AD-A043 488 PATTERN ANALYSIS AND RECOGNITION CORP ROME N Y F/G 9/2
EVALUATION OF RELATIONAL INTELLIGENCE ANALYSIS SYSTEM.(U)
JUL 77 C P MAH F30602-75-C-0241
UNCLASSIFIED PAR-77-7 RADC-TR-77-255 NL

| OF |
AD
A043488

END
DATE
FILMED
9 -77
DDC

RADC-TR-77-255
Final Technical Report
July 1977

EVALUATION OF RELATIONAL INTELLIGENCE ANALYSIS SYSTEM

Pattern Analysis and Recognition Corporation

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Services (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

This report has been reviewed and approved for publication.

APPROVED:

ROBERT N. RUBERTI
Project Engineer

APPROVED:

ROSS H. ROGERS, Colonel, USAF
Chief, Intelligence & Reconnaissance Division

FOR THE COMMANDER:

JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (DAP) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: RADC-TR-77-255
2. GOVT ACCESSION NO.:
3. RECIPIENT'S CATALOG NUMBER:
4. TITLE (and Subtitle): EVALUATION OF RELATIONAL INTELLIGENCE ANALYSIS SYSTEM
5. TYPE OF REPORT & PERIOD COVERED: Final Technical Report
6. PERFORMING ORG. REPORT NUMBER: PAR Report #77-7
7. AUTHOR(s): Dr. Clinton P. Mah
8. CONTRACT OR GRANT NUMBER(s): F30602-75-C-0241
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Pattern Analysis & Recognition Corporation, 228 On the Mall, Rome NY 13440
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 62702F, 45940211
11. CONTROLLING OFFICE NAME AND ADDRESS: Rome Air Development Center (IRDT), Griffiss AFB NY 13441
12. REPORT DATE: July 1977
13. NUMBER OF PAGES: 39
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Same
15. SECURITY CLASS. (of this report): UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A

16. DISTRIBUTION STATEMENT (of this Report): Approved for public release; distribution unlimited.
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report): Same
18. SUPPLEMENTARY NOTES: RADC Project Engineer: Robert N. Ruberti (IRDT)
19. KEY WORDS (Continue on reverse side if necessary and identify by block number): Intelligence data analysis; English query languages information processing; relational data management
20. ABSTRACT (Continue on reverse side if necessary and identify by block number):

The Rapidly Extensible Language (REL) system under evaluation here allows a user to define a "natural" computer programming language by extending an English-like base language through dynamic definitional facilities. The scheme seems especially promising for science and technology intelligence data analysis at FTD because of a need to let analysts with relatively little computer experience work interactively with on-line data through a conversational system. The present study finds that REL meets this need partially; where it tends to fall short, recommendations are made for future

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) development of possible remedies.

TABLE OF CONTENTS

EVALUATION

An evaluation study of the REL system has been completed. It addresses the applicability of REL as an aid to the scientific and technical intelligence analyst. In general, there appear to be several advantages it offers particularly because of the unpredictable modes of access to data in scientific intelligence analysis. The principal disadvantage pointed out in the report is that the prototype system has not been completed. Since the completion of this report a successful prototype has been demonstrated. Continued work is being planned to experiment with the prototype with scientific intelligence data. This effort is included as part of TPO Thrust R3D.

Robert N. Ruberti

ROBERT N. RUBERTI
Project Engineer

SECTION 1

INTRODUCTION

1.1. OBJECTIVES

The original objective of the effort covered by RADC Contract F30602-75-C-0241 was to evaluate the effectiveness of an experimental data analysis system at FTD consisting of a version of the REL natural language processing system and data bases on Soviet aircraft and on Soviet personalities. Because of unexpected circumstances, however, the system was never brought up at FTD; and so there was no opportunity to measure its efficiency in handling fairly large volumes of data or to observe its usefulness for persons actually carrying out intelligence tasks. This report will be limited to an evaluation based on materials describing the general operation of REL. A more complete evaluation will have to wait until an FTD system is actually available for testing.

The main question to be addressed here is how well a REL system could in principle help out science and technology intelligence analysts. The report first will consider the reasons for having an intelligence data analysis system in the FTD situation. It will then examine how an implementation of a simple relational data base with a REL English query language might serve as such a system; some issues to be covered in this regard will include the value of a natural language user interface, the implications of a relational data representation for storage and retrieval of information, and the appropriateness of various sorts of data analysis in an interactive system with many unsophisticated users. Recommendations for further development of the REL system in light of science and technology intelligence needs will be made at the conclusion.

1.2. BACKGROUND

For intelligence information to serve a purpose, it must ultimately be transmitted to interested users in a timely way. At present, printed material is the standard media for such information, but it is far from ideal. To begin with, printed matter quickly gets out of date in areas like science and technology; and updating of this information tends to be slow because of the time needed to compile, to edit, to publish, and to distribute changes. Furthermore, the collected information is usually bulky and difficult to manipulate. Data relevant to a task may be scattered across different pages in many volumes so that much of an analyst's time may be taken up by such operations as gathering statistics from different places, reorganizing information into tables to test hypotheses, or simply scanning through data to find particular items of information.

