USL / DBMS NASA / RECON WORKING PAPER SERIES

Report Number

DBMS.NASA/RECON-6

The USL/DBMS NASA/RECON Working Paper Series contains a collection of reports representing results of activities being conducted by the Computer Science Department of the University of Southwestern Louisiana pursuant to the specifications of National Aeronautics and Space Administration Contract Number NASW-3846. The work on this contract is being performed jointly by the University of Southwestern Louisiana and Southern University.

For more information, contact:

Wayne D. Dominick

Editor
USL/DBMS NASA/RECON Working Paper Series
Computer Science Department
University of Southwestern Louisiana
P. O. Box 44330
Lafayette, Louisiana 70504
(318) 231-6308

CONCEPTS AND IMPLEMENTATIONS

OF

NATURAL LANGUAGE QUERY SYSTEMS

I-Hsiung Liu

Computer Science Department
University of Southwestern Louisiana
P. O. Box 44330
Lafayette, Louisiana 70504

June 1, 1984

ABSTRACT

The currently developed user language interfaces of information systems are generally for experienced users. These interfaces commonly ignore potentially the largest user group, namely, causual users. This project discusses the concepts and implementations of a natural query language system which satisfy the nature and information needs of causual users by allowing them to communicate with the system in the form of their native language. In addition, a framework for the development of such an interface is also introduced for the MADAM (Multics Approach to Data Access and Management) system at the University of Southwestern Louisiana.

TABLE OF CONTENTS

CONCEPTS AND IMPLEMENTATIONS OF NATURAL LANGUAGE QUERY SYSTEMS

1. THE IMPORTANCE OF NATURAL LANGUAGE QUERY SYSTEMS

The major function of a computerized information system is to enable its users to retrieve and modify any subset of the data in the data bases, as well as to provide support to its users in their decision-making activities. In other words, the computerized information system is developed to serve its users and satisfy their immediate needs for information. Thus, the success or failure of an information system is ultimately decided by the users the system is supposed to serve.

The predominant criterion in evaluating an information system, from the user's viewpoint, is whether the system allows him to freely communicate to it and satisfy his needs. Therefore, the interface problem in user-system interaction must be seriously considered while developing an information system. The concept of a multi-user query system is, therefore, applied in the development of most information systems to allow different user groups to communicate with the system by using specific data base sublanguages. But, these multi-user query systems all too often ignore the nature of the largest group of the user population - casual users.

Casual users, as defined in [Codd, 74] are ones "whose interactions with the system are irregular in time and motivated by (their) jobs or social roles." Such users may not only lack knowledge about computers, programming, logic, or relations, but also are not willing to learn an artificial language. The only query language which they are willing to use to interact with a data base system is their native language.

After examining the characteristics of casual users, it is conceivable that the traditional approach of query language design, which assumes that the users are willing to develop the appropriate skills and learn the "user-supports" to operate an information system, cannot cope with the nature of those casual users. For example, Query-By-Example has been proven by many behavioral researches [Greenblatt, 78; Waxman, 78; Zloof, 78] to be an easy-to-use query language for non-programmer users. But, before a user can manipulate this language, he still requires about three hours or four sessions of instruction and a knowledge of first-order predicate calculus. In order to satisfy the information need of casual users, therefore, a natural language query system should be developed so that the casual users can freely employ their native languages to specify what they want while interacting with the system.

The purpose of this project is to briefly examine the current development of natural language query systems, and present a framework for such a system for the "SEARCH" subsystem of MADAM. Finally, the difficulties involved in the construction of such a query system will be evaluated and some solutions to those problems are proposed for future studies.

2. METHODOLOGY

Recognizing the importance of a natural language query system, considerable research has been performed in this field during the last decade [Codd, 74; Lockemann, 75; Waltz 77; Goodman, 77]. This project intends to examine these research activities and, based on the knowledge obtained through the above work, to propose a framework for the development of a natural language query system within MADAM.

By definition, a "framework" identifies the relationships between the parts, and reveals the areas in which further development will be required [Spragne, 1980]. Thus, in this project, it is necessary to examine the relationships between the natural language query system and the existing subsystems of a database system, especially the "SEARCH" subsystem, as well as the relationships between various components in the natural language query system itself. In addition, issues addressing the development of such a system, such as the restrictions and the reasonable alternative approaches to developing such a query system, are to be discussed.

To accomplish the above purposes, the method applied in this project is a library research approach. The research proceeded in the following five steps:

(1) To discuss the concepts of a natural language query system. The major concern lies in the conceptual relationships between the natural language query system and other database sublanguages within the database system, as well as the relationships between components within a natural language query system.

(2) To discuss the implementation of the above relationships.

(3) Based on the above discussions, to generate a framework for the development of a natural language query system for MADAM.

(4) To discover the difficulties involved in the development of the above system.

(5) To propose some directions to solve the above problems.

3. CONCEPTS OF NATURAL LANGUAGE QUERY SYSTEMS

The primary objective of a natural language query system is to permit casual users to engage in effective communication with a formatted database system by applying their native language. To develop such a query system, it is necessary to understand the interrelationships between the natural language query system and the database system, as well as the interrelationships between the components within the natural language query system. In this section, these relationships are examined, and the implementation of these relationships will be discussed in the next section.

