RESEARCH IN NATURAL LANGUAGE RETRIEVAL SYSTEMS*

Edward A. Stohr

Jon A. Turner

Yannis Vassiliou

Norman H. White

Center for Research on Information Systems
Computer Applications and Information Systems Area
Graduate School of Business Administration
New York University

Working Paper Series

CRIS #30

GBA # 82-10(CR)

This paper was presented at the Fifteenth Annual Hawaii International Conference on System Sciences, January 1982.

*This research is supported by a grant from the International Business Machines Corporation.

'Linguists are no different from any other people who spend more than nineteen hours a day pondering the complexities of grammar and its relationship to practically everything else in order to prove that language is so inordinately complicated that it is impossible in principle for people to talk,' Lanacker, Language and its Structure, New York: Harcourt Brace Jovanovich Inc., 1973.

1. INTRODUCTION

Is natural language an appropriate interface for a data base query system? Is it superior to formal query languages such as SEQUEL (1) or screen format approaches such as Query-by-Example (23)? Do we understand the complex semantic and syntactic transformations of human communication well enough to implement such systems? Would natural language be precise enough to eliminate erroneous responses from the data base? Are there classes of users and/or tasks for which natural language would be most appropriate? To paraphrase the title of the well-known paper by Hill (8); "Wouldn't it be nice if we could query a data base in ordinary English - or would it?"

In this paper we provide an overview of a research study which we hope will shed some light on these issues. Laboratory studies and field tests will be conducted using USL (User Specialty Languages)-an experimental information retrieval system currently under development at the IBM Heidelberg Scientific Research Center (10). Here we describe the USL system, and outline some major research questions and the strategy for conducting the research. Subsequent papers will provide detailed descriptions of each phase of the project and present the results.

2. THE USL SYSTEM

The study of computerized 'natural' language systems dates back to the early days of artificial intelligence research in the 1950's. After 20 or so years of effort we have not managed to build systems that can handle the full complexity of natural languages. However considerable progress has been made and we now appear to have the technology to build practical (though limited) systems.

Figure 1 shows some of the major areas of natural language research and references some representative systems, most of which have been experimental in nature. The subtree for inquiry systems is shown in more detail than the others because of its relevance to the present research. Within the area of information retrieval utilizing natural (or 'semi-natural') language interfaces, two different approaches have been adopted in an attempt to overcome the difficulty of understanding natural languages. The first, exemplified by such systems as BASEBALL (3) and LUNAR (21), is to restrict the domain of discourse and to build specialized systems using artificial intelligence techniques. This approach has been relatively successful but suffers from a high initial cost and lack of transportability. A second approach relies on a generalized data base management system (DBMS) which contains a description of the domain of discourse, performs the data retrieval function and allows the system developers to concentrate on the 'front-end' language translation interface. A DBMS has the further advantage that it is application independent.