A plausible solution to this problem is to put all information on computer mass storage, where it could be revised and manipulated easily and where it could be directly accessible to users at remote terminals. The problem with this scheme, though, is that the majority of users will not be skilled enough programmers to take full advantage of data if it is simply put into the form of formatted computer files. There would have to be additional software to let a user deal with stored information in a flexible way without having to resort to a standard programming language like COBOL or FORTRAN.

This leads logically to the idea of systems like REL, which could accept directives in a natural language for carrying out general sorts of analysis on collections of stored data. The claim for these systems is that they would be conversational, requiring no formal training for effective use. They would supposedly contain preprogrammed statistical, organizational, and search functions that could be called by a user without detailed knowledge of either the implementation of a data base or the operation of the computer providing access to it. The goal of the present report is to examine how well REL as a specific example of such systems would actually work out in the environment of science and technology intelligence analysts.

1.3. THE REL SYSTEM

The REL ("Rapidly Extensible Language") System is a data analysis facility with various base languages from which a user through trial and error could build up an access language appropriate to a particular problem area. This could in practice be almost anything; it need not be a natural language in the usual sense. For the purposes of the present report, however, discussion will center on a version of REL with the base language REL English, together providing the nucleus for natural language programming of all kinds.

The REL system was developed by Professor F.B. Thompson of the California Institute of Technology as a logical culmination of proven technology in language processing, semantic modeling, automatic programming, data base management, and information storage and retrieval. It is a conservative system from a theoretical standpoint, taking what is already known to work efficiently instead of trying to break new ground. On the whole, there appear to be no serious obstacles in principle to its successful implementation; and in fact, several versions of it are currently running, including one at RAND Corporation accessible through the ARPA Net.

From the user's point of view, REL in its natural language configuration would consist of four main parts:

- A storage management subsystem providing memory paging and other services.

- A language processor consisting of a parser and a syntax-directed interpreter.

- A REL English package with a grammar of fundamental English constructions, a definition of simple relational data structures, and arithmetic and statistical routines.

- A user language package defining a vocabulary of terms for entities, concepts, and relations in some domain of interest.

The REL English package plus the language processor establishes the basic framework for data analysis. A sentence entered as input to the system is parsed according to the syntactic rules of REL English; the results are then interpreted in terms of set-theoretic and arithmetic operations yielding classes of entities or numerical functions over such classes as output. Examples of possible input sentences would include:

1. What Soviet interceptors have the XX-YY-ZZ radar?

2. How many fighter-bombers have a military power speed greater than Mach 2?

3. What is the correlation between the wing span of Soviet fighters and their ferry range?

4. What are the operational ceilings and the combat radiuses of Soviet fighters that carry the AAA-NNN missile and that have an overall length of less than LLL meters?

The first sentence would be treated as a directive to examine all of the entities listed in the class "interceptors" and to return a list of those associated with "XX-YY-ZZ" through the relation "radar." The second sentence is treated similarly except that the length of a list instead of the list itself is returned. The third sentence illustrates a function over more than one class. The fourth shows how conditions can be conjoined and how more than one result can be specified.

The mechanics of how all of this comes about will not be discussed here. For a more thorough description of the REL system and the syntax of REL English, the reader should consult the available documentation for REL [see Thompson and Dostert, "Practical Natural Language Processing: The REL System as Prototype," in Advances in Computers 13, Yovits and Rubinoff (eds.), Academic Press: New York, 1975; and Thompson, "REL User's Reference Guide," California Institute of Technology, 1977]. The present report will aim more at the problem of what the underlying assumptions of REL imply for its application to intelligence data analysis. Subsequent sections of this report will deal specifically with the topics of REL as a relational data base system, REL English as a natural language for relational data analysis, strategies in REL for computing with large data files, semantic problems with intelligence data in REL, and the usefulness to intelligence analysts of the sorts of extensibility provided by REL.

SECTION 2

REL AS A RELATIONAL DATA BASE SYSTEM

2.1. RELATION MODELS OF DATA

In a strict sense, it is somewhat misleading to talk about relational data bases as a distinct kind of entity. The concept of relations is really nothing more than a way of looking at stored data; and in the end, any organizational scheme could be viewed relationally, whether it is sequential, hierarchical, network, or whatever. The main advantage of the relational viewpoint is that it lets a user work with a data base at an abstract level, dispensing with most of the details of its implementation. This is similar in idea to programming in a high-level language like PL/1 or FORTRAN instead of an assembly language specific to a particular processor.

The cost of abstraction in a relational data base is the additional overhead required to map references to relational entities into references to the actual kinds of data structures called for in a given application. This overhead can often be quite large because of the difficulty of optimizing the mapping of relational references in all possible cases; but if a data base is intended for interactive access as opposed to access by applications programs, then the relative inefficiency of a relational approach is outweighed by its greater convenience for the user. This is especially true for the science and technology intelligence data analysis problem, which typically involves an unsophisticated computer user sifting through bodies of data in unpredictable ways.

