Natural Language Interface for Web-based Databases*

J. ANTONIO ZÁRATE M.[1], RODOLFO A. PAZOS R.[1], ALEXANDER GELBUKH [2], JOAQUÍN PÉREZ O.[3, 1]
[1] National Center for Research and Technology Development (CENIDET),
[2] Computing Research Center (CIC) of National Polytechnic Institute (IPN),
[3] Institute for Electrical Research,
MEXICO

Abstract: - Advances in the work on interfaces that facilitate access to databases through Internet are presented. Increasing needs of the users that access computer resources, the technological advance in this field, and the limitations of the graphic interfaces and forms motivate the development of new solutions in human-machine interfaces. In recent years, natural language processing has received a new impulse and achieved sufficient maturity to become a real solution in human-machine interfaces. A general architecture of a system of natural language interface to Web-based databases is described, as well as the current advance of the project. A detailed review of the history and the state of the art of the problem is given.

Key Words: - Intelligent Communication Systems, Natural Language Interface, Relational Database, Natural Language Processing, Syntax, Semantics.

1 Introduction

The quick growth of the Internet is creating a society where the demand of storage services, organization, access and analysis of information is constantly increasing. The Internet era has changed the research directions in all areas of computer science, especially those related to databases [8].

The growing necessity by users without wide knowledge of computers to access data over the Internet has resulted in the development of many types of interfaces, e.g., QBE (query by example), forms, embedded languages, etc. These tools, even if they simplify the task of the user, always imply some degree of difficulty when translating what the user would normally express to another person, into a structured form appropriate for the query engine.

For users to be able to express a query easily, natural language interfaces to databases (NLIDB) are a very promising solution. They have attracted interest since the 70s, but several development problems have not yet been completely solved. Most of the NLIDBs are not really complete interfaces in natural language, since only the query component is designed to accept a query in a language restricted to the context of the specific database, though some such systems also accept restricted language expressions for data update [1].

* Work partially supported by Mexican Government (CONACyT, COSNET, and SNI) and RITOS-2 (CYTED).

2 Previous work in NLIDBs

2.1 Natural Language Processing

The first work on natural language interfaces (NLIs) was done by Warren Weaver in 1947 with translation systems. Due to the complexity of the problem and to hardware limitations, Weaver had to restrict the goals to a "microcontext," although some advances were made in the elaboration of a dictionary. At the end of the 70s, Victor Yngve of MIT proposed a grammatical method for NLP based on dictionaries.

In the early 70s, in Cambridge, Leningrad, Grenoble, and Texas some work was done on the "interlingua" approach: the idea that any natural language can be expressed in a universal representation. Heavily criticized, this idea, impossible to validate, was the origin of "knowledge representation." It also helped to conclude that NLP needed more knowledge than pure syntax of the language.

After that, a new era of semantic processing (based on semantic rather than syntactic patterns) was pioneered by Wilks, Weinzenbaum (Eliza and Doctor developed in 1966), and Colby (Parry implemented in 1975). Another branch of this idea tried to associate formal systems with NLP; examples are Student of Brobow (1968) and Baseball written by Chomsky, Green, Wolf, and Laughery. This system was one of the first database access systems.

Other interesting projects are the following: SHRDLU by Terry Winograd (1972) suggested a procedural representation of sentences; Margiede Roger Schank (around 1970) used conceptual dependences to represent sentences [14].

2.2 Interfaces to Databases

Meanwhile, concerning NLIDBs, the first antecedents are at the end of the 70s and the beginning of the 80s, with ad hoc built systems such as Lunar [45]: a system for search of chemical analysis of lunar rocks. Before that decade, it is difficult to speak of this type of interfaces since database technology did not reach its maturity until the introduction of the relational data model by Codd [12].

Other systems appearing in the 70s are the following: Rendezvous [13], built in the IBM laboratory in San Jose, California, provided help to users for formulating their queries; Ladder [19] allowed to access large databases of different DBMSs, make spelling error correction, and carry out elliptic reasoning. This system was based on semantic grammars, a mechanism that mixes syntactic and semantic processing, which allowed the system to have better understanding capacities. The problem was that the semantic grammar that allowed it to be adjusted very well to a certain domain, restricted its portability since rewriting the grammar was required whenever the application was to be changed. Other systems developed in the 70s were Planes [42] and Philiqa1 [34].