3.1 Relationships Between Natural Language Query Systems and the Data Base System

The data base design can be separated into at least three design levels (Figure 3-1). The first level, the user's logical level or information structure level, is the logical representation of "facts" in reality. The second level, data base level or system's logical level, is not visible to the users and at this level, the logical structuring of data is shown. The third level is the physical storage structure level where the data are actually structured and stored in secondary storage.

```
Fact                      REALITY
                             |
                             |
                             |
                             |
                 +-----------------------+    Information
Information      |  USER'S LOGICAL LEVEL |    Structure Description
Structure        +-----------------------+    Language ( ISDL )
                             |
                             |
                             |
                             |
                 +-----------------------+
                 |     SYSTEM'S LOGICAL  |   Data Structure
Data             |          LEVEL        |   Description Language
Structure        +-----------------------+   ( DDL )
                             |
                             |
                             |
                             |
                 +-----------------------+
                 |    PHYSICAL STORAGE   |   Storage Structure
Storage          |          LEVEL        |   Description
Structure        +-----------------------+   Language ( SDL )
```

FIGURE 3-1 Levels of Data Base Design [Nijessen, 1974]

While using a query language to retrieve information from a data base system, the casual user is supposed to know "what 'fact' he wants to obtain from the system and how to express his intention in his preferred language", instead of knowing "how to access the data from the data base system". Thus, a good query language should be based on the concepts of the data description model and try to avoid as many as possible of the concepts which are machine dependent [Ghosh, 77]. Currently developed query languages for casual users are mainly of the form of restricted English, such as SEQUEL [Boyce, 1974] and QUERY-BY-EXAMPLE [Zloof, 1978]. Although these languages have a high degree of machine independence, they are not "natural" to casual users [Greenblatt, 78; Waxman, 78]. As Codd suggested, the only way to entice the casual user to interact with a data base system is to permit him free use of his native language. Therefore, the development of a natural language query system is necessary and important. The following discussion explains that the development of such a query system that relates to an existing data base system is possible by extending the concept of the hierarchy of user languages.

3.2 Hierarchy of User Languages

The hierarchy of data base interfaces, as presented in

Kraegeloh's description [Kraegeloh, 75], can be defined as follows:

(1) Each interface is defined in terms of a lower interface, and may itself serve as the basis for definition of a higher interface.

(2) There is exactly one interface which cannot be defined in terms of another interface and hence serves as the ultimate basis for all other interfaces.

Following the above definition, [Lockemann, 75] introduced some general rules on the design of user languages. He claimed that, keeping to those rules, a new interface can be defined in terms of its immediate predecessor, and at any level of the hierarchy, a user can formulate his queries without the necessity of knowing the language existing in the lower levels.

Based on their concepts, it is obvious that one possible step of abstraction is the definition of a new interface level which accepts natural language as input. To this new interface, the existing high level query language such as SEQUEL becomes the immediate predecessor. This query language, acting as the intermediate of the natural language, translates the natural language input into a formal query language, processes it down to the lowest level of the hierarchy, and then translates the results or responses into the natural language acceptable to the user.

3.3 Development of Natural Language Query Systems

The above discussion claims that a natural language query system is possible to be developed on the existing hierarchy of user languages. Some systems that support natural language have already been developed both in the DBMS and AI research fields, such as RENDEZVOUS, LUNAR, ELIZA and PLANES. The major difference among those systems lies in the emphasis on the type of dialogue which the system uses to communicate with its users. Those types of dialogue are as follows [Codd, 1974]:

Stroking Dialog

In this approach, when the system receives the queries from its user, it assures the user that it has been listening to him and invites him to cotinue his queries. Weigenbaum's ELIZA is of this type.

Contributive Dialog

In this approach, dialog involves contributive utterances and questions about changes of state in the given environment, but not questions about the meaning of previous utterances. Winograd's SHURDLU is of this type.

Clarification Dialog

In this approach, dialog involves queries about previous utterances in the dialog. Codd's RENDEZVOUS and its version 1, version 2 and Waltz's PLANES are of this type.

Although there are some distinctions among the above approaches and systems, the ideas about the structure of a natural language query system which built up the user-system interaction are quite similar as shown in Figure 3-2. This diagram shows the interaction between the user and the major components of the natural language query system, as well as the relationships between those components.