In order to be more precise about relational data bases, it is helpful to define a few terms here semiformally:

- Atomic values are indivisible data items, including fixed- and floating-point numbers of a given radix and precision, character strings of a given length, and Boolean values.

- Relations are associations established between a given number of atomic values of certain types, not necessarily distinct; for example, a 3-way association between a 24-character string and two five-place decimal fixed-point numbers.

- An interpretation for a relation is a labeling of the relation and each of its components so as to specify its significance; for example, the 3-way association above could be interpreted as a "geographic location" relation with its components standing for "military base name," "longitude," and "latitude."

This departs somewhat from the formal system of Codd ["A Relational Model of Data for Large Shared Data Banks," Communications of the ACM 13:6 (1970)] by incorporating a notion of data semantics in the last definition. Otherwise, there is no major difference.

Relational interpretations will provide a means of connecting up a relational data base with an interactive user interface through establishment of a consistent scheme of reference to data. This will become especially important in the case of a natural language front-end, because of the frequency of paraphrases corresponding to simple permutations of labeling in an interpretation. The mechanism of such paraphrases and permutations will be described more fully in the section on natural language.

2.2. RELATIONAL DATA ANALYSIS

Superficially, a relational association can be thought of as a kind of formatted data record; and a collection of such associations, as a kind of file. The analogy extends further in that it is convenient to designate certain components of a relation as being "keys," in the same way that certain fields in records are. The relational approach departs from the conventional record-file approach, though, by encouraging the definition of new relations as a standard analytical technique. This use of new relations is typically so unrestricted that it becomes impractical to tie them to actual physical allocations of records.

The ability to define new relations gives a user the power to make associations over large bodies of data. Relational systems recognize the importance of this kind of data analysis by usually providing a wealth of high-level operators for the formation of new relations from old ones; Codd, for example, lists the following:

- Projection - The deletion of one or more components from a relation; for example, the "geographic position" relation described above could be mapped by projection into a "geographic latitude" relation by deleting the "longitude" component.

- Join - Combining two relations according to common atomic values in given components of each relation; this could be used to define a relation expressing the complement of military bases by joining the association of equipment with bases and the association of bases with military units.

- Composition - The formation of a relational association of atomic values via an indirect association through an intermediate atomic value; this would allow an association between radars and geographic locations to be obtained from an association between radars and military bases and an association between bases and longitudes and latitudes.

- Restriction - The selection of a set of associations from a relation according to keys in another relation; for example, using the association of radars with bases and the association of bases with longitudes and latitudes to obtain a new relation linking bases with radars to their geographic locations.

In a full relational system, these fundamental operations would probably be augmented with set-theoretic operations on relations of the same component types and with conditional selection operations based on associated atomic values. These together would constitute an extensive nonnumerical data analysis facility with great potential for intelligence applications if made available along with arithmetic and statistical functions in a convenient package for unsophisticated computer users. Although this would not replace an intelligence analyst in the overall task of recognizing significant patterns in data, it would certainly help the analyst out by taking over many of the routine data manipulations implicit in dealing with any large data base.

At present, the difficulty of implementing relational data analysis systems efficiently has been an obstacle to their development on a big scale. Nevertheless, it is possible to come up with quite workable systems by carefully tailoring them to particular applications. The REL system is a case in point here; it sacrifices a certain amount of generality in its treatment of relations in return for simplification of access to stored data. The results of this compromise are mixed, but still manage to show the usefulness on the whole of a high-level approach to data analysis.

2.3. REL ENGLISH DATA ANALYSIS

The major restriction on REL as a relational data analysis system is that all data must be expressed in terms of binary relations: those associating two atomic values at a time. Mathematically speaking, the restriction is not really serious in that relations of any degree can be constructed from binary relations; but with the current facilities of REL English, this will in practice be awkward for the typical user. To allow full use of relations of all degrees, REL English would have to provide the equivalent of the projection, join, composition, and restriction operators, which are not now available.

As it stands, however, REL English does provide for some rather important kinds of data analysis. These involve four basic types of data structures:

- named individual - standing for specific entities like persons ("Boris Gudonov"), places ("Riga"), or things ("Air Force One").
- class - an explicit list of individuals ("Noguchi," "Otzu," "Kurosawa")
- relation - a labeled list of associations between pairs of individuals (capital: "Nouackchott" - "Mauritania," "Cairo" - "Egypt," "Madrid" - "Spain")
- number relation - a labeled list of associations between individuals and numerical values (population: "Krasnoyarsk" - 501,000, "Pskov" - 105,000, "Novgorod" - 128,000)

From a logical viewpoint, these data structures define an extensional as opposed to an intensional system; the distinction here will be important in connection with the semantics of REL English discussed in Section 3.

The motivation behind an extensional approach is to force all stored data to be facts about specific individuals. General facts are prohibited because these normally require some mechanism of inference before they can be fully exploited. By taking an extensional approach, REL English greatly simplifies its semantics and avoids the formidable problem of implementing an efficient proof procedure. Some flexibility is lost, but this conforms on the whole with the conservative policy underlying the design of REL.

