TRANSPORTABLE NATURAL-LANGUAGE INTERFACES TO DATABASES

by
Gary G. Hendrix and William H. Lewis
SRI International
333 Ravenswood Avenue
Menlo Park, California 94025

I INTRODUCTION

Over the last few years a number of application systems have been constructed that allow users to access databases by posing questions in natural languages, such as English. When used in the restricted domains for which they have been especially designed, these systems have achieved reasonably high levels of performance. Such systems as LADDER [2], PLANES [10], ROBOT [1], and REL [9] require the encoding of knowledge about the domain of application in such constructs as database schemata, lexicons, pragmatic grammars, and the like. The creation of these data structures typically requires considerable effort on the part of a computer professional who has had special training in computational linguistics and the use of databases. Thus, the utility of these systems is severely limited by the high cost involved in developing an interface to any particular database.

This paper describes initial work on a methodology for creating natural-language processing capabilities for new domains without the need for intervention by specially trained experts. Our approach is to acquire logical schemata and lexical information through simple interactive dialogues with someone who is familiar with the form and content of the database, but unfamiliar with the technology of natural-language interfaces. To test our approach in an actual computer environment, we have developed a prototype system called TED (Transportable English Datamanager). As a result of our experience with TED, the NL group at SRI is now undertaking the development of a much more ambitious system based on the same philosophy [4].

II RESEARCH PROBLEMS

Given the demonstrated feasibility of language-access systems, such as LADDER, major research issues to be dealt with in achieving transportable database interfaces include the following:

* Information used by transportable systems must be cleanly divided into database-independent and database-dependent portions.
* Knowledge representations must be established for the database-dependent part in such a way that their form is fixed and applicable to all databases and their content readily acquirable.
* Mechanisms must be developed to enable the system to acquire information about a particular application from nonlinguists.

III THE TED PROTOTYPE

We have developed our prototype system (TED) to explore one possible approach to these problems. In essence, TED is a LADDER-like natural-language processing system for accessing databases, combined with an "automated interface expert" that interviews users to learn the language and logical structure associated with a particular database and that automatically tailors the system for use with the particular application. TED allows users to create, populate, and edit their own new local databases, to describe existing local databases, or even to describe and subsequently access heterogeneous (as in [5]) distributed databases.

Most of TED is based on and built from components of LADDER. In particular, TED uses the LIFER parser and its associated support packages [3], the SODA data access planner [5], and the FAM file access manager [6]. All of these support packages are independent of the particular database used. In LADDER, the data structures used by these components were hand-generated for a particular database by computer scientists. In TED, however, they are created by TED's automated interface expert.

Like LADDER, TED uses a pragmatic grammar; but TED's pragmatic grammar does not make any assumptions about the particular database being accessed. It assumes only that interactions with the system will concern data access or update, and that information regarding the particular database will be encoded in data structures of a prescribed form, which are created by the automated interface expert.

The executive level of TED accepts three kinds of input: questions stated in English about the data in files that have been previously described to the system; questions posed in the SODA query language; single-word commands that initiate dialogues with the automated interface expert.

IV THE AUTOMATED INTERFACE EXPERT

A. Philosophy

TED's mechanism for acquiring information about a particular database application is to conduct interviews with users. For such interviews to be successful,

The work reported herein was supported by the Advanced Research Projects Agency of the Department of Defense under contracts N00039-79-C-0118 and N00039-80-C-0645 with the Naval Electronic Systems Command. The views and conclusions contained in this document are those of the authors and should not be interpreted as representative of the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency of the U.S. Government.

* There must be a range of readily understood questions that elicit all the information needed about a new database.
* The questions must be both brief and easy to understand.
* The system must appear coherent, eliciting required information in an order comfortable to the user.
* The system must provide substantial assistance, when needed, to enable a user to understand the kinds of responses that are expected.