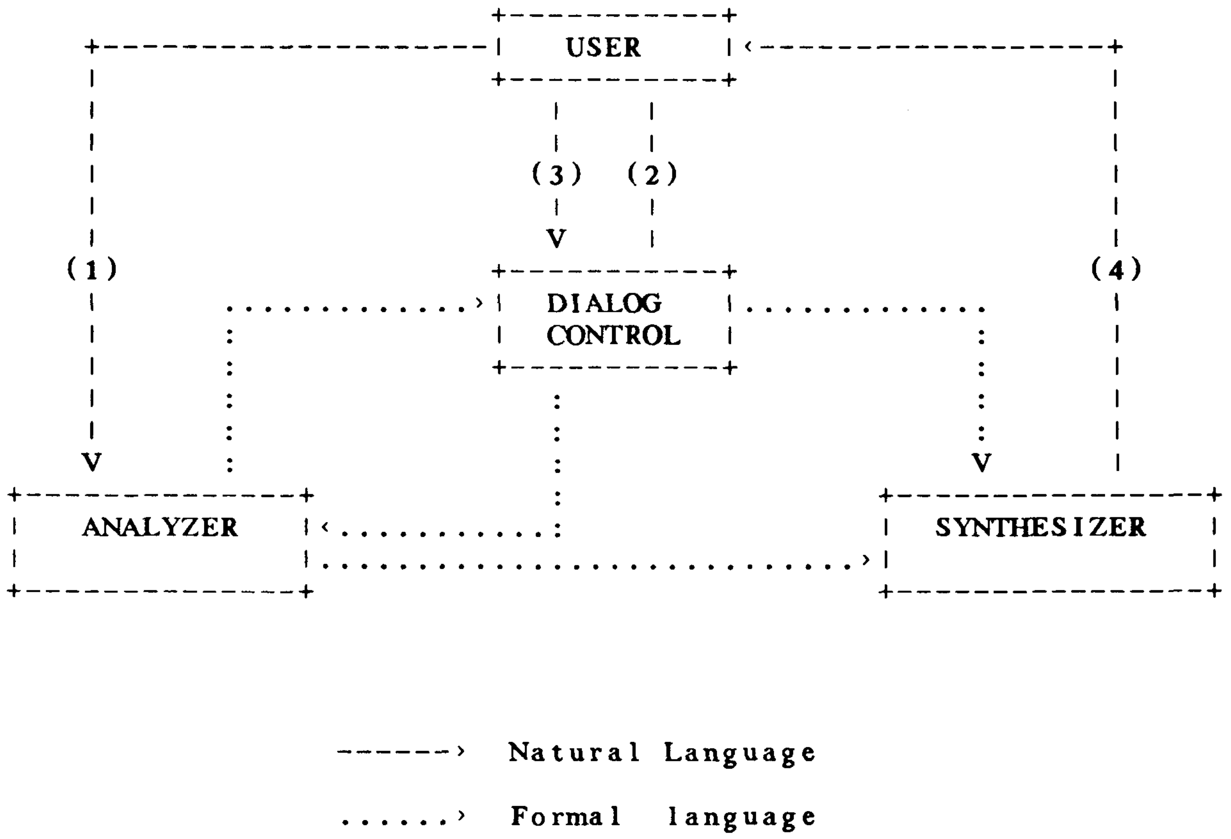


FIGURE 3-2 Components of a Natural Language Query System

3.4 Relationships Between Components of the Natural Language Query System

In Figure 3-2, the relationship of three major components of a natural language query system are identified, and the system processes user query as follows:

(1) The user initiates his request in the form of his native language which may be both syntactically and semantically complete, or it may involve certain ambiguity resulting from syntax errors, incompleteness or semantic incompatibility. After the analyzer of the system language processor receives the query, it first reduces the redundant words or phrases in the query; then analyzes the syntax of the query and its semantics; third, the results of the analysis are used to generate a formal query defined by the system. This formal query may be incomplete either syntactically or semantically. If it is so, then the formal query is tramsmitted to the dialog control. The dialog control examines the logical completeness of the query, and decides what strategy the system should apply to communicate with the user in order to achieve the logical completeness of the formal query.

(2) The decision is used to generate a question, usally in the form of multiple-choice, and transmitted to the user.

(3) After the user provides additional information, the dialog control examines the logical completeness and repeats the above procedure until the objective is achieved. After the logically complete query is generated, the data sublanguage statements are transmitted to the synthesizer.

(4) The synthesizer, after received the formal query, transforms it into a precise natural language restatement based on the formal query, and transmits this restatement back to the user.

If the user approves the system restatement, then the analyzer informs the synthesizer so that it can initiate the data retrieval operations. Otherwise, the above process will be repeated until there is an agreement reached between the user and the system.

The next part in this section explains the functions of each component involved in this four-step process.

3.4.1 The Analyzer

The analyzer performs three major functions: first, it translates as much as possible of the user's query into a data base language; second, it discovers and analyzes the translation difficulties such as the ambiguity involved in the query, as well as generates sufficient parameters to enable the dialog control to piece together dialog appropriate to the state of translation and the state of the user interaction; finally, it provides interpretation of the user's response.

The way an analyzer performs the above functions is based on the technique of the implementation. In general, there are two consecutive operations involved in the analyzer: parsing and interpretation. In the parsing phase, the user's request is transformed into normalized form by reducing the redundant words or phrases. In the interpretation phase, the feature and value representation of the user's request is translated into the formal query language, which then will be used to proceed to the actual information retrieval activity.

3.4.2 The Dialog Control

The dialog control provides a base for experimentation with dialog styles and tactics. For example, it is the mechanism to express hōw fully the user is kept informed of the system's progress in understanding his input and what type of dialog the system should generate in responding to the user's query.

The decisions on the construction of "re-statement" or "para-phrase" feedback to the user are the operations involved in this component. This feedback ensures that the further operation of the system will be consistent with the user's intention, or it clarifies the segment of the user's query which the system is unable to understand.

3.4.3 The Synthesizer

The synthesizer is responsible for translating the system's response, a restatement or the answer, back into natural language from the formal data sublanguage.

The above three principal components are fundamental to all the natural language query systems. Therefore, in implementing such a system, the implementor has to ensure that the system's design is able to perform the functions required in each component. In the next section, the implementation of two sample natural language query systems, RENDEZVOUS and PLANES, are overviewed to examine the techniques required in implementing such a query system.

4. IMPLEMENTATION OF NATURAL LANGUAGE QUERY SYSTEMS

The discussion of the implementation of a natural language query system in this project is to be divided into three parts: first, to describe the general rules for developing such a system; second, to overview a block diagram developed by Codd [Codd, 78] which includes the concepts discussed in the previous section; and finally, to discuss the functional performance of the implemented systems.

4.1 The General Rules of the Implementation

In his "Seven Steps to Rendezvous With the Casual User", [Codd, 74] proposed seven steps required in the implementation of a natural language query system. They are:

(1) Selecting a simple data model which can describe the data in a relatively simple way, both syntactically and semantically

The selection of a data model is strongly affected by the fact that data definition would be performed by untrained personnel. It is believed that, in terms of the logical view of the data, the relational data model is by far the most comprehensible to the non-computer-professional user The requirement that all data be in tables, with named field specifiers, unique keys, and no multiple-valued field entries, would guarantee first normal form without causing conceptual difficulty on the part of the user. Unlike CODASYL DBTG systems, where the relationships among data items are explicitly stated via set relationships and the data manipulation language must navigate among the sets, in a relational scheme the data can remain simple in structure and the complexity can be buried in the access methods employed by the software, which are invisible to the user. The ability of the relational systems to create new relationships among data items in response to each new query also results in efficient use of space, as a complex network of links and pointers does not have to be maintained.

Based on the above comparison between two mainstream data models, by selecting the relational model, the user, particularly the casual user, can be effectively isolated from the actual data base organization, and this is also the main reason that both the RENDEZVOUS and PLANES systems apply the relational approach.