The principal nonnumerical form of data analysis supported by REL English is the creation of classes of individuals according to given relational criteria. Once a class is established, it becomes possible for a user to carry out numerical computations such as evaluating arithmetic expressions containing values associated with individuals in the class or calculating various statistical functions over these values for the entire class. The search for a particular individual in a data base generally reduces to obtaining the intersection of a number of classes.

Classes can be established in several ways. The most direct method is to declare a new class with a specified name and to assert the membership of individuals in the class. This corresponds to typing something like the following in REL English:

```
NATO NATION:=CLASS
BELGIUM IS A NATO NATION.
NORWAY IS A NATO NATION.
        .
        .
        .
```

This results in the long-term allocation of space in a data base to store the list of actual members in a class. In many cases, however, a user will be interested in a class only as an intermediate result, and so it will be more convenient to allocate only temporary space for it. Such classes are established by description in terms of other classes and have their membership computed dynamically for each reference to them. For example, the REL English noun phrase "NATO nations that control nuclear weapons" establishes a temporary class expressed as a restriction of the known class "NATO Nations".

An extremely useful feature of REL English allows the user to assign a name to a class established by description; this is done as in the following:

```
DEF:  WESTERN NUCLEAR POWER:  NATO NATION THAT CONTROLS NUCLEAR WEAPONS
```

The "DEF" facility merely serves to set up "macro-language" substitutions for expressions in input sentences; its effect, however, is to provide a rudimentary way for users to deal with concepts as well as to save them some typing at a terminal. A "concept", to be sure, is an intensional entity with all the attendant hazards; but its use in REL English is strictly limited and kept separate as much as possible from the use of explicit classes in order to avoid trouble. The major hazard for the user is in the case queries of the form

IS XXX A YYY?

The response for this may differ according to whether "YYY" is an explicit class or a conceptual class [for details, see Thompson and Dostert, op. cit., Section 5.1.].

In a sense, the main purpose of the binary relations of REL English is really to provide for the definition of conceptual classes by allowing properties to be attached to individuals. Binary relations by themselves have limited value for the kind of nonnumerical associational data analysis typically connected with relational data base systems; but within the overall scheme of conceptual classes in REL English, they can in effect be composed into associations between widely dispersed items of data by incrementally generating the key components of these associations as conceptual classes. The technique is not easy since it requires skill to break a problem up into the right parts and to define the right classes [see Thompson, op. cit., Section I-D], but it is still useful in that it can be carried out at a high level of data abstraction.

The underlying operation of the REL system is organized for maximum efficiency in dealing with conceptual classes. The emphasis is on reduction of references to pages of data on secondary storage, the slowest but also the most critical part of a large data base system. Because the membership of a conceptual class generally has to be computed at each reference from data scattered over pages that cannot all fit into main memory at the same time, it becomes important to avoid excessive rereading of data pages due to overlaying. This sort of optimization is accomplished in REL in part by manipulating conceptual classes in terms of descriptions instead of explicit lists where possible [see Greenfeld, "Computer System Support for Data Analysis," REL Project Report #4, California Institute of Technology, 1972]. It should perhaps be noted that this example is where extensional operations on data as explicit lists are actually less efficient than intensional operations on data through descriptions, since the latter approach allows a savings in I/O time.

SECTION 3

NATURAL LANGUAGES IN REL

3.1. NATURALNESS AND EXTENSIBILITY

The notion of a natural language in REL is unorthodox in that it is meant to apply to any language that has been shaped and refined over an extended time to meet the needs of a community of users. This approach allows REL to sidestep a great deal of difficulty by making the user assume the responsibility for defining what a workable natural language system should consist of. The basic syntax and semantics of a language are supplied in REL English; but the actual substance of it has to be built up through the REL extension facilities.

Whether or not this kind of naturalness will succeed for an area like science and technology intelligence data analysis remains to be seen. So far, there have been only sketchy, anecdotal accounts of how REL has been used for some relatively simple problems [see Dostert, "REL - An Information System for a Dynamic Environment," REL Report No. 3, California Institute of Technology, 1971]. The true test of the system will be whether its users will put up with the day-to-day operation of the system long after its novelty has worn off.

In the absence of hard facts from an actual FTD installation, the evaluation here of REL as a natural language system will have to rest on two theoretical points: the adequacy of REL English as core subset of full

English and the usefulness of the REL extension facilities for a person who is not an expert on linguistics or systems programming. The discussion will mostly center around specific problems that might arise for an intelligence analyst working with a data base of facts about Soviet aircraft. Since the actual Soviet Aircraft Handbook is classified, the examples here will be concerned with a dummy Soviet bomber data base generated by Battelle Corporation as a preliminary test of REL and with a hypothetic data base derived from the annual Soviet Aerospace Almanac of Air Force magazine.

3.2. A REL SYSTEM FOR SOVIET AIRCRAFT

Although much could be said about REL English purely on formal grounds, it is more revealing simply to see how it handles the kinds of queries an analyst is likely to direct at a science and technology data base. There are as yet no transcripts of actual REL working sessions to draw examples from, but it is still possible to gain some insight into the effectiveness of REL English for intelligence analysis by looking at a test data base and some queries put together by Battelle Corporation. The data base was intended as a simple, unclassified version of the Soviet aircraft data base planned for the FTD REL prototype system; it is reproduced for reference in Appendix A along with the Battelle queries.