All these points cannot be covered herein, but the sample transcript shown at the end of this paper, in conjunction with the following discussion, suggests the manner of our approach.

B. Strategy

A key strategy of TED is to first acquire information about the structure of files. Because the semantics of files is relatively well understood, the system thereby lays the foundation for subsequently acquiring information about the linguistic constructions likely to be used in questions about the data contained in the file.

One of the single-word commands accepted by the TED executive system is the command NEW, which initiates a dialogue prompting the user to supply information about the structure of a new data file. The NEW dialogue allows the user to think of the file as a table of information and asks relatively simple questions about each of the fields (columns) in the file (table).

For example, TED asks for the heading names of the columns, for possible synonyms for the heading names, and for information about the types of values (numeric, Boolean, or symbolic) that each column can contain. The heading names generally act like relational nouns, while the information about the type of values in each column provides a clue to the column's semantics. The heading name of a symbolic column tends to be the generic name for the class of objects referred to by the values of that column. Heading names for Boolean columns tend to be the names of properties that database objects can possess. If a column contains numbers, this suggests that there may be some scale with associated adjectives of degree. To allow the system to answer questions requiring the integration of information from multiple files, the user is also asked about the interconnections between the file currently being defined and other files described previously.

C. Examples from a Transcript

In the sample transcript at the end of this paper, the user initiates a NEW dialogue at Point A. The automated interface expert then takes the initiative in the conversation, asking first for the name of the new file, then for the names of the file's fields. The file name will be used to distinguish the new file from others during the acquisition process. The field names are entered into the lexicon as the names of attributes and are put on an agenda so that further questions about the fields may be asked subsequently of the user.

At this point, TED still does not know what type of objects the data in the new file concern. Thus, as its next task, TED asks for words that might be used as generic names for the subjects of the file. Then, at Point E, TED acquires information about how to identify one of these subjects to the user and, at Point F, determines what kinds of pronouns might be used to refer to one of the subjects. (As regards ships, TED is fooled, because ships may be referred to by "she.")

TED is programmed with the knowledge that the identifier of an object must be some kind of name, rather than a numeric quantity or Boolean value. Thus, TED can assume a priori that the NAME field given in Interaction E is symbolic in nature. At Point G, TED acquires possible synonyms for NAME.

TED then cycles through all the other fields, acquiring information about their individual semantics. At Point H, TED asks about the CLASS field, but the user doesn't understand the question. By typing a question mark, the user causes TED to give a more detailed explanation of what it needs. Every question TED asks has at least two levels of explanation that a user may call upon for clarification. For example, the user again has trouble at J, whereupon he receives an extended explanation with an example. See T also.

Depending upon whether a field is symbolic, arithmetic or Boolean, TED makes different forms of entries in its lexicon and seeks to acquire different types of information about the field. For example, as at Points J, K and Y, TED asks whether symbolic field values can be used as modifiers (usually in noun-noun combinations). For arithmetic fields, TED looks for adjectives associated with scales, as is illustrated by the sequence OPQR. Once TED has a word such as OLD, it assumes MORE OLD, OLDER and OLDEST may also be used. (GOOD-BETTER-BEST requires special intervention.)

Note the aggressive use of previously acquired information in formulating new questions to the user (as in the use of AGE, and SHIP at Point P). We have found that this aids considerably in keeping the user focused on the current items of interest to the system and helps to keep interactions brief.

Once TED has acquired local information about a new file, it seeks to relate it to all known files, including the new file itself. At Points Z through B+, TED discovers that the *SHIP* file may be joined with itself. That is, one of the attributes of a ship is yet another ship (the escorted ship), which may itself be described in the same file. The need for this information is illustrated by the query the user poses at Point G+.