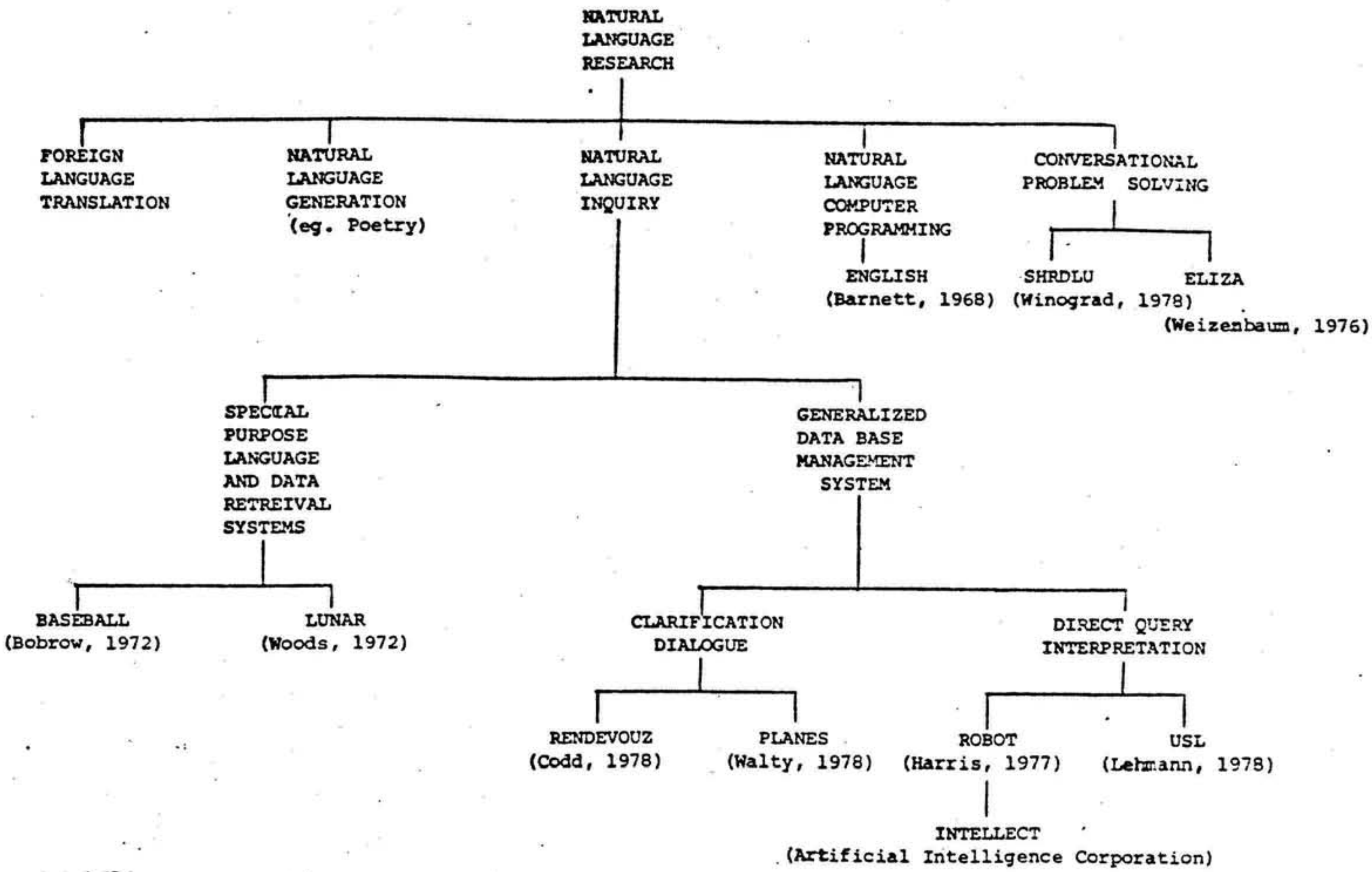

FIGURE 1

SOME DIRECTIONS IN NATURAL LANGUAGE RESEARCH

However this independence does not necessarily extend to the superimposed linguistic system.

The systems using a DBMS have had two varieties of interface. In RENDEZVOUS (4) and PLANES (18) 'clarification dialogue' has been used extensively to resolve ambiguities prior to the submission of a request to the DBMS. This tactic makes it easier to interpret retrieval requests correctly but slows down the interaction and requires that the system contains a language generation process as well as an interpretation process. The second variety of interface attempts an immediate interpretation of a retrieval request and only prompts the user for clarification when absolutely necessary. ROBOT (7), its successor INTELLECT (a product of Artificial Intelligence Corporation) and USL all use the latter approach.

INTELLECT utilizes ADABAS as its DBMS and is available commercially. The USL system uses a parser-generator (USAGE, (2)). Queries to USL are translated into the SQL query language and then processed by the underlying DBMS without further intervention.

Several approaches to the translation of natural languages have been attempted. The first builds upon a formal language vocabulary to give an English-like appearance. However it is still necessary for the user to think in terms of the formal language. A second approach is to build a model capable of representing (at least some of) the semantics of the real world problem situation. This is typically the approach adopted by artificial intelligence research employing such techniques as semantic nets, (5), frames, (12) and conceptual graphs (16). User queries are parsed and interpreted in terms of the model and then translated to data base accesses. A third approach is essentially linguistic. Here an attempt is made to gain an in-depth understanding of the semantic and syntactic systems underlying human speech and to translate these directly. This is the approach adopted by USL. The original grammatical studies were for the German language. The system now has Dutch and English grammars and a Spanish Grammar is being developed.

USL was designed to provide users having little or no computer experience with a means for accessing and analysing data. The users however should have a knowledge of their area of appplication. To achieve portability, the USL system has no knowledge of any subject matter that might vary from application to application. Instead it has a dictionary of general words (prepositions, conjunctions, the verbs 'to be' and 'to have', names of months, etc) and grammatic rules that allow it to interpret a subset of English. Language rules and word definitions that are application dependent must therefore be added by the system analyst for each separate application. Some of the definitions are of course contained in the data base schema. Others are added by expanding the set of grammatical rules that come with the system.