One of the main goals of the 80s was portability of the interfaces to different domains. An example of this is Chat80 [43], which translated user queries into expressions in a logical query language (LQL) and evaluated them against a database in Prolog. Chat80 served as a base for other prototypes, such as Masque [2] [4] [5] developed at the beginning of the 90s, which translated the query in LQL to SQL (Structured Query Language) to permit its execution against any DBMS that supports this standard. In a Ph.D. thesis that was the continuation of Masque, an NLI for temporal databases was developed [3].

Most of the work developed for NLIDBs was in the 80s decade. Some of the results obtained are represented by systems like Ask [39] that allowed working with its own database, external databases, electronic mail, and other systems. Janus [30], in the same way, had interfaces with different systems (databases, expert systems, graphical systems, etc.) to provide transparent access to these resources. This was one of the few systems that allowed queries involving time. Team [16] [17] was a portable easily configurable interface. Other systems developed in that decade were Datalog [18], Ldc, Eufid [38], TQA [24], and Teli [37].

In spite of the great effort made in the 80s decade, the hopes that this type of interfaces became of common use failed probably due to the inherent difficulties of language, to the idea of looking at NLIDBs as exotic systems, and to the emergence of friendlier graphical and form-based interfaces. Although at that time commercial prototypes appeared, their use has been very limited [15].

In the 90s, although the research in this specific area was no longer as intensive as in the previous decade, the advance in general NLIs, as well as in the theory of speech, integration of agents to reasoning, multimedia interfaces, generation of more complete dictionaries, search for formalisms, etc., contributed to the emergence of general purpose products. These translated a query in natural language to a logical form that was transformed into a standard form understandable for the DBMSs.

Among such commercial systems, we can mention:

- AICorp's Robot [20] was the origin of IBM's Intellect [41]. IBM also developed a multilingual interface LanguageAccess [22]; there is a Spanish version (Sylvia project) [26].
- Rus [9] and Irus [7] were the origin of BBN'Parliance [6].
- Natural Language 32, developed by the firm with the same name, derived from its DataTalker [32].
- Linguistic Technology's English Wizard evolved to EasyAsk, which allows complex queries (using subqueries, clauses HAVING, LIKE, EXISTS, etc.), output in different forms (spreadsheets, graphics, tables, etc.), execution within several languages such as C/C++, Visual Basic, Power-Builder, Delphi, Informix. It works with ODBC and works only under Win32. English Wizard/Voice, based on DragonDictate, converts a spoken query into a query in SQL [32].
- Themus, an interface to Oracle with the capability of learning through feedback [21].
- SystemX [11] translates queries in natural language into SQL. It has a wide coverage of English, accepts passive, imperatives, possessive, relations, prepositional sentences, and quantification. It has a modular structure, which allows translations of natural language into other database languages. In case of ambiguous queries it presents the possible interpretations to the user.
- Edite handles questions in Portuguese, English, French, and Spanish on tourist resources. It uses an intermediate query language (ILI) translated into SQL, which separates the linguistic component from the knowledge of the database, making it a portable product, although it is necessary anyway to make some changes [29].
- CINDOR (conceptual interlingua document retrieval) is a system for document retrieval that allows formulating queries in several languages (English, French, Spanish, and Japanese) translating the query to a language-independent

conceptual representation (interlingua). Unlike search engines (Yahoo, Altavista, etc.), it allows formulating the query without using keywords that have to match the document contents [33].

- SQ-Hal translates a query in natural language into SQL. It is platform, database, and DBMS independent. It can learn the grammar rules not introduced initially [32].
- MS English Query is a component of SQL Server 7.0 allowing the formulation of queries in English. One needs to provide it with the knowledge about the entities of the database and the relationships among them. It supports COM, Visual C++, Visual Basic, ASP, and DBMS using OLE [32].
- DB Valet [44] is a prototype that transforms English sentences into SQL using rules [40] [27].

The advance of database technology has also affected the development of NLIDBs, making it more difficult. The emergence of data models such as object-oriented, semantic, entity-relationship, unified modeling language (UML), etc. led to storing of more complex information. Also, the development of databases capable to manage more complex data (temporal, spatial, geographical, videotapes, images, etc.) increases the variety of queries compared to those feasible with the simple databases of the past.