(2) Selecting a high level logic as the internal target

In the natural language query system, the user expresses his request by using potentially poorly formulated language statements. This informal query may be ambiguous and/or semantically incompatible with the data base description. In order to enable the system to detect those problems, the user's source stātements have to be translated into an internal, precise language which is acceptable to the system. The data sublanguage ALPHA, a relational calculus language [Codd, 1971], which was selected as the internal target by both RENDEZVOUS and PLANES, and DEDUCE, which was selected by RENDEZVOUS 1 and 2, are examples. The reason for selecting such a relational calculus language is because of its simplicity, completeness, nonproceduralality and extensibility. Most of all, "the ALPHA-like languages provide a common core of features that will be required in some shape or form in all natural languages" [Codd, 1974, 1978].

(3) Introducing a strategy by which the system can keep the dialog closely tied to the data base description and the user's intended query

Most often, a user's query, while in the form of natural language, will omit information necessary to form an adequate query, such as ellipsis, using pronouns instead of nouns, or missing necessary information. There are many different strategies that have been applied to allow the system to address these problems. In RENDEZVOUS, the analyzer simply ignores the incomprehensible words, during the analysis stage, translating the remaining comprehensible words into formal languages, and piecing them together to determine what things are missing. In

PLANES, in addition to the above strategy, a context register, the history keepers which store all semantic constituents of user's requests and answers to earlier questions and other information, and concept case frames, a template representation of an entire series of questions about the data base, are used by the analyzer. By applying the concept of pattern matching, if incomprehensible words are encountered, the analyzer looks back through past context register values to fill in missing elements in order to solve the above problems, and to extend the system's ability in the future analysis process.

(4) Introducing system re-statement of user's query

The purpose of the introduction of re-statement is to ensure that the system has correctly interpreted the user's query.

(5) Separating query formulation from the data base search

The purpose of this separation is to protect the user from what may be expensive searches for information he does not want.

(6) Employing multiple-choice interrogation as fall-back

When a user's query consists of certain words or phrases which are incomprehensible to the system such as a syntax error, undefined data item or record, or certain misspelled words, and the system cannot generate a formal query without understanding them, the system needs to form multiple-choice question as a fall-back in order to prevent further problems with the interaction between the user and the system.

(7) Providing a definition capability

In a non-dialog environment, the user must observe the need for a definition and supply it to the system's library of definitions. But in the dialog environment of the natural language query system, the system must take this responsibility; it must detect the needs for definitions and extract them for the user.

4.2 A Block Diagram of a Natural Language Query System

By examining the research in the natural language query systems, particularly in Lockemann's KAIFAS, and Codd's RENDEZVOUS and its version 1, a block diagram which presents the structure of a natural language query system and describes the relationships between components of such a system can be introduced as Figure 4-1.

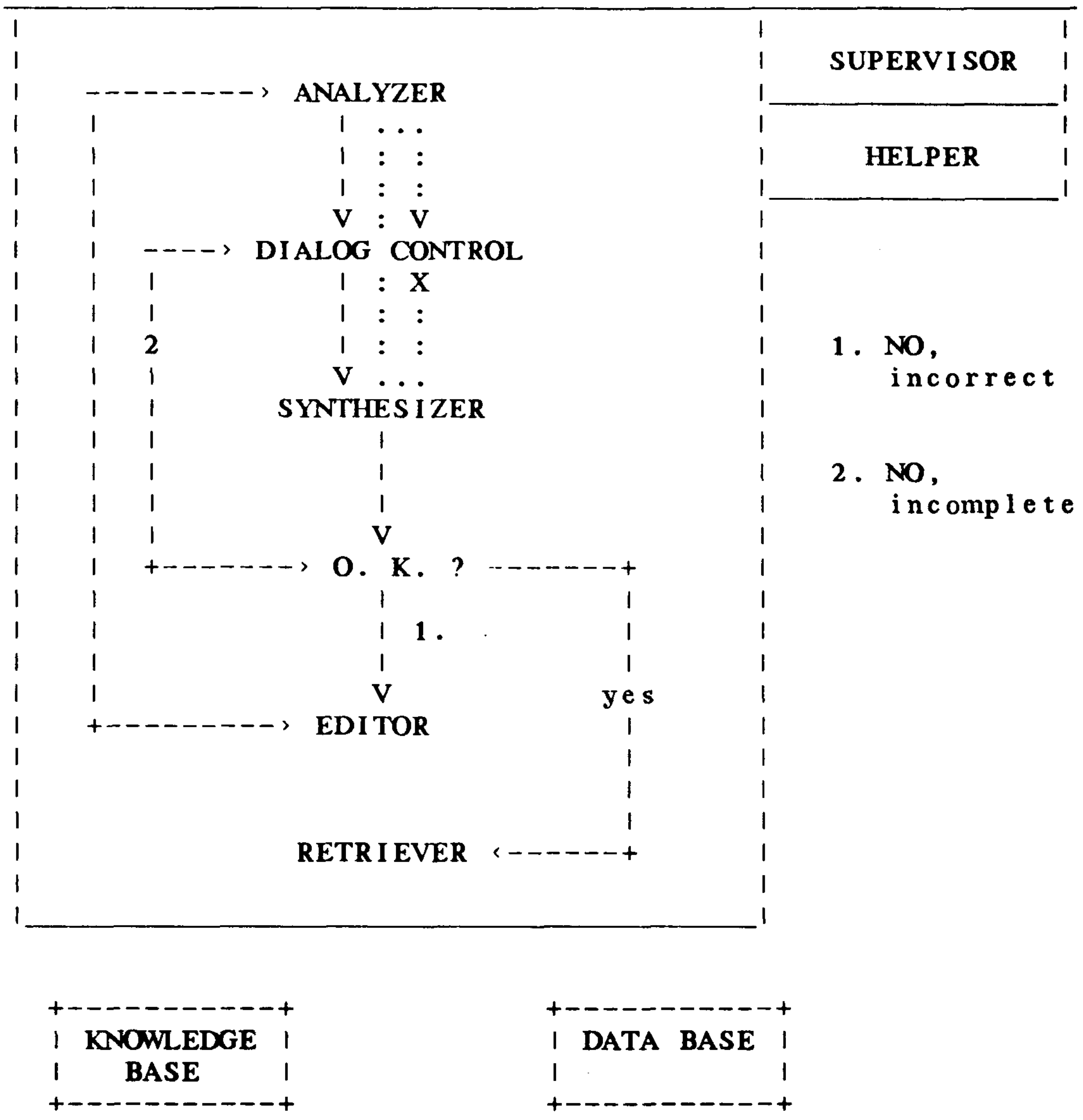
4.2.1 Analyzer

The analyzer, as mentioned in the previous section, has three major functions:

(1) To translate as much as possible of the user's query into a data base sublanguage.