Figure 2 shows the major components of the USL system. A syntactic analysis is first performed by the parser using grammatical rules and definitions stored in both the USL and Application Dictionaries. Each grammatical rule references an interpretation routine which is executed if the rule is invoked. A successful parse generates a preliminary query string expressed in SQL. The

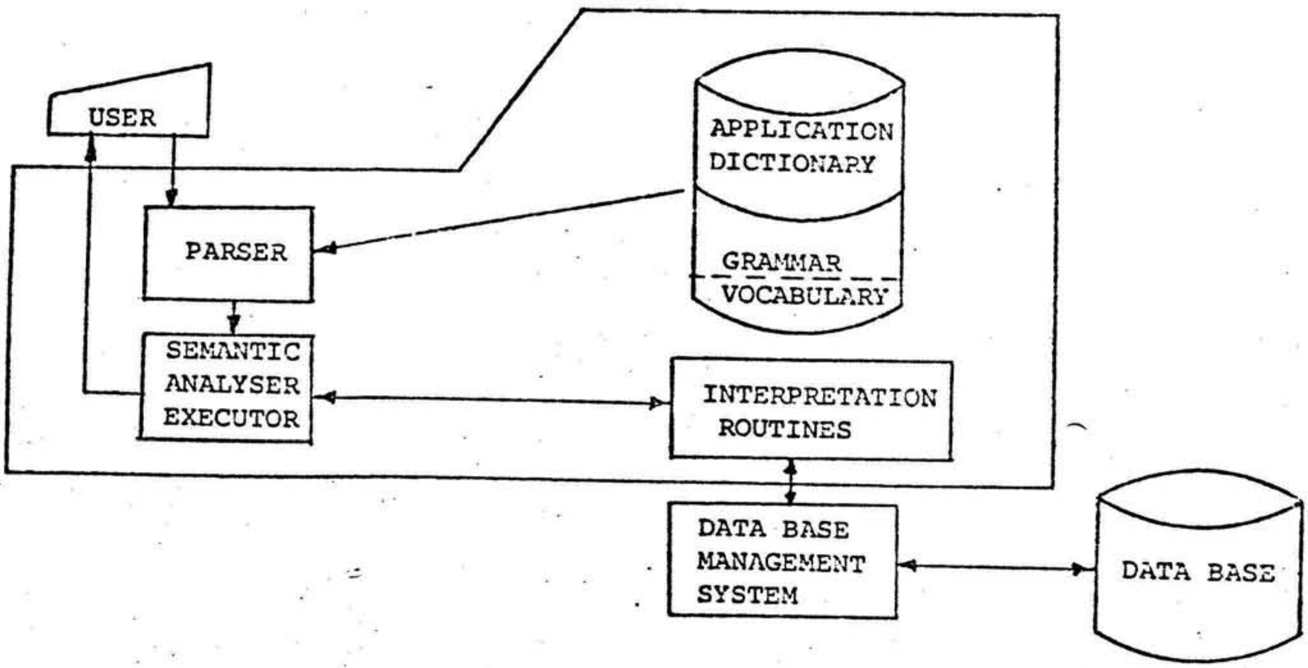

FIGURE 2
STRUCTURE OF USL SYSTEM

preliminary string is optimized to replace virtual by real relations, where posssible, and to eliminate unnecessary joins. The optimized SQL query is passed to the DBMS and the result returned to the user. As an example, for the Alumni data base to be described later, the query, 'How many alumni have no donations?', is translated to the SQL query:

```
SELECT COUNT (UNIQUE ID)
FROM ALUMNI WHERE SOURCECODE IS LIKE 'AL%'
AND ID NOT IN (SELECT ID FROM GIFTSUMMARY);
```

An interesting option of the USL system is that the SQL query can be printed at the user's terminal prior to accessing the DBMS. For users with some knowledge of SQL this mechanism provides a form of clarification dialogue. A similar tactic is used by INTELLECT which also echoes back a formal statement of the user's request.

The USL parser works in a left-to-right bottom-up fashion producing a parse tree reflecting the surface structure of the input sentence. The grammar is primarily context-free but there are some context-sensitive and transformational elements. In some (infrequent) cases more than one valid parse tree will be produced in which case several SQL queries are generated each producing a response from the DBMS. The user then has to choose the correct result.

Over 800 grammatical rules expressed in a modified BNF comprise the application-independent English grammar supplied with USL. Table 1 shows some of the linguistic capabilities of the system. The examples apply to the Alumni Data Base application that will form the basis for our field and laboratory tests (see Section 5). As is usual with natural language processing systems the capabilities in Table 1 are not always complete in the sense that some English language variants will not be succesfully parsed. The USL system also has some capabilites for:

(1) Updating a data base (using imperative English sentences rather than interrogative forms)

(2) Computations such as sum, average and count

(3) Defining and manipulating variables and functions

(4) Creating new relations.

It has no deductive capability beyond that required for language translation.