2.3 Previous Work of the Authors

The first project developed at the beginning of the last decade by the Distributed Systems Group of CENIDET, Mexico, was the Distributed Database Management System (SiMBaDD). In recent years, the group has focused on the problems of data access via Internet, with particular interest in interfaces to databases that would be sufficiently friendly for the great number of new users usually inexperienced in handling databases via Internet. Some examples of projects developed for this purpose are the following:

- A QBE tool that allows inexperienced users to access databases through Internet, in a platform-independent way (implemented in Java) [10].
- A QBE tool for multi-databases in Internet. This work improved the interface in such aspects as querying multiple tables in different databases, subqueries, and help windows. In this project, the current architecture of the system was defined [25].
- An EzQ tool for multi-databases in Internet. This project in now under development. The aim is to improve the interface, mainly, in what concerns the ease with which inexperienced users can formulate queries involving joins without the need for the user to understand the concept of join [28].

The architecture of the QBE tool is shown in Fig. 1.

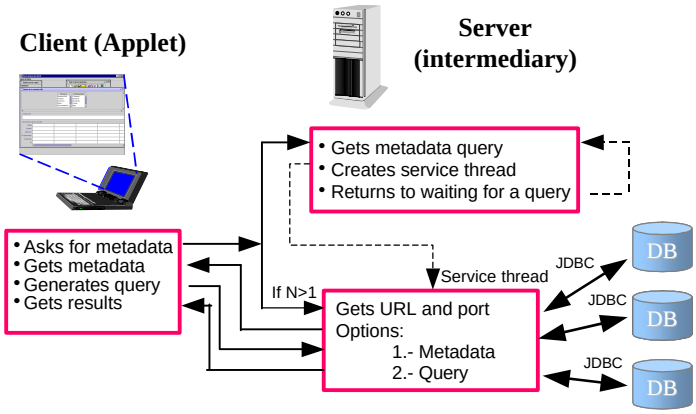

Fig. 1. Architecture of the QBE tool built by the group.

These developments have lead us to the conclusion that the next step is the implementation of interfaces in natural language [1], since we consider that we have exhausted the possibilities of other types of database interfaces, either by means of programming, like in the project SiMBaDD [36], or by means of graphic tools for inexperienced users [10] [25] [28].

It is well-known [35] that natural language interfaces to databases are not the panacea to solve all the problems of human-machine interaction. However, in the same study [35] it is demonstrated that in the cases when several tables are involved or when the solution is not similar to the examples previously known to the user, the interfaces in natural language prove to be simpler than the graphical interfaces or programming environments.

An experiment carried out with Intellect [1] has demonstrated that natural language is an effective method for the interaction of casual users with a good knowledge of the database in a restricted environment. The approaches to evaluation of such type of interfaces are given in [1].

3 Natural Language Query System

The authors are developing a NLIDB system for the Spanish spoken in Mexico. It has additional elements with respect to other similar systems [14] [23] [31], a better language coverage, much better portability of DBMS and operating system, transparent access through Internet, and a greater use easiness.

The architecture used previously (Figure 1) was modified in a substantial way. The three-level client-intermediate-server structure is preserved, but the functionality of each level has been changed. The client is now much simpler, which partially solves the problems of the current QBE interface, at the cost of a more active role of the intermediary level.

The new architecture of the system is shown in Figure 2. The client was changed to present to the user the representation of the knowledge stored in the

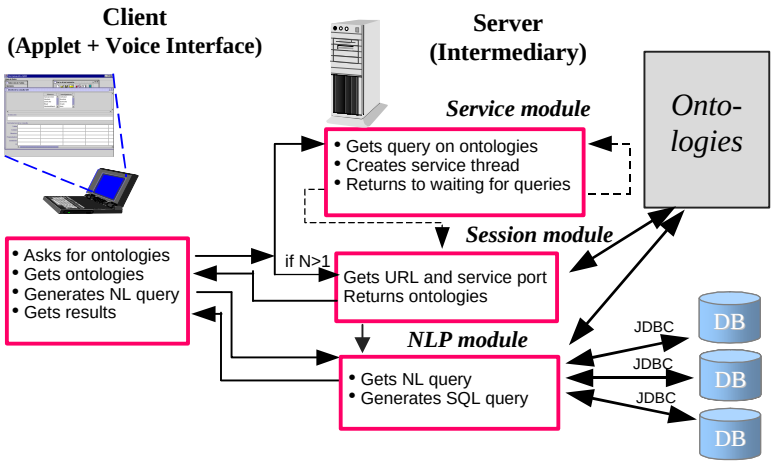

Fig. 2. System architecture.

database through an ontology, unlike the QBE that shows the database designer's abstractions through tables that most of the times are difficult to understand for the inexperienced users and which also lack a lot of very important semantic information. The presentation of this ontology permits the user to better understand the contents of the database, which facilitates query formulation.