(2) To discover and analyze the translation difficulties, as well as generate sufficient parameters to enable the dialog control to piece together dialog appropriate to the state of the translation and the state of the user interaction.

(3) To provide interpretation of the user's response.



```
 ________________________________________________________________________
|                                                     |                  |
|                                                     |   SUPERVISOR     |
|    ----------> ANALYZER                             |__________________|
|    |             | . . .                            |                  |
|    |             | :   :                            |     HELPER       |
|    |             | :   :                            |__________________|
|    |             V :   V                            |
|    |    ----> DIALOG CONTROL                        |
|    |    |        | :   X                            |
|    |    |        | :   :                            |
|    |    2        | :   :                            |    1. NO,
|    |    |        V . . .                            |       incorrect
|    |    |      SYNTHESIZER                          |
|    |    |           |                               |
|    |    |           |                               |    2. NO,
|    |    |           |                               |       incomplete
|    |    |           V                               |
|    |    +-------> O. K. ? -------+                  |
|    |                |            |                  |
|    |                | 1.         |                  |
|    |                |            |                  |
|    |                V           yes                 |
|    +----------> EDITOR           |                  |
|                                  |                  |
|                                  |                  |
|                                  |                  |
|               RETRIEVER <--------+                  |
|                                                     |
|_____________________________________________________|


   +-----------+                    +-----------+
   | KNOWLEDGE |                    | DATA BASE |
   |   BASE    |                    |           |
   +-----------+                    +-----------+
```

FIGURE 4-1 A Block Diagram of a Natural Language Query System

The way an analyzer perform these functions is based on the implementation of the system. In general, the operations of the analyzer have been implemented as two types of analysis, namely, the syntactical analysis, which is used to reduce redundancies, and the semantic interpretation. For example, in RENDEZVOUS, the concept of transformational grammars [Barr 81; Feigenbaum, 81] is applied in the syntactical analysis: the analyzer, first, by using standard suffix transformations, drops redundant words such as "please" and transforms the words in the user's query into a normalized form independently; then, the phrase transformer recognizes the types of phrases and replaces them by phrases which are syntactically closer to the formal query language. In the interpretation phase, some of the phrase transformation rules, which are called "semantic templates", are applied to check the semantic compatibility of phrase components with one another by using the data description as the guide.

In the PLANES system, the above operations are further divided into four steps: first, the words of the user query are evaluated and substituted with the canonical words and synonyms, redundant words being reduced and ignored; second, the semantic ATNs (Augumented Transition Networks) are applied to the input request in order to determine the specific meaning of each phrase; third, concept case frames are applied to the above phrases in order to find the pattern of the question; and finally, the filled-in concept case frame is translated into a formal query expression for use with the relational data base system in the information retrieval phase.

4.2.2 The Dialog Control

The Dialog Control takes the analyzer output, tests its logical completeness, and poses only those questions to the user that will yield a logically complete formal query. In order to produce a logically complete formal query for the information retrieval phase, the dialog control needs to determine certain tactics to allow the system to interact with the user so that the ambiguity involved in the user query can be clarified.

4.2.3 The Synthesizer

The Synthesizer translates the logically complete formal query output from the Dialog Control into a precise natural language re-statement. The operation of the synthesizer is the process of text generation [Barr, 81; Feigenbaum, 81]. The implementation of this process employs essentially the same concepts and grammar rules as the analyzer.

4.2.4 The Retriever

This component is invoked only when the user has approved the re-statement. After the retriever is invoked, it takes the logically complete formal query from the Dialog Control and retrieves data from the data base in order to generate an answer to that query. This retrieved answer is output to the synthesizer which then translates it into a precise natural language statement to the user. Thus, the major operation of the retriever is data base search. To process this operation, it is necessary to construct a set of search programs (generally written in LISP, APL or Pascal). By applying this set of search programs, the system performs following operations sequentially:

(1) Selecting the relations or files to be searched.

(2) Selecting an order for searching these files.

(3) Generating an expression for testing and selecting tuple values to return while searching.

(4) Generating a program to combine data, possibly from a number of different relations, so that the proper answer will be returned.

(5) Deciding when to save the results of a search for

future use.

In addition to the above four major components, there are some equally important components that need to be implemented in the development of a natural language query system. The main function of these components is to provide support to the user, analyzer, dialog control and synthesizer during their interactions. These components are briefly described as follows:

(1) KNOWLEDGE BASE: This provides time-independent semantic and linguistic information about the data base to allow the system to interact with reasonable intelligence concerning any natural language query whose formal counterpart lies within the class of formal queries supported as an internal interface (e.g. transformation rules for words and phrases).

(2) EDITOR: This provides a menu for the user to change his query or the system's version.

(3) HELPER : This provides a menu for the user to obtain general information about the kinds of data stored in the data base.

(4) SUPERVISOR: This component is responsible for invoking

each of the components cited above whenever appropriate.

The most important function provided by a natural language query system is that the system completely protects casual users from any data sublanguage; on the other hand, experienced users may still use the data sublanguage defined by the system to process their query.