To better illustrate linkages between files, the transcript includes the acquisition of a second file about ship classes, beginning at Point J+. Much of this dialogue is omitted but, at L+, TED learns there is a link between the *SHIP* and *CLASS* files. At M+ it learns the direction of this link; at N+ and O+ it learns the fields upon which the join must be made; at P+ it learns the attributes inherited through the link. This information is used, for example, in answering the query at S+. TED converts the user's question "What is the speed of the hoel?" into "What is the speed of the class whose CNAME is equal to the CLASS of the hoel?."

Of course, the whole purpose of the NEW dialogues is to make it possible for users to ask questions of their databases in English. Examples of English inputs accepted by TED are shown at Points E+ through I+, and S+ and T+ in the transcript. Note the use of noun-noun combinations, superlatives and arithmetic. Although not illustrated, TED also supports all the available LADDER facilities of ellipsis, spelling correction, run-time grammar extension and introspection.

V THE PRAGMATIC GRAMMAR

The pragmatic grammar used by TED includes special syntactic/semantic categories that are acquired by the NEW dialogues. In our actual implementation, these have rather awkward names, but they correspond approximately to the following:

* <GENERIC> is the category for the generic names of the objects in files. Lexical properties for this category include the name of the relevant file(s) and the names of the fields that can be used to identify one of the objects to the user. See transcript Points D and E.
* <ID.VALUE> is the category for the identifiers of subjects of individual records (i.e., key-field values). For example, for the *SHIP* file, it contains the values of the NAME field. See transcript Point E.
* <MOD.VALUE> is the category for the values of database fields that can serve as modifiers. See Points J and K.
* <NUM.ATTR>, <SYM.ATTR>, and <BOOL.ATTR> are numeric, symbolic and Boolean attributes, respectively. They include the names of all database fields and their synonyms.
* <+NUM.ADJ> is the category for adjectives (e.g. OLD) associated with numeric fields. Lexical properties include the name of the associated field and files, as well as information regarding whether the adjective is associated with greater (as in OLD) or lesser (as in YOUNG) values in the field. See Points P, Q and R.
* <COMP.ADJ> and <SUPERLATIVE> are derived from <+NUM.ADJ>.

Shown below are some illustrative pragmatic production rules for nonlexical categories. As in the foregoing examples, these are not exactly the rules used by TED, but they do convey the nature of the approach.

```
<S> -> <PRESENT> THE <ATTR> OF <ITEM>
         what is the age of the reeves
       HOW <+NUM.ADJ> <BE> <ITEM>
         how old is the youngest ship
       <WHDET> <ITEM> <HAVE> <FEATURE>
         what leahy ships have a doctor
       <WHDET> <ITEM> <BE> <COMPLEMENT>
         which ships are older than reeves

<PRESENT> -> WHAT <BE>
             PRINT

<ATTR> -> <NUM.ATTR>
          <SYM.ATTR>
          <BOOL.ATTR>

<ITEM> -> <GENERIC>
            ships
          <ID.VALUE>
            reeves
          THE <ITEM>
            the oldest ship
          <MOD.VALUE> <ITEM>
            leahy ships
          <SUPERLATIVE> <ITEM>
            fastest ship with a doctor
          <ITEM> <WITH> <FEATURE>
            ship with a speed greater than 12

<FEATURE> -> <BOOL.ATTR>
               doctor / poisonous
             <NUM.ATTR> <NUM.COMP> <NUMBER>
               age of 15
             <NUM.ATTR> <NUM.COMP> <ITEM>
               age greater than reeves

<NUM.COMP> -> <COMP.ADJ> THAN
              OF
              <GREATER> THAN

<COMPLEMENT> -> <COMP.ADJ> THAN <ITEM>
                <COMP.ADJ> THAN <NUMBER>
```

These pragmatic grammar rules are very much like the ones used in LADDER [2], but they differ from those of LADDER in two critical ways.

(1) They capture the pragmatics of accessing databases without forcibly including information about the pragmatics of any one particular set of data.

(2) They use syntactic/semantic categories that support the processes of accessing databases, but that are domain-independent and easily acquirable.