By definition of the American Heritage Dictionary, "ontology is the branch of metaphysics associated with the nature of being." The artificial intelligence community has adopted it to refer to the group of concepts or terms used to describe some area of knowledge or to build its representation. An ontology can consist of concepts of very high level organized around a knowledge base.

To be able to introduce the ontology to the user, the client communicates with the intermediary module. The latter generates a session thread that forwards the query to a module that builds the ontology from the information of the data dictionary and from an "expert" that contributes with the information that is not usually present in the data dictionary. This module of ontologies forwards the ontology to the module of the intermediary's session, which returns it to the client. Finally, it is formatted for final presentation to the user.

After the user connects to the database and is presented the information on it through ontologies, he/she introduces a query using the voice interface. The output of this voice interface is received by the client and passed on to the module of the intermediary's session, which passes it to the natural language processing module (NLP). The architecture of the NLP module is quite standard, except that a module that acts between the data dictionary and the NLP lexicon is added, as shown in Fig. 3.

The NLP module receives a query in natural language from the session module and returns it translated to SQL. The session module sends the SQL query to the DBMS, which returns the result of the query. The session manager forwards the result to the client, which formats it and presents it to the user.

The main reason to add the module of interconnection between dictionaries is that both the data dictionary and the lexicon have a lot of information that jointly can facilitate the work of the syntactic and semantic analyzers and probably other analyzers such as the context discourse analyzer.

Usually in a lexical dictionary there is information on how the words are related to each other through synonymy, antonimy, holonimy, meronimy, etc. This significantly helps in the processing of natural language. At the same time, a data dictionary holds information of the attributes, entities, and views that constitute the database, as well as the relationships between the attributes and entities, and even some semantic information that, similarly to the lexicon, can substantially improve the processing of natural language.

As an example, consider the query "what is the size of the shirt of John Smith." The lexicon would indicate that the query is correct, but the information of the data dictionary represented through ontologies would facilitate detection of possible anomalies at early stages of the analysis. In this example, the requested data is not defined as an attribute. However, the user can consult the information on synonyms without the necessity for them to be defined in the ontology, since the synonyms are usually defined in the lexical dictionaries.

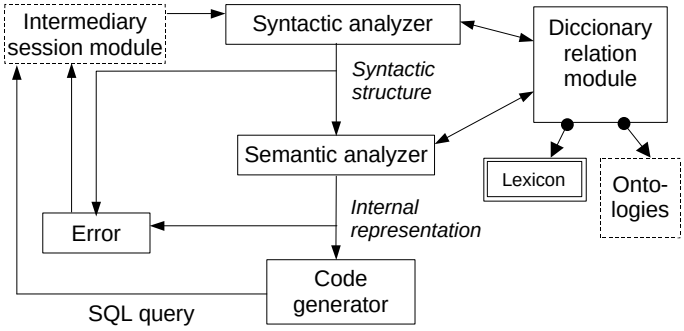

Fig. 3. NLP module.

4 Current Progress

We are developing a prototype of the system, especially as to implementation of the syntactic analyzer and the structuring of the lexicon. An important part of this project is the design of

ontologies from the data dictionary and the connection of this ontology with the lexicon, to prove that this connection helps the processing of natural language.

The current dictionary, developed according to WordNet standards, has 18 categories of nouns and 7 categories of verbs; also, it includes a database of verbs and their conjugations that helps the semantic analyzer to simplify its work.

For obtaining ontologies from a data dictionary, the dictionary of the DBMS Oracle 8i was chosen since it is widely used and the necessary technical information can be easily obtained.

The other modules are being developed both at CENIDET and at the Technology Institute of Ciudad Madero, within the frame of the Ibero-American Network of Software Technologies (RITOS2).

5 Final Remarks

To develop natural language interfaces is very important because of the necessity of providing access to computers to all members of society. With NLIs the access language to computers will be the same that people use, either in written or spoken form.

A study [35] carried out by a group of information system administrators on the usefulness of different applications of natural language interfaces concluded that those used for obtaining information from databases was preferred by users over those of information retrieval and text preparation. This type of interfaces left very far behind other applications such as language translation.