5. NATURAL LANGUAGE QUERY SYSTEM FOR MADAM

MADAM is a bibliographic information storage and retrieval system developed within the DBMS Project of the Computer Science Department at the University of Southwestern Louisiana [Dominick, et. al., 80]. This system includes three major subsystems: Data Base Search Subsystem, Data Base Administration Subsystems, and System Administration Subsystem.

Prior to communicating with the MADAM system for information needs, users have to learn to handle a set of system commands. This requirement may keep casual users away from using such a system, or limit their use of the system. Thus, to serve this user group, the development of a natural language query system for MADAM is necessary. In this section, a framework for such a subsystem within the MADAM system is proposed based on the discussion in the previous sections.

In the framework presented here, the application of the natural language query sub-system is restricted and prepared only for the Data Base Search Subsystem of MADAM. The major reasons for imposing such a restriction are as follows:

(1) Among the three subsystems within MADAM, the users of the Data Base Administration and System Administration Subsystems are typically information system and database professionals. Those who can use only the Data Base Search Subsystem may be the users who need a natural language query system.

(2) The currently developed natural language query systems, such as RENDEZVOUS, PLANES and LUNAR, are mainly concerned with fact retrieval rather than updating data values or modifying data bases. The major reason for this development approach is that data base search is the primary function of any information system [Wiederhold, 83]. Thus, it is reasonable to develop a natural language query system for the Data Base Search Subsystem at the pilot stage.

The framework proposed here is organized into two major parts. The first part specifies four levels of the information retrieval process and describes required modifications to the existing MADAM system; the second part suggests a development approach for the natural language query system.

5.1 Four Levels of the Information Retrieval Process

It is helpful to describe how the system understands a user's query in the form of natural language, processes it, produces correct answer, and generates response to the user in the form of natural language. In this framework, the above process can be divided into four levels (see Figure 5-1). In the discussion on this subject, the first two levels are of primary concern since the introduction of these two levels would provide MADAM with the capability to understand and answer queries in the form of natural language in addition to formal query language commands.

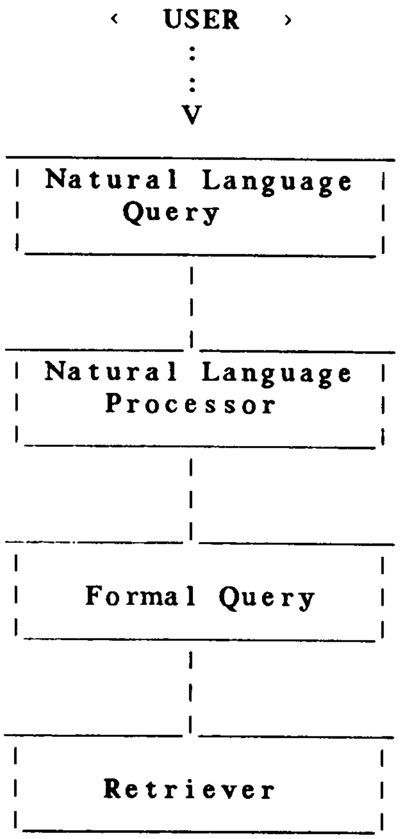


FIGURE 5-1 Four Levels of the Information Retrieval Process

5.1.1 Natural Language Level

This level is the interface level of the casual user-system interaction. At this level, the user and the system communicate with each other in the form of natural language (dialog and/or multiple-choice questions). Each information retrieval activity is initiated when a user inputs his query and is actually terminated when the system's response is accepted by its user at this level.

The techniques involved in the design of this level is the capability of the system in translating natural language input into a system defined formal language and that of the natural language output generation. Each time the system generates a re-statement of the user's query, it should output the re-statement along with the user's query so that the user may compare his own query with the system's perception of his query. If the system finds any ambiguity in the user's query, it may generate questions in the form of a multiple-choice questions along with some hints and the user's source statement so that the user may review his intention and modify the query statement using the assistance provided by the system.

5.1.2 Natural Language Processor

The user query in the natural language query system is non-procedural and informal; on the other hand, the commands used in the Data Base Search Subsystem of MADAM are non-procedural, but highly structured and formal. Thus, it is necessary to have a natural language processor which can perform the syntactical and semantic analysis on a natural language user query and generate a formal query acceptable by the existing system.

The design of such a processor, based on the discussion in the last sections and especially Figure 3-1 and Figure 4-2 should include the analyzer, the dialog control and the synthesizer, as well as other assistance components such as the knowledge base and helper.

The basic work of the analyzer is to transform the natural language query, syntactically and semantically, into one of three types of formal query expressions in the Data Base Search Subsystem by applying the grammar rules, relations, tuples and operators stored in the knowledge base. The synthesizer performs the same function in the reverse direction.

In designing these two components, as discussed in the previous sections, the designer should, first, construct a set of grammar rules based on the relations in the data base and the selection of appropriate techniques of natural language understanding, such as transformational grammars and case grammars; second, develop a knowledge base of sufficient size and capability. This knowledge base, which must be installed into the MADAM system will serve as the brain of the Natural Language Processor in natural language understanding. Thus, the design of such a knowldege base is not only important but critical to the success of the natural language query system.

Different from most developed natural language query systems, in which a relational calculus language, ALPHA, is applied as data sublanguage, the formal query applied in MADAM is a higher-level language. Thus, in the design of this phase, a two-step translation may be required. In this process, a relational calculus language may be used as an intermediate between natural language query and the data sublanguage used by the MADAM system.

The Dialog Control needs to check the logical completeness of a user query and pose questions to the user in order to organize a logically complete formal query. Thus, some semantic analysis techniques such as ATNs should be considered in the design of this component.

The helper component will be constructed by modifying the help commands in the MADAM Data Base Search Subsystem so that the result of each "help" command can be presented in a form of precise natural language.