The test data base looks straightforward, but that is accidental. The prohibition on general facts in REL forces aircraft types to be defined as generic individuals. This turns out to be all right because of the types of facts in the data base as it stands, but it can be troublesome in a more general case. For example, there could also be facts in a data base about specific individuals ("Air Force One has a XX-YY radar") or about classes ("There have been 10,000 Mig-21's produced since 1970") that would be extremely awkward to express in the framework of generic individuals alone. The problem here is not a weakness in the relational data representation, but rather the constraint of consistency arising from a REL English user interface.

An analysis of the Battelle queries raises still some more issues of interest. To get an idea of how REL English would handle these queries in the absence of a working REL system, a version of REL English was brought up by putting a translation of its syntactic rules into the PARLEZ natural language system development facility at PAR Corporation. This provided a REL English parser, which could be used to determine the syntactic acceptability of user queries. The semantics of a query would not be evaluated in this way, but this is easy enough for a person to do on a test data base the size of the Battelle one.

The experiment on the whole showed at least the syntactic adequacy of REL English for the Battelle queries, once the necessary definitions were entered. The only syntactic problem of any importance was the omission of the preposition "for" in the list of REL English function words as of February, 1976, preventing a successful parsing of queries 5 and 11. The immediate remedy for this would be to define "for" equivalent to "of", but this kind of language extension is beyond the scope of the facilities generally accessible to a nonsophisticated user.

All of this, however, turns out to be a relatively minor point; a more serious problem is in the semantics of queries, especially with respect to nomenclature and reference. The REL system requires that a user know the precise names of individuals, classes, and relations in a data base; variants of names can always be defined, but even these ultimately have to be linked explicitly back to the official names. It is likely, therefore, that a query at first try will be rejected on semantic grounds because of a loose use of names. For example, in Query 6, the expression "missions" would not be recognized because the target relation is actually called "aircraft mission" in the data base.

REL complicates the situation by providing uninformative diagnostics for the user in case of a failure in semantic interpretation, typically "Eh?". The diagnostic would be the same in the case of a syntactic failure in parsing, such as when "for" is undefined. The user therefore would have the chore of finding out what went wrong. The best strategy would perhaps be to print out all the correct names for relations in a data base before submitting a query; but this can be time-consuming and still not foolproof since the person who generated Query 6 incorrectly probably had a complete tabulation of the aircraft data base at hand. The observation here is that, since a person seems naturally prone to dropping off components of long noun phrases, it may be appropriate for a natural language system to address this problem directly instead of forcing the user to handle it piecemeal through definitions for every situation that arises. For example, a best match facility for names would have been handy for Query 6.

The problem of nomenclature, however, goes further than simply finding best matches. A more difficult matter is illustrated in Query 2 with the expression "swing wing bomber" -- that is, "bomber whose aircraft type is SGW." REL English would handle the latter form easily enough, but not the former one. The trouble is that an atomic value is used as a qualifier for a class without specifying the intermediate relation. The difference between "SWX" and "swing wing" prevents the REL system from inferring the relation. Worse yet, it is hardly easier for a user to discover the missing relation by scanning a list of those defined in a system since the relation name sought is the rather obscure "aircraft type".

Queries 3, 4 and 5 bring up another problem of nomenclature. In the test data base, there is a relation called "maximum fuselage length," but there is also a built-in REL English function called "maximum". It is unclear from the available documentation how REL would handle this problem of ambiguity, although it would be desirable for it to allow the free use of terms normally treated as function names. In fact, it might even be good to be able to compute function values by name if they are often needed but require going through a great deal of computation.

Query 7 illustrates still another aspect of the nomenclature problem. The expression "LF-31" is being used to identify the Bongo bomber, but REL English would probably fail to recognize such usage since "LF-31" is defined as an atomic value (a data table entry), not as a name of an individual (a column label for the data table). One might try to get around this difficulty by allowing reference to individuals by certain "key" relations, but in this case, the relation "aircraft designation" cannot be a key since two aircraft types actually have the same designation.

Finally, Query 13 shows where the nomenclature problem crosses over into the domain of syntax. The noun phrase "the PBX-24 UHF system" should actually have been expressed as "the PBX-24 as UHF system" for best results with REL. REL English would probably handle the first form correctly, but this is likely to be inefficient. The noun phrase will probably be evaluated as "PBX-24" by the REL English individual-relation rule for merging noun phrases [see Thompson and Dostert, op. cit., Section 5.2.]; the semantic interpreter would then probably have to reconstruct the "UHF" relation by looking for associations between "aircraft" and "PBX-24". The entire process could have been short-circuited if a system could somehow preserve the information provided within the original expression.

On the whole, REL English would seem rather mediocre in handling the Battelle queries, probably failing for about half of them; the exact level of performance here will have to be determined in actual test runs with REL. For example, in Query 10, it is impossible to tell solely on the basis of documentation whether the REL reference facility is up to making the connection between "the Bongo bomber" and "condition one". The conclusion from all of this is that REL English seems fairly strong on syntax, but perhaps inconveniently primitive in its semantics at present. There is still much room for improvement.