There are many aspects in natural language processing to work on, such as linguistics, computational linguistic, psychology, psycholinguistics, etc. In Mexico, though there is some work related to NLP, very few projects deal with database querying.

The work on natural language interfaces is necessary because there are more and more people that need access to computer resources but do not have experience in this nor usually time to acquire it. Also, being Spanish the third language in the world by the number of native speakers (around 390 million), it is very important to develop appropriate tools for this huge market.

References:

[1] I. Androutsopoulos, G.D. Ritchie, P. Thanisch, *Natural Language Interfaces to Databases, An Introduction*, citeseer.nj.nec.com/1natural.html.

[2] I. Androutsopoulos, G. Ritchie, and P. Thanisch, An Efficient and Portable Natural Language Query Interface for Relational Databases, *6th Intern. Conf. on Industrial & Engineering Applications of Artificial Intelligence and Expert Systems*, Edinburgh, U.K. Gordon and Breach Publishers Inc., Langhorne, US, 1993.

[3] I. Androutsopoulos, *A Principled Framework For Constructing Natural language Interfaces to Temporal Databases*, PhD thesis, research paper No. 709, Dept. of Artificial Intelligence, Edinburgh University, Scotland, UK. 1996.

[4] P. Auxerre, *MASQUE Modular Answering System for Queries in English, Programer's Manual*, technical report AIAI/SR/11, Artificial Intelligence Applications Institute, University of Edinburgh, March 1986.

[5] P. Auxerre and R. Inder, *MASQUE Modular Answering System for Queries in English, User's Manual*, technical report AIAI/SR/10, Artificial Intelligence Applications Institute, University of Edinburgh, June 1986.

[6] BBN Systems and Technologies, *BBN Parlance Interface Software – System Overview*, 1989.

[7] M. Bates, M.G. Moser, and D. Stallard, The IRUS transportable natural language database interface, in L. Kerschberg (Ed.), *Expert Database Systems*, Benjamin/Cummings, 1986.

[8] P. Bernstein, M. Brodie, S. Ceri, D. De Witt, M. Franklin, H. García-Molina, J. Gray, J. Held, J. Ellerstein, H.V.Jagadish, M. Lesk, D. Maier, J. Naughton, H. Pirahesh, M. Stonebraker, and J. Ullman, *The Asilomar Report on Database Research* , 1998.

[9] R.J. Bobrow, The RUS System, *Research in Natural Language Understanding*, BBN report 3878. Bolt Beranek and Newman Inc., Cambridge, Massachusetts, 1978.

[10] G. Carreón Valdés, *Herramienta para Consultas EzQ para Multibases de Datos en Internet*, M.S. thesis, CENIDET, Cuernavaca, Mexico.

[11] N. Cercone, P. McFetridge, F, Popowich, D. Fass, C. Groeneboer, G. Hall, *The SystemX Natural Languaje Interface: Design, Implementation and Evaluation*, Centre for Systerns Science, Simon Fraser University, British Columbia, Nov. 1993.

[12] E.F. Codd. A Relational Model for Large Shared Data Banks, *Comm. of the ACM*, 13(6), 1970.

[13] E.F. Codd. Seven Steps to Rendezvous with the Casual User, in J. Kimbie and K. Koffeman (Eds.), *Data Base Management*, North-Holland Publishers, 1974.

[14] E. Chay Coyoc, *Una Interfaz en Lenguaje Natural en Español para Consultas a Bases de*