Since the Natural Language Processor performs many functions and operations, it is necessary to have a driver which invokes the appropriate operation by initiating a specific subroutine mentioned above based on the state of the transformation process. Therefore, it is necessary to design a Supervisor routine to invoke and supervise various operations.

5.2 The Development Approach for the Natural Language Query System

It is predictable, in the development of a natural language query system within MADAM, that many problems and difficulties will be encountered, for example, how large of the knowledge base is required for such a system? Can the system being developed understand any natural language query? How many rules are needed to be generated for the syntactic and semantic transformation?

5.2.1 The Traditional Approach of Query System Design

Traditionally, commercially available database systems usually offerred comparatively general-purpose interfaces which were suitable only for "DB specialists". Under this approach, two basic assumptions on the users of computerized information system were widely accepted. The assumptions and their critiqueing references are:

(1) Whenever a user conceives a query, he can convey his intent to the system faithfully and precisely [Codd, 74].

(2) The user is willing to develop the appropriate skills and learn the "user-supports" to operate an information system [Eason, 75].

Since the user group of a natural language query system is casual users, the above assumptions are obviously invalid. For example, many researchers have found that, when using an information system, the casual user tends to expect a tool which fits his needs and does not expect to modify his own behavior to fit the system needs. Additionally, the requirement of learning formal query languages usually causes one of three responses:

(1) Non-use of the system.

(2) Limited use of the system.

(3) The use of a human intermediary.

Therefore, in order to develop a natural language query system which can provide an effective casual user-system interface, traditional thinking concerning the nature of users must be adjusted.

5.2.2 Experimental Approach

To adjust the traditional approach of query system development, this project proposes an experimental approach. The fundamental assumption of this approach is that, in an information system providing a natural language interface, the system has to "think" and "speak" as its user, a human being. Based on this assumption, the designer has to understand the mainstream concepts of behavioral science, such as learning psychology and human communication theories, as well as of natural language understanding and knowledge representation in the AI field.

Based on the concepts of behavioral sciences, it is reasonable to understand that it is impossible, at the early stage of development, to have a system whose "experience" and "intelligence" allows it to understand all the queries or issues of human conversation, and to use natural language fluently. On the contrary, a natural language query system should have a limited capability of both natural language understanding and natural language generation, and this capability should grow gradually by the increase of its "experience" and "vocabulary".

The experimental approach proposed in this project is based on the above consideration. The development of a natural language query system should be accomplished by consecutive experiments. In each experiment, the designer should restrict the scope of queries in limited topics and obtain certain distinct natural language requests in these topic areas through interviewing "sampled" casual users; then, develop a subset of a natural language query system which has the ability to "understand" these queries and "respond" to these queries. The developed system, then, should be integrated with previous developed subsets of the system. In such a way, the knowledge base of the system can be expanded as well as the system's ability to understand and use the natural language.

6. CONCLUSION

Computerized information systems are created for the benefit of the end users. According to Codd's estimate [Codd, 71], "by the mid 1990s, the home/casual user of such systems will be the dominant factor in the total utilization of database resources". If computerized information systems are to become everyday tools of casual users, the needs and desires of the casual users must be accommodated. Therefore, the development of natural language query systems must become a major trend in information system development.

REFERENCES

[Abrial, et.al, 74]. J. R. Abrial, "Data Semantics," in J. W. Klimbie and K. L. Koffemann(eds.), Data Base Management, North-Holland Co., 1974, pp. 1-59.

[Amble, et.al, 79]. T. Amble, K. Bratbergsengen, and O. Risenes, "ASTRAL: A Structured and Unified Approach to Data Base Design and Manipulation", in Bracchi and Nijssen(eds.), Data Base Architecture, North-Holland Co., Holland, 1979.

[Anderson, et.al, 78]. N. D. Anderson, and W. A. Burkhard, "MINISEQUEL : Relational Data Management System", in B. Shneiderman (eds.), Data Bases: Improving Usability and Responsiveness, Academic Press, London, 1978.

[Barr, 81]. Barr, Avron and E. A. Feigenbaum, The Handbook of Artificial Intelligence, vol. 1., Heuristech Press, Stanford, 1981.

[Boyce, et.al, 74]. R. F. Boyce, "Specifying Queries as Relational Expressions", in J. W. Klimbie and K. L. Koffemann (eds.), Data Base Management North-Holland Co., 1974.

[Bretmann, et.al, 79]. B. Bretmann, E. Falkenberg and R. Mauer, "CSL: A Language for Defining Conceptual Schemas", in Bracchi and Nijassen (eds.), Data Base Architecture, North-Holland, 1979.

[Codd, et.al, 71]. E. F. Codd, "A Database Sublanguage Founded on the Relational Calculus", Proc. ACM-SGFIDET Workshop on Data Description, Access and Control, Nov. 1971, ACM, N.Y., 35-68.

[Codd, et.al, 74]. E. F. Codd, "Seven Steps to Rendezvous With the Casual User," in J. W. Klimbie and K. L. Koffemann (eds.), Data Base Management, North-Holland Co., 1974, pp. 179-199.

[Codd, et.al, 78]. E. F. Codd, "How About Recently?", in B. Shneiderman (eds.), Data Bases: Improving Usability and Responsiveness, Academic Press, London, 1978, pp. 3-28.

[Daniels, et.al, 82]. D. Daniels, "An Introduction to Distributed Query Compilation in R*", in H. J. Schnerder (eds.), Distributed Data Bases, North-Holland Co., 1982, pp. 291-309.

[Date, 77]. C. J. Date, An Introduction to Database Systems (2nd ed.), Addison-Wesley Co., Reading, Mass., 1977.

[Dominick, 82]. W. D. Dominick, C. D. Michelson, and M. U. Farooq, MADAM User Guide, USL Computer Science Department, Lafayette, LA, 1982.