It is worth noting that, even when a particular application requires the introduction of special-purpose rules, the basic pragmatic grammar used by TED provides a starting point from which domain-specific features can be added.

VI DIRECTIONS FOR FURTHER WORK

The TED system represents a first step toward truly portable natural-language interfaces to database systems. TED is only a prototype, however, and much additional work will be required to provide adequate syntactic and conceptual coverage, as well as to increase the ease with which systems may be adapted to new databases.

A severe limitation of the current TED system is its restricted range of syntactic coverage. For example, TED deals only with the verbs BE and HAVE, and does not know about units (e.g., the Waddel's age is 15.5, not 15.5 YEARS). To remove this limitation, the SRI NL group is currently adapting Jane Robinson's extensive DIAGRAM grammar [7] for use in a successor to TED. In preparation for the latter, we are experimenting with verb acquisition dialogues such as the following:

```
> VERB
 Please conjugate the verb
    (e.g. fly flew flown) > EARN EARNED EARNED
EARN is:
    1  intransitive (John dines)
    2  transitive (John eats dinner)
    3  ditransitive (John cooks Mary dinner)
    (Choose the most general pattern) > 2

 who or what is EARNED? > A SALARY
 who or what EARNS A SALARY? > AN EMPLOYEE
 can A SALARY be EARNED by AN EMPLOYEE? > YES
 can A SALARY EARN? > NO
 can AN EMPLOYEE EARN? > NO
    .
    .
 Ok., an EMPLOYEE can EARN a SALARY
 What database field identifies an EMPLOYEE? > NAME
 What database field identifies a SALARY? > SALARY
```

The greatest challenge to extending systems like TED is to increase their conceptual coverage. As pointed out by Tennant [8], users who are accorded natural-language access to a database expect not only to retrieve information directly stored there, but also to compute "reasonable" derivative information. For example, if a database has the location of two ships, users will expect the system to be able to provide the distance between them--an item of information not directly recorded in the database, but easily computed from the existing data. In general, any system that is to be widely accepted by users must not only provide access to primary information, but must also enhance the latter with procedures that calculate secondary attributes from the data actually stored. Data enhancement procedures are currently provided by LADDER and a few other hand-built systems, but work is needed now to devise means for allowing system users to specify their own database enhancement functions and to couple these with the natural-language component.

A second issue associated with conceptual coverage is the ability to access information extrinsic to the database per se, such as where the data are stored and how the fields are defined, as well as information about the status of the query system itself.

In summary, systems such as LADDER are of limited utility unless they can be transported to new databases by people with no significant formal training in computer science. Although the development of user-specifiable systems with extensive conceptual and syntactic coverage continues to pose a challenge to research, a polished version of the TED prototype, even with its limited coverage, would appear to have high potential as a useful tool for data access.

REFERENCES

1. L. R. Harris, "User Oriented Data Base Query with the ROBOT Natural Language Query System," Proc. Third International Conference on Very Large Data Bases, Tokyo (October 1977).

2. G. G. Hendrix, E. D. Sacerdoti, D. Sagalowicz, and J. Slocum, "Developing a Natural Language Interface to Complex Data," ACM Transactions on Database Systems, Vol. 3, No. 2 (June 1978).

3. G. G. Hendrix, "Human Engineering for Applied Natural Language Processing," Proc. 5th International Joint Conference on Artificial Intelligence, Cambridge, Massachusetts (August 1977).

4. G. G. Hendrix, D. Sagalowicz and E. D. Sacerdoti, "Research on Transportable English-Access Media to Distributed and Local Data Bases," Proposal ECU 79-103, Artificial Intelligence Center, SRI International, Menlo Park, California (November 1979).

5. R. C. Moore, "Handling Complex Queries in a Distributed Data Base," Technical Note 170, Artificial Intelligence Center, SRI International Menlo Park, California (October 1979).