Datos, MS thesis, ITESM-Cuernavaca, Mexico, 1990.
[15] S.M. Dekleva, Is Natural Language Querying Practical? *Data Base*, May 1994.
[16] B.J. Grosz. TEAM: A Transportable Natural-Language Interface System, *1st Conf. on Applied Natural Language Processing*, Santa Monica, California, 1983.
[17] B.J. Grosz, D.E. Appelt, P.A. Martin, and F.C.N. Pereira, TEAM: An Experiment in the Design of Transportable Natural-Language Interfaces, *Artificial Intelligence*, 32, 1987.
[18] C.D Hafner and K. Godden, Portability of Syntax and Semantics in Datalog, *ACM Transactions on Office Information Systems*, 3(2), 1985.
[19] G. Hendrix, E. Sacerdoti, D. Sagalowicz, and J. Slocum, Developing a Natural Language Interface to Complex Data, *ACM Transactions on Database Systems*, 3(2), 1978.
[20] L.R. Harris, The ROBOT System: Natural Language Processing Applied to Data Base Query, *ACM'78 Annual Conference*, 1978.
[21] V. O. Huerta H., *Un Método para el Reconocimiento a Bases de Datos en Interrogaciones en Lenguaje Natural*, MS thesis, ITESM-Cuernavaca, Mexico.
[22] International Business Machines; http://www-4.ibm.com/software/speech/es/.
[23] H. Jiménez S., *Adaptación de Algoritmos de Aprendizaje Mecánico para Obtención de Reglas en Corpora del Español de México*, PhD thesis, BUAP, Puebla, Mexico.
[24] R. Kasper, A Flexible Interface for Linking Applications to Penman's Sentence Generator, *DARPA Speech and Natural Language Workshop*, 1989.
[25] A. May Arrioja, *Herramienta para Consultas Basadas en Ejemplos (QBE) para Multibases de Datos en Internet*, MS thesis, CENIDET, Cuernavaca, Mexico, 2000.
[26] F. Marcos Marín**,** A. Moreno, C. Olmeda, J. Martínez, and S. Guilarte. Proyecto Sylvia; www.lllf.uam.es/proyectos/sylvia.html.
[27] R. J. Money, *Inductive Logic Programing for Natural Language Processing*, Dept. Computer Sciences, Univ. of Texas.
[28] F. Rasgado Celaya, *Herramienta para Consultas Basadas en Ejemplos (QBE) para una Base de Datos en Internet*, MS thesis, CENIDET, Cuernavaca, Mexico, 1999.
[29] P. Reis, J. M. Nuno Mamede, *Edite – A Natural Language Interface to Databases: a New Dimension for an Old Approach.* INESC, Portugal.
[30] P. Resnik. *Access to Multiple Underlying Systems in JANUS*, BBN report 7142, Bolt Beranek and Newman Inc., Cambridge, 1989.
[31] G. R. Rocher Silva, *Traducción de Queries en Prolog a SQL*, BE thesis, Escuela de Ingeniería, Universidad de las Américas-Puebla, 1999.
[32] J. Rojas, J. Torres, A Survey in Natural Language Databases Interfaces, *8vo. Congreso Intern. de Investigación en Ciencias Computacionales*, Inst. Tecnol. de Colima, Mexico 2001.
[33] M. Ruiz, A. Diekema, P. Sheridan, *CINDOR Conceptual Interlingua Document Retrieval: TREC-8 Evaluation*, MNIS-TextWise Labs, Syracuse, NY 13202.
[34] R.J.H. Scha, Philips Question Answering System PHILIQA1, *SIGART Newsletter*, No. 61. ACM, New York, 1977.
[35] V. Sethi, *Natural Language Interfaces to Databases: MIS Impact, and a Survey of Their Use and Importance*, University of Pittsburgh, PA, USA.
[36] DCC, *SiMBaDD Sistema Manejador de Bases de Datos Distribuidas*, CENIDET, Cuernavaca, Mexico, www.sd-cenidet.com.mx/simbadd.
[37] D. Stumberger, B. Ballard, Semantic Acquisition in TELI, *24th Annual Meeting of ACL*, New York, 1986.
[38] M. Templeton and J. Burger, Problems in Natural Language Interface to DBMS with Examples from EUFID, *1st Conference on Applied Natural Language Processing*, Santa Monica, California, 1983.
[39] B.H. Thompson and F.B. Thompson, Introducing ASK, A Simple Knowledgeable System, *1st Conference on Applied Natural Language Processing*, Santa Monica, CA, 1983.
[40] B.H. Thompson, C. Raymond, and J. Money, *Automatic Construction of Semantic Lexicons for Learning Natural Language Interfaces*, Stanford Univ., University of Texas.
[41] Trinzic Corporation, Bethesda, MD. *INTELLECT – Natural Language System.* (commercial triptych).
[42] D.L. Waltz, An English Language Question Answering System for a Large Relational Database, *Comm. of the ACM*, 21(7), 1978.
[43] D. Warren and F. Pereira, An Efficient Easily Adaptable System for Interpreting Natural Language Queries, *Comp. Linguistics*, 8 (3-4), 1978.
[44] A. W. Penn, *DB Valet: A natural language database interface*, MS thesis, Univ. of Lousville, 2000, www.louisville.edu.
[45] W.A. Woods, Procedural Semantics for a Question-Answering Machine, *Fall Joint Computer Conference*, NY, 1968. AFIPS.