1. WH-QUESTIONS - Which/Who/What/When
   Who lives in New York? Which alumnus lives in New York?
2. YES/NO QUESTIONS
   Did Smith make a pledge?
3. COMMANDS
   List the French donors.
4. NEGATIONS
   Which donors did not give in 1981?
5. RELATIVE CLAUSES
   List the alumni who are members of the Tax Society.
6. ADJECTIVES
   Who are the inactive alumni?
7. GENETIVE ATTRIBUTES
   What is the address of Smith?
8. APPOSITIONS
   Which alumni of the school GBA live in New York?
9. AND-COORDINATION/OR-COORDINATION
   What is the name and address of Smith? Who lives in New York or Boston?
10. QUANTIFIERS - all/at least/how many...
    Who gave at least two donations? List all alumni who are finance majors.
11. COMPARISON - greater/more/less/how much...
    Who gave more than Smith?
12. POSSESIVE PRONOUNS
    Who donates more than his company?
13. LOCATIVE ADVERBS - lives in/at/from-to...
    Who lives at 40 West Street?
14. TIME ADVERBIALS - how long/when/before...
    When did Jones give a donation?
15. AGGREGATES - sum/average/largest/maximum...
    What is the average age of active donors? What is the sum of donations?
16. SUBTOTALS AND ORDINALS
    What is the sum of the pledges for each school? What are the 5 highest donations?
17. ARITHMETIC,DEFINITIONS OF VARIABLES AND FUNCTIONS
    What is (2 * 5) - 3?
    X = 1.5
    Y = the Pledges of Smith
    Store Y * r

TABLE 1
CAPABILITES OF USL

3. APPLICATION DEVELOPMENT IN USL

The USL system provides the capability of parsing natural language sentences together with a limited vocabulary of words commonly used in expressing queries in English. Application specific concepts and vocabulary must be added during the systems development process. These in turn have to be correlated with the contents of the relational data base. Words may represent the names of relations, the attributes (columns) of relations or specific values within tuples. Proper nouns and numbers are recognized by default as values in relations. Common nouns, verbs and adjectives are defined by association with the columns of real or virtual relations. This is done by establishing a 'view' or virtual relation for each concept (statement or assertion) necessary to the real application. Each column in a virtual relation has a defined 'domain' and 'role'. The standard domains are ZAHL (Number), WORT (word or character string), DATUM (Date, time of day) and CODE (numeric code). The standard roles are grammatical concepts such as NOM (nominative case), ACC (accusative case), DAT (dative case) and VON (genitive attribute) as well as time and place roles. The roles are associated with the columns by prefixing the rolename to the column name in the definition of the view. These relationships are best explained by an example. Suppose we have a "real" (or "base") relation:

GIFTSUMMARY (DONOR, AMOUNT, FISCALYEAR, ...).

The verb "give" can be defined in USL by first establishing a view using the statement.

```
DEFINE VIEW GIVE (WNOM-DONOR, ZACC-AMOUNT, DTA-FISCALYEAR)
AS SELECT DONOR, AMOUNT, FISCALYEAR FROM GIFTSUMMARY;
```

Here the prefix WNOM defines DONOR as a character string (W) in the nominative case (NOM). Similarly AMOUNT is defined in the accusative case with a number domain (Z). Finally FISCALYEAR is defined as denoting time (TA) with domain date (D).

Thus the assertion, "Donors give donations in a year", is established.

Continuing the above example, after this process has been completed USL will be able to interpret questions such as: "who gives what", "who gives what when", "Did Smith give 5000?", "How much did Smith give in 1981", "Did Smith give more than Jones in the year 1981?"

In addition to defining the views and rolenames the system developer must expand the definition of the vocabulary for the application by defining: (1) non-standard plural-forms for nouns, (2) non-standard verb tenses, (3) prepositions used with nouns and other surface structure contextual associations, (4) synonyms. A prompting program is available to assist in this task. Each such definition results in a grammatical rule in the application dictionary.

This explanation of the systems development process has been of necessity very brief. However, it can be seen that the use of USL requires additional work for the system developers (data administrators) beyond that required to establish the DBMS. For the Alumni application to be described in Section 5 approximately 150 views and 350 grammatical rules have been defined. However, the primary difference between developing a data base with a formal language interface and developing one with a 'natural' language interface lies in the need to discover how users refer to data in their written and verbal communications. While the data administrator need not (it seems) be an expert linguist, some training will probably be required. Finally, the linguistic aspects of the application have to be taken into account through all stages of the data base design since some relational schemas may be more suitable from the linguistic point of view than others.

Although of subsidiary importance to the major research questions to be described in subsequent sections of this paper this study should help to establish some of the important considerations in designing natural language query systems.