6. P. Morris and D. Sagalowicz, "Managing Network Access to a Distributed Data Base," Proc. Second Berkeley Workshop on Distributed Data Management and Computer Networks, Berkeley, California (May 1977).

7. J. J. Robinson, "DIAGRAM: A Grammar for Dialogues," Technical Note 205, Artificial Intelligence Center, SRI International Menlo Park, California (February 1980).

8. H. Tennant, "Experience with the Evaluation of Natural Language Question Answerers," Proc. Sixth International Joint Conference on Artificial Intelligence, Tokyo, Japan (August 1979).

9. F. B. Thompson and B. H. Thompson, "Practical Natural Language Processing: The REL System as Prototype," pp. 109-168, M. Rubinoff and M. C. Yovits, eds., Advances in Computers 13 (Academic Press, New York, 1975).

10. D. Waltz, "Natural Language Access to a Large Data Base: An Engineering Approach," Proc. 4th International Joint Conference on Artificial Intelligence, Tbilisi, USSR, pp. 868-872 (September 1975).

EXAMPLE TED TRANSCRIPT

In this transcript, the user begins by telling TED about the following file.

FILE *SHIP*

NAME	CLASS	AGE	DOC	ESCORTED.SHIP
AMERICA	KITTY.HAWK	17.5	D	AMERICA
CONSTELLATION	KITTY.HAWK	19.0	D	CONSTELLATION
ENGLAND	LEAHY	15.0	D	CONSTELLATION
HOEL	ADAMS	17.0	N	AMERICA
REEVES	LEAHY	16.0	D	AMERICA
WADDEL	ADAMS	15.5	D	AMERICA

```
  |******************TRANSCRIPT STARTS HERE*******************
  |
  |
  |Welcome  to  TED! For instructions please type  "HELP."
  |
  |Type a question or command.  After any ">" type "?" for help.
  |
A |1> NEW                    [**User asks to introduce new file**]
  |
B |The name of the new file is (type file name) > *SHIP*
  |
C |The fields of file *SHIP* are (type sequence of fields)
  |> NAME CLASS AGE DOC ESCORTED.SHIP
  |
D |What names do you want to use to refer to a subject of the
  |*SHIP* file? > SHIP BOAT PLATFORM
  |
E |To identify a particular SHIP, which field values should
  |be listed? (type a sequence of fields) > NAME
  |
F |Is a SHIP human? (YES or NO) > N
  |
G |If there are other names for the attribute NAME, please list
  |them. (type a sequence of names or a <CR> if there are none)
  |> ID DESIGNATION
  |
H |CLASS is (A,F,S)? >?   [**User asks for help.
  |                        Gets longer form of question.**]
  |
I |CLASS is:
  |1  an arithmetic field (values  may be added, subtracted etc.)
  |2  a feature field (values are Boolean, T or F, YES or NO ...)
  |3  a symbolic  field (values  are usually nouns or adjectives)
  |(Please type 1, 2, or 3) > 3
```

```
J |Can  the  values  of  this symbolic field be used as modifiers?
  |(YES or NO) > ?
  |                                [**User asks for help.  Gets example.]
  || An example file is
  ||
  ||                         EMP*
```

