, April, 1985.

[11] Marinov M., Using Planning Techniques in object-oriented design. In Proceedings of the 16th International Conference on SAER '2002, Bulgaria, pages 176-180, 2002.

[12] Martin, P., D. Appelt, B. Grosz, F. Pereira. TEAM: an experimental transportable natural-language interface. In Proceedings of 1986 on fall joint computer conference, November, 1999

[13] Napier, A., R. Batsell, N. Guadango, D. Lane. Impact of a restricted natural language interface on ease of learning and productivity. Communications of the ACM, Volume 32, Issue 10, October, 1989.

[14] Sopena, L., Natural language grammars for an information system. In Proceedings of the 6th annual international ACM SIGIR conference on Research and development in information retrieval, June, 1983.

[15] Templeton, M., and J. Burger. Problems in Natural Language Interface to DBMS with Examples from EUFID. In Proceedings of the 1st Conference on Applied Natural Language Processing, Santa Monica, California, pages 3-16, 1983.

[16] Waltz, D.. An English Language Question Answering System for a Large Relational Database. Communications of the ACM, Volume 21, Issue 7, July 1978.

[17] Woods, W., R. Kaplan, and B. Webber. The Lunar Sciences Natural Language Information System: Final Report. BBN Report 2378. Bolt Beranek and Newman Inc. Cambridge, Massachusetts, 1972.

[18] Yates, A., O. Etzioni, D. Weld. A Reliable Natural Language Interface to Household Appliances. In Proceedings of the Conference on IUI'03, Miami, Florida, USA, January 12-15, 2003.

ABOUT THE AUTHOR

Assist. Prof. Silyan Arsov, Department of Computer Systems and Technologies, University of Ruse,

Phone: +359 82 888 276, E-mail: sarsov@ecs.ru.acad.bg