4. MAJOR RESEARCH QUESTIONS

Research in 'semi-natural language' front-ends to data base management systems (SNL's) has reached a point where technical feasibility may no longer be a major issue. As discussed in the previous section, there are now several prototype experimental systems and at least one commercially available system. The major question to be resolved is 'has it been worth the effort?' - will such systems gain user acceptance and will they eventually replace, or at least supplement, more formal language approaches? This can only be resolved by suitable experiments and evaluations of the use of such systems in practical applications (22).

In this section we list some arguments and counterarguments that are commonly made for and against natural language based systems, review some related experimental reseach work and formulate a simple representation of the retrieval process.

Discussions of the arguments for and against natural language systems appear in, for example, (13), (22), (7) and (15). Briefly the arguments <u>for</u> the use of natural language are as follows:

F1. Humans already know natural language so that many people who would not invest the time and energy to learn a formal language may be willing to use an SNL.

F2. The use of an SNL should reduce the burden of remembering or relearning formal language conventions after periods of disuse. Even data processing professionals perform some functions only occasionally and could avoid the need to consult reference manuals if an SNL were available.

F3. Often the data to be manipulated by the machine is in natural language form. A natural language understanding system would avoid the painstaking and error-prone task of translating such data into the formats required by current data base systems.

F4. The conceptual structure underlying human thought and communication has evolved over centuries of use to be an efficient mechanism for conveying complex ideas. It is unlikely that we could develop an artificial structure that would be as 'user friendly'.

F5. Current SNL retrieval systems produce acceptable error rates without rephrasing and have fast enough response times to make them commercially feasible.

The arguments against the use of SNL retrieval systems can be summarized as follows:

A1. Natural language is much less precise than a formal language. Using an SNL would nullify the major advantages (speed and precision) afforded by computers. It will also lead to unresolvable ambiguities and/or possible errors due to misunderstanding.

A2. The rigor of a formal notation system aids users by forcing them to think more clearly about their problem. Formulation is the most difficult aspect of problem solving; coding into a formal language is relatively simple.

A3. Unrestricted natural language systems are currently infeasible and likely to remain so for the foreseeable future. The subsets of natural language supported by SNL's may have as many rules and exceptions to be learned as formal languages. Moreover learning and retention may be impaired because of the interference affect of natural language knowledge.

A4. The additional development costs, the need for clarification dialogue and processing overheads will make SNL's less cost-effective than formal languages.

A5. The use of an SNL may lead to unrealistic expectations of the computer's power and to questions that no computer system can answer.

A6. Existing formal language systems are easy to learn and use and adequate for most purposes.

Some of the above claims and counterclaims appear to be contradictory. This is so for example with respect to:

(1) the precision of SNL's for expressing complex retrievals (F5 versus A1)

(2) the ease of learning and retention (F1 and F2 versus A3)

(3) the benefit or otherwise of encoding queries from 'natural' to 'artificial' formats (F4 versus A2).

The arguments F1 and F2 together with A4 above have led a number of observers (for example (15)) to speculate that SNL's will be useful, if at all, only for situations characterized by a fairly large number of casual (non-dataprocessing) users who have a good knowledge of the semantics of their application. A field test aimed at providing a partial test of this thesis will be conducted during the present research study (see Section 5).

We now review some experimental research on both formal retrieval languages and SNL's. Although some of the results are presented in a tabular format it must be emphasized that the experiments are not comparable because of wide disparities in such important experimental variables as ability and background of subjects, training techniques, complexity of the sample application and 'level' of language taught. The experiments with formal languages have mostly involved pencil and paper exercises using student subjects. Although the experimental designs are varied and have been concerned with many different aspects of language design the primary performance measures are the mean percentage of correctly coded queries and (sometimes) the mean time to formulate a query. Typical results for three such experiments (averaged over all classes of subject and degrees of query difficulty) are shown in Table 2.

Reference*	Query Language	Instruction Time (hours)	Average Formulation Time (minutes)	Percentage of Accurate Queries
a.	SEQUEL	10:00-12:00		71.2
	SQUARE			66.2
b.	QBE	1:35	0.9	75.2
	SEQUEL	1:40	2.5	72.8
	Algebraic	2:40	3.0	67.7
c.	QBE	2:00-3:00		67.0

TABLE 2
FORMULATION TIME AND QUERY ACCURACY-
EXPERIMENTS ON FORMAL LANGUAGES

*References
a = Reisner, 1977 (14)
b = Greenblatt and Waxman, 1978 (16)
c = Thomas and Gould, 1975 (17)

These results seem to be both encouraging and discouraging. On the one hand the figures can be taken as showing that a reasonable proficiency in the various languages can be attained after a relatively short training period. On the other hand the percentage of correct queries obtained is not all that high. Some results that seem to be common across these and other similar experiments are:

R1. Performance differences between individuals are very high (for example from 33% to 93% correct answers in the Query-by-Example Study).