NAME	SALARY	SEX	AGE	SPOUSE	STORE	DEPARTMENT
J.FISHER	17000	M	38	M.FISHER	PLAZA	HARDWARE
L.KING	12000	F	24	B.KING	PLAZA	SHOE
M.FISHER	15000	F	37	J.FISHER	PARK	SHOE

```
  ||
  ||      In  this  file  about employees, it makes sense to
  || talk  about "hardware employees" (employees in the hardware
  || department), "Plaza employees" (employees at the Plaza
  || store), but it makes less sense to speak of "M.Fisher
  || employees", or "38 employees".  Therefore, the value of the
  || STORE and DEPARTMENT fields are good modifiers,  but the
  || values of the SPOUSE field are not appropriate. TED does not
  || allow the values of arithmetic fields to be modifiers. Thus,
  || AGE is not a modifier field.
  ||      The actual values recorded in the database for the SEX
  || field of the example file are marginally suitable as
  || modifiers.  However, the synonyms MALE and FEMALE may
  || subsequently be declared for the values M and F.  Therefore
  || it is reasonable to declare the field SEX to be a
  || modification field.
  |
K |Can the values of this symbolic field be used as modifiers?
  |(YES or NO) > Y
  |
L |If there are other names for the attribute CLASS, please list
  |them. (type a sequence of names or a <CR> if there are none)
  |>
  |
M |AGE is (A,F,S)? > A        [**Arithmetic.  See H & I above.**]
  |
N |If there are other names for the attribute AGE, please list
  |them. (type a sequence of names or a <CR> if there are none)
  |>
  |
O |Corresponding adjective is? > ?  [*User asks for expanation.]
  |
P |If there is a word wwww such that the question
  |        HOW wwww IS THE SHIP?
  |is equivalent to
  |        WHAT IS THE AGE OF THE SHIP?
  |please type wwww (else type <CR>). > OLD
  |
Q |If there are synonyms for OLD, please list them (type a
  |sequence of words and multiwords) >
  |
R |If there are antonyms for OLD, please list them (type a
  |sequence of words and multiwords) > YOUNG NEW
```

```
S  |DOC is (A,F,S)? > F           [**Feature.  See H & I above.**]
   |
T  |Positive entry? > ?           [**User asks for explanation.**]
   |
U  |A feature field must have a single positive value that TED can
   |look for.  Typically this would be T, TRUE, YES, POSITIVE, etc.
   |What will this value be for DOC (type a single word) > D
   |
V  |If there are other names for the attribute DOC, please list
   |them. (type a sequence of names or a <CR> if there are none)
   |> DOCTOR (MEDICAL PERSONNEL)
   |
W  |ESCORTED.SHIP is (A,F,S)? > S   [**Symbolic. See H & I above.]
   |
X  |Can the values of this symbolic field be used as modifiers?
   |(YES or NO) > N
   |
Y  |If there are other names for the attribute ESCORTED.SHIP,
   |please list them. (type a sequence of names or a <CR> if there
   |are none) > (ESCORTED SHIP)
   |
Z  |Does the file *SHIP* have any links to itself? (Any particular
   |record in the file *SHIP* has some SHIP as it's subject. Call
   |that SHIP S1. In describing S1, it is possible that the record
   |may make reference to some second SHIP S2. If it does then
   |because S2 is itself a SHIP, S2 is  potentially  the  subject
   |of some other record in the file, thus providing a link from
   |the record of  the  file that describes S1 to the record that
   |describes S2. Does file *SHIP* have any such self-referencing
   |link?) (Type YES or NO) > Y
   |
A+ |the fields of the file *SHIP* are:
   |    NAME CLASS AGE DOC ESCORTED.SHIP
   |Each  record  of  the  file describes  some SHIP S1.
   |In describing S1, reference is make to some other SHIP S2.
   |The field (or fields) of the file that identify the second
   |SHIP S2 are (type a sequence of fields from the file)
   |> ESCORTED.SHIP
   |
B+ |In describing a SHIP S1, the file references a second SHIP S2