3.3. REL EXTENSION FACILITIES

Because of the sketchy nature of base languages like REL English, the REL system must rely heavily on its extension facilities in order to meet the language needs of a community of users. Two levels of extensibility are possible: one can take advantage of definitional features built into a base language for relatively superficial additions, or one can go into the REL metalanguage to change the underlying structure of a user language. The latter alternative generally implies a major reprogramming effort, and so is probably beyond the ability of most users. The descriptions of language extension in REL documentation in fact concentrates almost entirely on the kind of extensibility possible within a base language. The discussion here will take the same approach.

REL English allows definitions of individuals, classes, relations and their converses, verbs and their objects, and general transformations. The last seems by far to be the most versatile in that it can serve to define abbreviations, paraphrases, and conceptual classes. It provides a way of getting around many of the restrictions of the extensional semantics of REL English. For example, Query 7 of the Battelle test set uses "dimensions" to refer to a class of properties, which is normally forbidden in an extensional semantic scheme; it is possible, though, to define "dimensions" with a paraphrase as follows:

DEF:DIMENSIONS:OVERALL FUSELAGE LENGTH AND MAXIMUM FUSELAGE WIDTH
AND MAXIMUM FUSELAGE HEIGHT AND WING SPAN

This would allow Query 7 to be handled properly by REL without compromising the semantic scheme.

The use of definitions such as this makes REL English into a rather powerful user language. It does, however, exact a penalty in terms of added complexity in sentence analysis, requiring the implementation of a parser capable of handling general rewrite grammars. There may be significant slowdowns in query processing if many definitions have to be expanded. One of the questions remaining to be answered by experimentation is in fact whether REL is able to support the numbers of definitions required in a science and technology intelligence data analysis system without degrading overall response times unacceptably.

There is also the question of how far one can sufficiently extend REL English for intelligence applications by base language definitions alone, since these leave the underlying semantics of REL English alone. The sorts of data in the Soviet Aircraft handbook, for example, is most naturally expressed in terms of _n_-component relations, and this structure will tend to be reflected in the kinds of queries one would probably make of it.

> What is the combat radius of the MIG-21F on mission XX with external wing tanks?

What is the maximum military power speed of the MIG-25 at 15,000 feet?

REL English cannot yet handle such queries in a convenient way because of its minimal support at the user level for associations of more than two atomic values at a time. Other semantic weaknesses uncorrectable by base language definitions would include inability to store negative information ("The SU-15 carries no guns"), no direct means for associating degrees of uncertainty with relational assertions ("The XYZ has a dry weight of YYYYY ± ZZZZ"), no efficient way of handling exceptions ("All XYZ have an ABC, except for XYZ-1") or variations ("The SSS-2 is a reconnaissance version of the SSS-1"), and no provision for facts about classes ("There are 200 XYZ's based in DDD") as well as about individuals.

The REL English definitional facilities also cannot alter the underlying syntax of the language. No use of definitions, for example, would allow the queries

WHAT IS THE WING SPAN OF THE BINGO?

WHAT IS THE EMPTY WEIGHT OF THE BONGO?

to be entered simply as

WING SPAN OF THE BINGO?

EMPTY WEIGHT OF THE BONGO?

so as to reduce the amount of typing done by a user. Abbreviations can be established for terms and concepts interpretable as individuals, classes, and relations; but syntactic structure in REL English such as "what is" can only be changed by actually going into the metalinguistic description of the language. This plus the basic semantics of REL English thus places clear practical limits on extensibility.

SECTION 4

CONCLUSIONS

4.1. OUTLOOK FOR REL

REL is by no means perfect as an interactive system for natural language data analysis. Yet for all of its syntactic and semantic shortcomings, it does seem to provide some useful computer services fairly efficiently to persons who want to work with a large data base without having to be concerned with the details of access programs and data structures. Furthermore, it is the only system of its type up to now that has shown any real promise of being immediately applicable to problems in the real world.

The REL natural language approach has certain advantages for an area like science and technology intelligence analysis, which generally involves unpredictable modes of access to large amounts of data in diverse forms. A modest natural language capability like that offered by REL should let an intelligence analyst range freely over such data in ways that roughly correspond to the analyst's own lines of thought rather than to the conventions of a particular data base implementation. It would not replace the analyst, but it should enhance the process of analysis in places where a person would typically become bogged down by large masses of information.

The only really bad mark against REL has been the persistent inability to get a new prototype system running for FTD after many extensions of deadlines. The problem seems to be managerial, however, rather than technical. The basic ideas of REL seem to be altogether realizable within the current state of the art in language processing and data base management; the need at present is to develop these ideas further. Given the existence of various working versions of REL, it may be appropriate now to try to get potential users to become more active in determining the course of development for a more mature REL system, since this is really the only way of insuring that it is truly natural.