R2. Programmers perform better than non-programmers under some conditions.

R3. The number of errors and the time to formulate a query increase with the complexity of the retrieval problem.

R4. Many of the errors that were made were of a minor clerical nature (some 50% in the SEQUEL-SQUARE study). Many of these involved spelling and/or the inaccurate specification of data base attributes.

Because of their recent advent there have been far fewer results with SNL's and what has been reported is even less conclusive. For the ROBOT system, Harris (7) reports that experienced users in a commercial application were achieving a 90% level of accceptance for queries. Harris also reports that the time to build applications using ROBOT is 'in the order of a week' and that the average computer response time on a sample of questions was approximately 10 seconds.

A similarly low rate of errors was found in an evaluation of the USL system in a field setting (9). In this study a sample of 2,214 queries made by a single user was found to have an average error rate of only 6.6%. Two other real-life applications of USL are described in (11) which analyses the kinds of queries and errors which were encountered. The average time to develop the linguistic portion of the applications was approximately two weeks.

About all that can be said about these experiments is that some users may be able to work comfortably with a subset of natural language after a suitable period of practice. Whether acceptable error rates can be achieved by most people, and in particular, inexperienced casual users, is still opened to doubt. This is particularly true in the light of some less favorable experiences with other (non-data base oriented) natural language systems.

Figure 3 presents a simplified view of query processing which recognizes that it may be necessary to reformulate the same query several times before a correct solution is obtained and also that a user may sometimes give-up without obtaining a successful answer. The acceptability and ease of use of a retrieval system depends on the efficiency and psychological impact of all the processes shown in the figure. If the arguments A1 through A4 above (concerning the imprecision and ambiguity of natural languages and the restricted nature of SNL's) are correct we should expect, on the average, to see more iterations per query when using an SNL than a more formal query language. This would be mitigated by the extent to which argument F4 (concerning the difficulty users may have in translating from a 'natural' to a 'formal' query statement) is true. The use of an SNL may also reduce the tendency for clerical errors (see R4 above):

a) The vocabulary recognized by an SNL can be fitted to that customarilly used in the application thus reducing the need for users to memorize and understand the naming and other conventions required by the data base management system.

b) Natural language queries are usually shorter than their formal language equivalents (see for example the sample query in Section 2).

Finally, and most importantly, the relative difficulty of expressing queries in the natural and formal languages will influence the error rate. It seems likely that very simple queries can be expressed equally well in either format. However some queries (see