[Eason, et.la, 75]. K. D. Eason, L. Damodaran, and T. F. M. Steward, "Interface Problems in Man-Computer Interaction", in E. Mumford and Sackman (eds.), Human Choice and Computers, North-Holland Co., 1975, pp. 91-105.

[Ghosh, 77]. S. P. Ghosh, Data Base Organization for Data Management, Academic Press, New York, N. Y., 1977.

[Greenblatt, et.al, 78]. D. Greenblatt, and J. Waxman, "A Study of Three Database Query Languages", in B. Shneiderman, Data Bases: Improving the Usability and Responsiveness, Academic Press, London, 1978, pp. 77-97.

[Grosz, et.al, 77]. B. J. Grosz, "The Representation and Use of Focus in a System for Understanding Dialog", 5th International Joint Conference On Artificial Intelligence -1977, vol. 1, MIT, 1977, pp. 67-76.

[Haseman, 77]. W. D. Haseman, and A. B. Whinston, Introduction to Data Management, Irwin Inc., 1977, pp. 202-226.

[Jackson, 74]. P. C. Jackson, Introduction to Artificial Intelligence, Petrocelli/Charter, N. Y., 1974, pp. 169-342.

[Kalinichendo, 74]. L. A. Kalinichendo, and V. M. Ryvkin, "Problems of High Level Database Access Language Implementation in Inverted Storage Structure Environment", in Klimbie and Koffemann (eds.), Data Base Management, 1974, pp. 201-209.

[Lockemann, 75]. P.C. Lockemann, "Data Base User Language for the Non-programmer," University Karlsruhe: D-75 Karlsruhe 1, pp. 183-212, 1975.

[Merrett, 74]. T.H. Merrett, "The Extended Relational Algebra, a Basis for Query Languages," in B. Shneidermann (ed.), Data Bases: Improving Usability and Responsiveness, pp. 99-128, 1974.

[Nijssen, 74]. G.M. Nijssen, "Data Structuring in the DDL and Relational Model," in Klimbie and Koffemann (eds.), Data Base Management, pp. 363-379, 1974.

[Schank, et al, 1981]. R.C. Schank and C.K. Riesbeck, "Inside Computer Understanding," Hillsdale: Lawrence Erlbaum Asso., 1981.

[Spragne, 1980]. R.H. Sprague, "A Framework for the Development of Decision Support System," MIS Quarterly, pp. 1-26, Dec. 1980.

[Stacey, 74]. G.M. Stacey, "The Interface Between a Database and its Host Language," in Klimbie and Koffemann (eds.), Data Base Management, pp. 305-315, 1974.

[Waltz, et al, 77]. D.L. Waltz and B.A. Goodman, "Writing a Natural Language Data Base System," in 5th International Joint Conference on Artificial Intelligence-1977, MIT, pp. 144-50, 1977.

[Wedekind, 74]. H. Wedekind, "On the Selection of Access Paths in a Data Base System," in Klimbie and Koffemann (eds.), Data Base Management, 1974.

[Wiederhold, 83]. G. Wiederhold, Data Base Design (2nd ed.), N.Y.: McGraw-Hill, 1983.

[Weizenbaum, 66]. J. Weizenbaum, "Eliza: A Computer Program for the Study of Natural Language Communication Between Man and Machine," CACM, Vol 9, No. 1, pp 36-45, 1966.

[Winograd, 71]. T. Winograd, "Procedures as a Representation for Understanding Natural Language," MAC-TR-84. Cambridge, Mass.: MIT Project MAC. 1971.

[Winston, 79]. P.H. Winston and R.W. Brown, "Artificial Intelligence: An MIT Perspective, vol. 1, Cambridge: MIT Press, 1979.

[Woods, 73]. W.A. Woods, "Progress in Natural Language Understanding: An Application to LUNAR Geology," Proc. AFIPS 42, NCC, pp. 441-50. 1973.

[Zloof, 78]. M.M. Zloof, "Design Aspects of the Query-By-Example Data Base Management Language," in B. Shneidermann (ed.), Data Bases: Improving Usability and Responsiveness. London: Academic Press, pp. 29-53, 1978.

International Joint Conferences on Artificial Intelligence, 5th International Joint Conference on Artificial Intelligence-1977, vol. 1, Cambridge, Mass., 1977.

4.7

1. Report No. IN-82

2. Government Accession No. 183552 ~~147384~~

3. Recipient's Catalog No.

4. Title and Subtitle 48 P.

USL/NGT-19-010-900: CONCEPTS AND IMPLEMENTATIONS OF
OF NATURAL LANGUAGE QUERY SYSTEMS

5. Report Date DATE
June 1, 1984 OVERRIDE

6. Performing Organization Code

7. Author(s)

I-HSIUNG LIU

8. Performing Organization Report No.

9. Performing Organization Name and Address

University of Southwestern Louisiana
The Center for Advanced Computer Studies
P.O. Box 44330
Lafayette, LA 70504-4330

10. Work Unit No.

11. Contract or Grant No.
NGT-19-010-900

12. Sponsoring Agency Name and Address

13. Type of Report and Period Covered
FINAL; 07/01/85 - 12/31/87

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

Working paper series report surveying the state-of-the-art in natural language query systems for information systems, including issues related to hierarchies of user languages, query language analyzers, dialog controllers, and synthesizers, and implementation considerations for natural language query systems.

This report represents one of the 72 attachment reports to the University of Southwestern Louisiana's Final Report on NASA Grant NGT-19-010-900. Accordingly, appropriate care should be taken in using this report out of the context of the full Final Report.

17. Key Words (Suggested by Author(s))

Natural Language Query Systems for IS&R,
Information Storage and Retrieval Systems

18. Distribution Statement

19. Security Classif. (of this report)
Unclassified

20. Security Classif. (of this page)
Unclassified

21. No. of Pages
46

22. Price*

*For sale by the National Technical Information Service, Springfield, Virginia 22161