   |which is identified by a value V of the field ESCORTED.SHIP.
   |In the record of the file for which S2 is the subject, what
   |field will have V as its value? (type a field name) > NAME
   |
C+ |Does the file *SHIP* have another link to itself?
   |(YES or NO) > N
   |==============================================================
   |
   |Type a question or command.    After any ">" type "?" for help.
   |
D+ |2> EDIT     [****User calls the TED editor and enters values
   |             into the file he has just described to TED.
   |             These interactions are omitted here. ******]
```

```
   |Type a question or command.    After any ">" type "?" for help.
   |
E+ |3> WHAT ARE THE ID AND AGE OF THE NEWEST LEAHY SHIP?
   |        [**User asks a question in English about his data.**]
   |
   |What are the AGEs (*SHIP*#2) and NAMEs (*SHIP*#2) of the
   |NEWEST SHIPs (*SHIP*#2) with CLASS (*SHIP*#2) equal to LEAHY?
   |              [**System prints its interpretation of query**]
   |
   |---------------
   ||NAME     AGE |
   ||-------------|
   ||ENGLAND 15.0 |
   |---------------
   |
F+ |4> BY ESCORTED SHIP AND CLASS, WHAT ARE THE NAMES AND AGES OF
   |     SHIPS WITH MEDICAL PERSONNEL?
   |
   |What are the CLASSes (*SHIP*#7) and ESCORTED.SHIPs (*SHIP*#7)
   |and AGEs (*SHIP*#7) and NAMEs (*SHIP*#7) of the SHIPs
   |(*SHIP*#7) with DOC (*SHIP*#7) equal to D?
   |
   |-----------------------------------------------------
   ||ESCORTED.SHIP  |CLASS      |NAME          AGE  |
   ||---------------|-----------|------------------|
   ||AMERICA        |ADAMS      |WADDEL        15.5 |
   ||AMERICA        |KITTY.HAWK |AMERICA       17.5 |
   ||AMERICA        |LEAHY      |REEVES        16.0 |
   ||CONSTELLATION  |KITTY.HAWK |CONSTELLATION19.0 |
   ||CONSTELLATION  |LEAHY      |ENGLAND       15.0 |
   |-----------------------------------------------------
   |
G+ |5> WHAT IS THE AGE OF THE ESCORTED SHIP OF THE REEVES?
   |
   |What are the NAMEs (*SHIP*#9) and AGEs (*SHIP*#9) of the SHIP
   |(*SHIP*#9) with  NAME equal to the ESCORTED.SHIP of REEVES
   |(*SHIP*#8)?
   |         [**To answer question, TED joins file SHIP to itself,
   |              introducing variables for two separate ships.**]
   |
   |---------------
   ||NAME     AGE |
   ||-------------|
   ||AMERICA 17.5 |
   |---------------
   |
H+ |6> WHAT IS THE AVERAGE AGE OF SHIPS?
   |
   |What is the MEAN AGE of the SHIPs (*SHIP*#10)?
   |
   |The Answer is MEAN equals 16.66667
   |
I+ |7> HOW MANY LEAHY SHIPS HAVE DOCTORS?
   |
   |Count the SHIPs (*SHIP*#18) with CLASS (*SHIP*#18) equal to
   |LEAHY and DOC (*SHIP*#18) equal to D
   |
   |The Answer is CNT equals 2
```

```
J+ |8> NEW                [**User asks to introduce new file. **]
   |    ***********************************************************
   |    *                                                         *
   |    *  The new file has the form:                             *
   |    *                                                         *
   |    *                   FILE *CLASS*                          *
   |    *          ---------------------------------------------  *
   |    *          |CNAME       |TYPE SPEED ASW NUMBER.MADE |     *
   |    *          |------------|--------------------------|     *
   |    *          |ADAMS       |DDG  32.4  Y    23         |     *
   |    *          |KITTY.HAWK  |CVA  31.2  N    3          |     *
   |    *          |LEAHY       |DLG  34.0  Y    9          |     *
   |    *          ---------------------------------------------  *
   |    *                                                         *
   |    *  We omit most of the acquisition dialogue, taking up    *
   |    *  again at the end of the NEW sequence.                  *