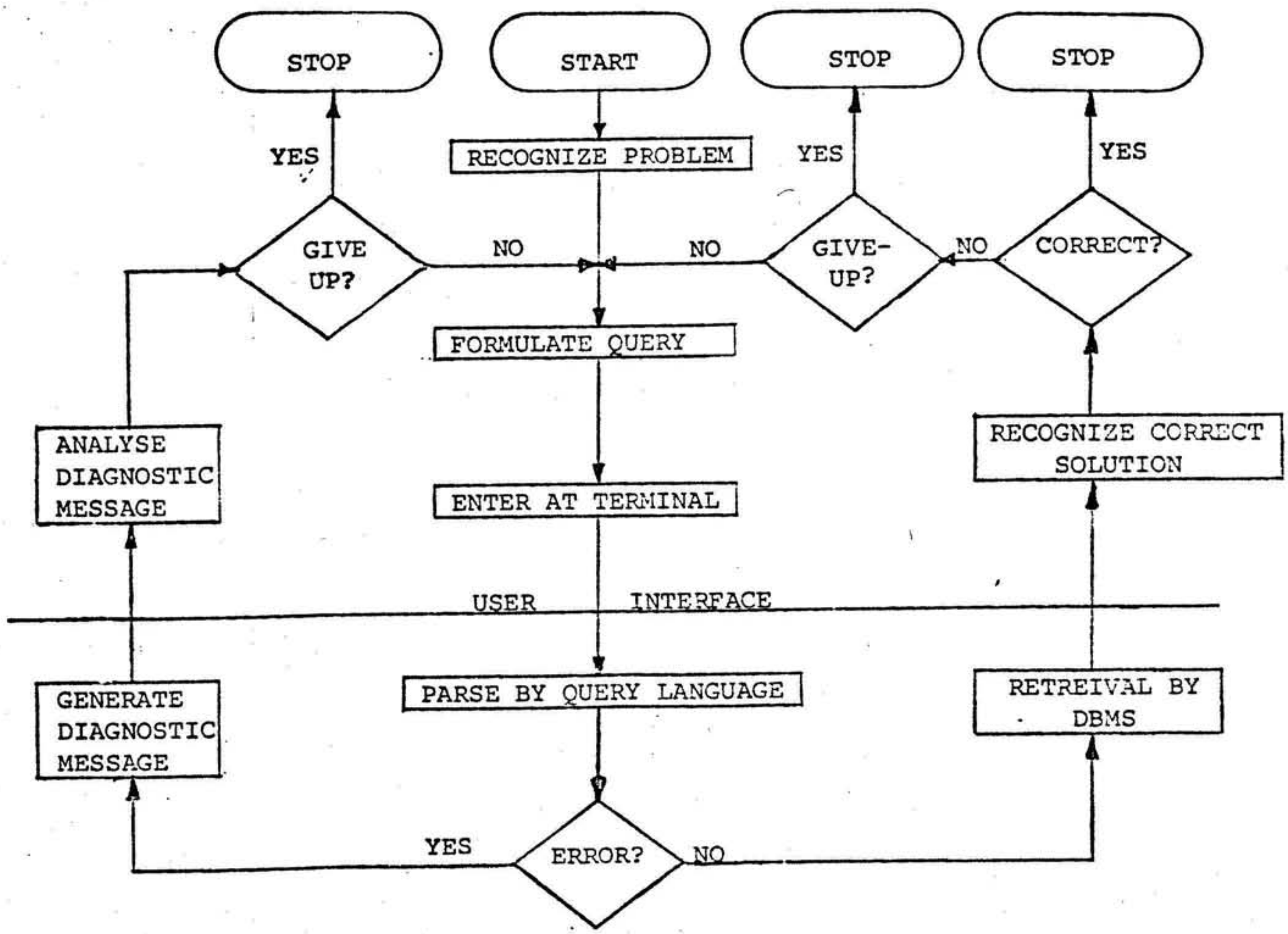

FIGURE 3
SIMPLIFIED VIEW OF QUERY PROCESSING

again the example in Section 2) can be expressed more simply in an SNL, while others (depending on the power of the SNL) may not be expressable at all. While it should be faster for users to express their queries in English rather than in a formal language (see Table 2 above) the overall time per succesful query will depend on the number of reformulations necessary to obtain an English-like query that is accepted by the SNL.

5. RESEARCH DESIGN

The Advanced language Project will incorporate two linked experiments, one in an actual field setting with 'live' data and 'real' users and one in a laboratory setting. The major purpose of the field experiment is to determine the suitability of the USL system for casual users who know the details of their application (see the discussion in Section 4). Data will be gathered on types of queries and their complexity, types of errors and their frequency, classes of users and their attitudes toward the system and so on.

The major purpose of the laboratory experiment is to test the relative performance of a semi-natural language (USL) versus a formal structured language (SQL) in a controlled environment. This experiment will follow the same general outline as some of the experiments carried-out by other researchers as discussed in Section 4. However some qualitative differences must be incorporated in the design because of the nature of an SNL - for example training procedures and query analysis will require special treatments. Data will be gathered on the student subjects and their background, on the length of time required to obtain different levels of proficiency in each language, on the relative times to formulate correct queries, relative error rates and so on.

Both experiments will utilize the same data base. The chosen application involves the fund-raising activities for the Graduate School of Business Administration (GBA) at New York University (NYU). The data base contains approximately 40,000 records extracted from an existing NYU Fund- Raising System. This is organized as five relations containing respectively, information identifying alumni and other donors, their educational background, their giving histories, detailed gift transactions and a 'data dictionary' relation that explains all data items and their codes.

The primary users in the field experiment will be officers of the GBA Development Office and of the NYU Alumni Federation. None of these users have had any prior computer experience. It is anticipated that the data base will be used primarily in a decision support mode for planning fund-raising appeals. However some queries related to controlling current operations are also anticipated.

Since the 'real' user population is limited it will be augmented by eight user assistants or 'chauffeurs' whose background and training can be controlled by the experimenters. During the first two months of the field test the chauffeurs will act as intermediaries. The users will explain their information requirements to the chauffeurs who will then formulate and enter the queries into the computer. Four chauffeurs will be trained in SQL and the other four in SQL. They will work in pairs: each user query will be answered by one chauffeur using USL and another using SQL. After several weeks the USL chauffeurs will be trained in SQL and the SQL group in USL. At the end of a similar period of time all chauffeurs will be allowed to user the language of their choice. Finally, the chauffeurs will be withdrawn, the 'real' users will be trained in USL and allowed to enter their own requests at the terminal.