4.2. RECOMMENDATIONS

Continued work on a REL system for intelligence data analysis is worth pursuing because of the existence of relatively advanced versions of REL accessible through the ARPA net. The main emphasis, however, should be not on supporting further development of REL, but rather on identifying how it best fits into the FTD environment. It will be important to expose analysts to the facilities now available in REL so as to get their comments on the present usefulness of the system and, if possible, their suggestions on how the system might be changed to meet their needs better. This should be carried out as a long-term study in order to get as close as possible to the use of REL under actual operational conditions at FTD.

This sort of experimentation should establish the basis for specific recommendations on improving REL for FTD applications. It is expected that these will include the following:

- The reliability and transportability of REL would be helped out if the system were programmed in a high-level language like PL/I. The current choice of IBM 360/370 assembly language in fact seems to have contributed significantly to the difficulty in bringing up an FTD prototype system. Since REL at present is still undergoing major change, any gain of efficiency from assembly language programming is irrelevant.

- The language specification facility in REL should be cleaned up to make it easier to modify the syntax of a base language; the current descriptions of base languages seem unnecessarily cryptic. Definitions made in REL English could also be simplified; there is no reason, for example, why a user should type

  FIGHTER := CLASS

  MIG-21 := NAME

  THE MIG-21 IS A FIGHTER.

  The last line is sufficient to imply the first two lines.

- The grammar of REL English should not always require input to be in the form of complete English sentences. In fact, language is much more natural when it takes advantage of context to shorten utterances. For example, consider the following series of queries:

```
WHAT IS THE EMPTY WEIGHT OF THE MIG-21?
WHAT IS THAT OF THE MIG-23?
WHAT IS THAT OF THE MIG-25?
```

Everything in the box is extra typing for the user that ends up contributing nothing.

- REL English diagnostics should be greatly expanded. The bottom-up parsing algorithm of REL makes it more difficult to have good diagnostics, but even so, there are some obvious things that could be done; for example, a message identifying an unknown word that prevents the successful parsing of a sentence. Another possibility is to recognize common incorrect forms within REL English itself and to assign error messages to them as their semantic interpretations.

- REL should also be better documented for the user. The examples provided in the recent reference guide are suggestive, but missing is a list of things that do <u>not</u> work so that the user can avoid wasting time in trying them out. It would also be helpful to have on-line documentation of some type for a user at a terminal.

- REL English should incorporate more aspects of intensional semantics where the cost would be low in terms of system overhead. For example, it would seem reasonable to allow special relations having a class as one component or to implement simple taxonomical hierarchies of concepts where more specific concepts would inherit properties from more general ones. Both of these are common features in present natural language systems and would be convenient for the classification of aircraft in the FTD environment.

- The use of classes intensionally defined by description should be allowed in assertions as well as in questions; if it is already, then it should be clearly documented. This kind of linguistic usage is basic to the semantics of restrictive relative clauses and is implemented to some extent in the special syntax of the REL English definition facility. A more general treatment would make a user language more natural with respect to implicit assertions while still staying clear of a complete inferential scheme like resolution. In fact, it might be semantically advantageous to have all classes of individuals be intensional and to make present REL English extensional classes into unary relations.

- REL English should define n-ary relations at least virtually for the user. This could be done by providing relational operators that would automatically construct n-ary relations from the fundamental binary relations of REL English. With such operators, an analyst would be able to have the ability to make associations as in a classical relational data system of the Codd type. It should be noted that the time component of relational information in REL

English already marks a departure from strict binary relations. This kind of auxiliary information, though, may have to be extended considerably for the FTD environment [c.f. Fact Control Information in FTD STIS].

- o REL English should know more about naming conventions and common permutations of names to free the user from having to make as many definitions as required now; the present approach tends to force a user to think in terms of the full official names for binary relations. It would also be useful to be able to get at components of names in some way so as to eliminate the need for a multiplicity of relations like "AIRCRAFT MISSION CODE NAME," "AIRCRAFT CODE NAME," "AIRCRAFT MISSION," and "AIRCRAFT DESIGNATION" as well as the actual name given to an individual.

- o The REL system should allow shared access to a single copy of the data base. The system now requires a user to have a personal copy of a data base if any extensions are to be made on it. This would seem to be wasteful in space since REL English more or less must be extended before it can really be useful; multiple copies would also complicate the problem of updating information. REL currently falls short of being a true data base management system in that it does not allow multiple users to have different submodels for viewing a common body of data. Problems of protection, security, and control of information need to be addressed here.

- REL should provide for textual information such as the note for the LF-31 in the Battelle test data base. This sort of information could be represented relationally, but it is simpler and more direct to save it as it is. The form of the Soviet Aircraft Handbook itself suggests the stored text approach; there is typically a textual description for each aircraft containing data that is inconvenient to put into table form. The STIS data base illustrates another possibility, the capability of attaching text comments as qualifiers for individual items of information.

Whatever the shortcomings of the present REL system, however, it will be important to fix upon a workable version of it and to make it available for experimental use in a variety of situations. The practical rather than the theoretical issues ought to be stressed here because the goal of REL after all is not so much to make technical progress as to put together proven methods from diverse areas of computer science into a reasonably powerful system for data analysis. There is a need for users at all levels of computer sophistication to gain experience with the system so as to provide more feedback in its development, particularly on the psychological relationship between system and user. The current REL system has many good ideas, but it is ultimately the user who will decide whether or not it really helps to solve any practical problems.