   |    ***********************************************************
   |
K+ |Does the file *CLASS* have any links to itself? (Any particular
   |record in the file *CLASS* has some CLASS as it's subject. Call
   |that CLASS C1.  In describing C1, it is possible that the
   |record may make reference to some second CLASS C2. If it does
   |then because C2 is itself a CLASS, C2 is potentially the
   |subject of some other record in the file, thus providing a link
   |from the record of  the  file that describes C1 to the record
   |that describes C2. Does file *CLASS* have any such self-
   |referencing link?)      (Type YES or NO) > NO
   |
L+ |*CLASS* has links to or from which of the following files:
   |(*CLASS* *SHIP*) > *SHIP*
   |
M+ |For the link between *CLASS* and *SHIP*, which of the following
   |is true:
   |        1) For each SHIP which is the subject of one of
   |the records of file *SHIP* , *SHIP* provides information
   |associating that SHIP with a unique CLASS of file *CLASS*
   |        2) For each CLASS which is the subject of one of
   |the records of file *CLASS* , *CLASS* provides information
   |associating that CLASS with a unique SHIP of file *SHIP*
   |
   |> 1
   |
N+ |The fields for the two files in question are:
   |      *CLASS*:    CNAME TYPE SPEED ASW NUMBER.MADE
   |      *SHIP*:     NAME CLASS AGE DOC ESCORTED.SHIP
   |The fields of file *SHIP* which identify a CLASS of file
   |*CLASS* are (type a sequence of fields from file *SHIP*)
   |> CLASS
   |
O+ |The field of *CLASS* associated with field CLASS of file
   |*SHIP* is > CNAME
```

```
P+ |The following are properties of CLASS which are the subjects
   |of the *CLASS* file:
   |     CNAME  TYPE  SPEED  ASW  NUMBER.MADE
   |The current link associates with each SHIP of file *SHIP* a
   |CLASS of file *CLASS* . Which properties of the CLASS (from
   |file *CLASS*) are inherited through this link as properties
   |of the associated SHIP (from file *SHIP*)? > TYPE SPEED ASW
   |
Q+ |In addition to the links previously declared, file *CLASS* is
   |linked to which of the following files: *CLASS* *SHIP*
   |(Type the name of a previously declared file or <CR>)? >
   |==============================================================
   |
   |Type a question or command.   After any ">" type "?" for help.
   |
R+ |9> EDIT       [***User calls the TED editor and populates the
   |               *CLASS* file with the data shown at J+ above.**]
   |
   |
   |Type a question or command.   After any ">" type "?" for help.
   |
S+ |10> WHAT IS THE SPEED OF THE HOEL?
   |
   |What are the NAMEs (*SHIP*#22) and SPEEDs (*CLASS*#23) of HOEL
   |(*SHIP*#22) with  CLASS (*SHIP*#22) equal to the CNAME
   |(*CLASS*#23) of the CLASSes (*CLASS*#23)?
   |
   |-------------    [**Note join on the *SHIP* and *CLASS* files**]
   ||NAME SPEED |
   ||-----------|
   ||HOEL 32.4  |
   |-------------
   |
T+ |11> BY EXCORTED SHIP AND TYPE, WHAT ARE THE NAMES AND CLASSES
   |OF SHIPS WITH ANTISUB WEAPONS?
   |
   |      ESCORTED <-spelling
   |
   |What are the TYPEs (*CLASS*#26) of the CLASS (*CLASS*#26) with
   |CNAME equal to  the CLASS of the SHIPs (*SHIP*#24) with CLASS
   |(*SHIP*#24) equal to the CNAME (*CLASS*#25) of the CLASSes
   |(*CLASS*#25) with ASW (*CLASS*#25) equal to Y?
   |          [**The print routine for the system's interpretation
   |             contains bugs, but answer is computed properly.**]
   |---------------------------------------
   ||ESCORTED.SHIP  |TYPE |NAME    CLASS |
   ||---------------|-----|--------------|
   ||AMERICA        |DDG  |HOEL    ADAMS |
   ||AMERICA        |DDG  |WADDEL  ADAMS |
   ||AMERICA        |DLG  |REEVES  LEAHY |
   ||CONSTELLATION  |DLG  |ENGLAND LEAHY |
   |---------------------------------------
```