The use of paid subjects in this way has several advantages. First, it increases the number of subjects tested. Secondly, it allows us to perform a partly controlled experiment within an otherwise uncontrolled field environment. Finally, it will enable us to test training techniques and to tune the data base and linguistics in a real environment prior to their use by the inexperienced end-users.

The laboratory experiments will utilize paid student subjects in a 'crossed' design similar to that employed with the chauffeurs. Additional memory retention tests will be given in both languages at the end of the experiment.

Since the field and laboratory experiments will be carried-out using the same data base during the same time period (approximately six months) the results of one can be used to help callibrate the other. For example actual queries from the chauffeur-driven portion of the field test will be included in the test given to the laboratory subjects.

REFERENCES

1. Astrahan, M.M., and Chamberlin, D.D., 'Implementation of a Structured English Query Language', Comm. ACM, Vol 18, No 10, October (1975), 580-588.

2. Bertrand, O. and Daudenarde, J.J., 'An Interactive Program Generator from Grammatical Description', IBM France Scientific Center, Doc F.008 (1980).

3. Bobrow, D., 'Natural Language Input for a Computer Problem Solving', in M. Minsky (Ed.), Semantic Information Processing, MIT Press, Cambridge, Mass (1972).

4. Codd, E.F., 'How About Recently? (English Dialogue with Relational Databases Using RENDEZVOUS Version 1)', In B. Shneiderman (Ed.), Databases: Improving Usability and Responsiveness, Academic Press, New York, (1978), 3-28.

5. Findler, N.V., ed. Associative Networks: The Representation and Use of Knowledge in Computers, Academic Press. New York, (1979).

6. Greenblatt, D. and J. Waxman, 'A Study of Three Database Query Languages', In B. Shneiderman (Ed.), Databases: Improving Usability and Responsiveness, Academic Press, (1978), 77-97.

7. Harris, L.R., 'User Oriented Data Base Query with the ROBOT Natural Language Query System', International Journal of Man-Machine Studies, 9, (1977), 679-713.

8. Hill, I.D., 'Wouldn't It be Nice If We Could Write Computer Programs in Ordinary English -- Or Would It?', Honeywell Computer Journal, 6,2, (1972), 76-83.

9. Krause, Jurgen, 'Preliminary Results of a User Study with the 'User Specialty Languages' System and Consequences for the Architecture of Natural Language Interfaces', IBM Heidelberg Scientific Research Center, TR79.04.003, May, (1979).

10. Lehmann, H. 'Interpretation of Natural Language in an Information System', IBM Journal of Research and Development, vol.22, no.5, September, (1978).

11. Lehmannn, H., N. Ott and M. Zoeppritz, 'User Experiments with Natural Language for Data Base Access', Proc. 7th International Conference on Computational Linguistics, Bergen (1978).

12. Minsky, Marvin, 'A Framework for Representing Knowledge', in Winston, P.H. Ed., The Psychology of Computer Vision, New York: McGraw Hill (1975).

13. Petrick, S.R., 'On Natural Language Based Computer Systems', IBM Journal of Research and Development, 20, 4, (July 1976), 314-325.

14. Reisner, Phyllis, 'Use of Psychological Experimentation as an Aid to Development of a Query Language', IEEE Transactions on Software Engineering, SE-3, 3, (1977), 218-229.

15. Schneiderman, Ben, Software Psychology: Human Factors in Computer and Information Systems, Winthrop, Cambridge, Mass. (1980).

16. Sowa, John F., 'Conceptual Graphs for a Data Base Interface', IBM Journal of Research and Development, vol 20, no.4, July (1976).

17. Thomas, J.C. and J.D. Gould, 'A Psychological Study of Query by Example', Proceedings of the National Computer Conference, 44, AFIPS Press, Montvale, New Jersey, (1975).

18. Waltz, D., 'An English Language Question Answering System for a Large Relational Database', Communication of the ACM, 21, 7, (July, 1978), 526-539.

19. Weizenbaum, J., 'ELIZA -- A Computer Program for the Study of Natural Language Communication Between Man and Machine', Communications of the ACM, 9, 1, (January, 1966), 36-45.

20. Winograd, T., Understanding Natural Language, Academic Press, New York, (1972).

21. Woods, W.A., 'Progress in Natural Language Understanding - An Application to Lunar Geology', Proceedings National Computer Conference, 42, AFIPS Press, Montvale, New Jersey, (1973), 441-450.

22. Woods, W.A., 'A Personal View of Natural Language Understanding', SIGART Newsletter No.1, (February, 1977).

23. Zloof, M.M., 'Query-by-Example: A Data Base Language', IBM Systems Journal, 16,4, (1977), 321-343.