APPENDIX A

TEST FILE RELATIONS — TEST FILE INDIVIDUALS

		"BINGO"	"BANGO"	"BONGO"	"BANGO-F"
1.	Aircraft Mission Code Name	Bingo Bomber	Bango Bomber	Bongo Bomber	Bango Ferry
2.	Aircraft Designation	TH-1	TH-2	LF-31	TH-2
3.	Aircraft Mission	Bomber	Bomber	Bomber	Ferry
4.	Aircraft Type	FXW	FXW	SGW	FXW
5.	Country	UR	UR	UR	UR
6.	Fuselage Length	97	78.5	84.25	78.5
7.	Overall Fuselage Length	104.6	86	84.25	86
8.	Maximum Fuselage Width	5.6	-	5.4	-
9.	Maximum Fuselage Height	7.1	-	6.9	-
10.	Ground Clearance	3	3.4	-	3.4
11.	Height of Tail above Ground	21	19.6	17.8	19.6
12.	Landing Gear Track	23	19.4	-	19.4
13.	Tandem Gear to Outrigger Gear Distance	-	-	40	-
14.	Wheel Base	-	-	7	-
15.	Wing Span	104.75	96	110.5	96
16.	Wing Area	1048.4	875	1206	875
17.	Wing Aspect Ratio	6.4	5.3	6.2	5.3
18.	Wing Dihedral	2.0	-	2.1	-
19.	Wing Cathedral	-	1.8	-	1.8
20.	Sweepback Leading Edge	48	-	-	-
21.	Sweepback Trailing Edge	40	-	-	-
22.	Sweepback Leading Edge Inboard	-	40	-	40
23.	Sweepback Leading Edge Outboard	-	35	-	35
24.	Sweepback Trailing Edge Inboard	-	22	-	22
25.	Sweepback Trailing Edge Outboard	-	27	-	27
26.	AMPR Weight	65000	80000	74000	80000
27.	Empty Weight	81000	95000	77000	95000
28.	Operational Weight Empty	101000	106000	89000	106000
29.	Maximum Takeoff Weight	158000	160000	147000	168000
30.	Normal Takeoff Weight	134000	121000	107000	128000
31.	UHF	-	PBX-24	Type Unknown	PBX-24
32.	VHF	Type Unknown	-	BEL SYS/4	-
33.	Intercom	TELE-48A	-	-	-
34.	Conditional Normal Takeoff Weight	-	-	107000 C1	-
35.	Condition One	-	-	With Auxiliary Tank	-
36.	Aircraft Code Name	Bingo	Bango	Bongo	Bango
37.	Notes	-	-	(See Attachment)	-

NOTE FOR LF-31

This aircraft normally carries a crew of four. The Bongo was first seen on June 19, 1975, and has yet to get off the ground. It is equipped for aerial refueling and the fuel transfer rate is approximately 400 gallons per minute. The Bongo is a swing wing, twin turbofan short-range weapon system equipped with 4 air-to-air heat-seeking missiles. Its design is similar to the Bungo, the LF-30, differing only in overall length due to the addition of the refueling mechanism.

BATTELLE QUERIES

1. What are the aircraft code names of Soviet bombers?
2. Which Soviet bombers are swing wing bombers?
3. What is the overall fuselage length of the Bingo?
4. What is the maximum fuselage width of the Bango?
5. What is the average landing gear track for Soviet aircraft?
6. What are the missions of Soviet aircraft?
7. What are the dimensions of the LF-31?
8. Does the Bingo have a UHF system?
9. Which Soviet bombers have VHF systems?
10. What is the conditional normal takeoff weight of the Bongo Bomber and what is condition one?
11. What are the notes for the Bongo?
12. Which Soviet bomber has the largest normal takeoff weight?
13. Which Soviet aircraft carries the PBX-24 UHF system?
14. What are the aircraft charactersitics of the TH-1?
15. What is the average fuselage length of Soviet Bombers?

BIBLIOGRAPHY

Codd, E.F.; "A Relational Model of Data for Large Shared Data Banks," Comm. ACM 13:6, June 1970.

Date, C.J.; An Introduction to Database Systems. Addison-Wesley: Reading, Massachusetts, 1975.

Dostert, B.H.; "REL - An Information System for a Dynamic Environment," REL Report No. 3, California Institute of Technology, 1971.

Greenfeld, N.R.; "Computer System Support for Data Analysis," REL Project No. 4, California Institute of Technology, 1972.

Thompson, B.H.; "REL User's Reference Guide," California Institute of Technology, 1977.

Thompson, F.B. and Dostert, B.H.; "Practical Natural Language Processing - The REL System as Prototype," In Advances in Computers 13, Yovits and Rubinoff (eds.), Academic Press, New York, 1975.

MISSION
of
Rome Air Development Center

RADC plans and conducts research, exploratory and advanced development programs in command, control, and communications (C^3) activities, and in the C^3 areas of information sciences and intelligence. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

AMERICAN REVOLUTION BICENTENNIAL 1776-1976