EUFID: A FRIENDLY AND FLEXIBLE FRONT-END FOR DATA MANAGEMENT SYSTEMS

Marjorie Templeton
System Development Corporation, Santa Monica, CA.

EUFID is a natural language frontend for data management systems. It is modular and table driven so that it can be interfaced to different applications and data management systems. It allows a user to query his data base in natural English, including sloppy syntax and misspellings. The tables contain a data management system view of the data base, a semantic/syntactic view of the application, and a mapping from the second to the first.

We are entering a new era in data base access. Computers and terminals have come down in price while salaries have risen. We can no longer make users spend a week in class to learn how to get at their data in a data base. Access to the data base must be easy, but also secure. In some aspects, ease and security go together because, when we move the user away from the physical characteristics of the data base, we also make it easier to screen access.

EUFID is a system that makes data base access easy for an untrained user, by accepting questions in natural English. It can be used by anyone after a few minutes of coaching. If the user gets stuck, he can ask EUFID for help. EUFID is a friendly but firm interface which includes security features. If the user goes too far in his questions and asks about areas outside of his authorized data base, EUFID will politely misunderstand the question and quietly log the security violation.

One beauty of EUFID is its flexibility. It is written in FORTRAN for a PDP-11/70. With minor modifications it could run on other mini-computers or on a large computer. It is completely table driven so that it can handle different data bases, different views of the same data base, or the same view of a restructured data base. It can be interfaced with various data management systems--currently it can access a relational data base via INGRES or a network data base via WWDMS.

EUFID is an outgrowth of the SDC work on a conceptual processor which was started in 1973.[1] It is now demonstrable with a wide range of sentences questioning two data bases. It is still a growing system with new power being added.

In the following sections we will explore the features that make EUFID so flexible and easy to use. The main features are:

- natural English
- help
- semantic tables
- data base tables
- mapping tables
- intermediate language
- security

1. NATURAL ENGLISH

EUFID has a dictionary containing the words that the users may use when querying the data base. The dictionary describes how words relate to each other and to the data base. Unlike some other natural language systems, EUFID has the words in the sentence related to fields in the data base by the time the sentence is "understood." More will be said about this process in the section on semantic tables.

EUFID is forgiving of spelling and grammar errors. If it does not have a word in the dictionary, but has a word that is close in spelling, it will ask the user if a substitution can be made. It also can "understand" a sentence even when all words are not present or some words are not grammatically correct. For example, any of these queries are acceptable:

"What companies ship goods?"
"Companies?" (list all companies)
"What company shop goods?"
("shop" will be corrected to "ship". The plural "companies" will be assumed)

Users are free to structure their input in any way that is natural to them as long as the subject matter covers what is in the data base. EUFID would interpret these questions in the same way:

"Center shipped heavy freight to what warehouses in 1976?"
"What warehouses did Center ship heavy freight to in 1976?"

Each user may define personal synonyms if the vocabulary in the dictionary is not rich enough for him. For example, for efficiency a user might prefer to use "wh" for "warehouse" and "co" for "company". Another user of the same data base might define "co" for "count".

2. HELP

Basically, EUFID has only four commands. These are "help", "synonym" (to define a synonym), "comment" (to criticize EUFID), or "quit". These four commands are described in the help module as well as the general guidelines for questions.

If the user hits an error while using EUFID, he will receive a sentence or two at his terminal which describes the problem. In some cases he will be asked for clarification or a new question as shown in these exchanges.

User: "What are the names of female secretaries' children?"
EUFID: "Do you mean
(1) female secretaries or
(2) female children?"
User: "2"
or
User: "What is the salary of the accounting department?"
EUFID: "We are unable to understand your question because "salary of department" is not meaningful. Please restate your question."

If the description is not enough to clarify the problem, the user can ask for help. First, HELP will give a deeper description of the problem. If that is not enough, the user can ask for additional information which may include a list of valid questions.

3. TABLES

EUFID is application and data base independent. This independence is achieved by having three sets of tables--the semantic dictionary tables, the data base tables, and the mapping tables which map from the semantic view to the data base. Conceivably, a single semantic view could map to two data bases that contain the same data but are accessed by different data management systems.

3.1 SEMANTIC TABLES

The semantic view is defined by an application expert working with a EUFID expert. Together they determine the ways that a user might want to talk about the data. From this, a list of words is developed and the basic sentence structures are defined. Words are classed as:

entities	(e.g., company)
events	(e.g., send)
functions	(after 1975)
parts of a phrase or idiom	(map coordinates)
connectors	(to)
system words	(the)
anaphores	(it)
two or more of the above	(ship an entity plus ship an event)

An entity corresponds approximately to a noun and an event to a verb. Connectors are prepositions which are dropped after the sentence is parsed. System words are conjunctions, auxiliaries, and determiners which participate in determining meaning but do not relate to data base fields. Anaphores are words that refer to previous words and are replaced by them while parsing. Basically then, the only words that relate to the items in the data base are entities, events, and functions.

Entities and events are defined using a case structure representation which combines syntactic and semantic information. Lexical items which may co-occur with an entity to form noun phrases, or with a verb to form verb phrases, fill cases on the entity or event. Cases are distinguished by the set of possible fillers, the possible connectors, and the syntactic position of the case relative to the entity or event. A case may be specified as optional or obligatory.

A sense of an entity or event is defined by the set of cases which form a distinct noun phrase or verb phrase type. Three senses of the word "ship" are illustrated in Figure 1.

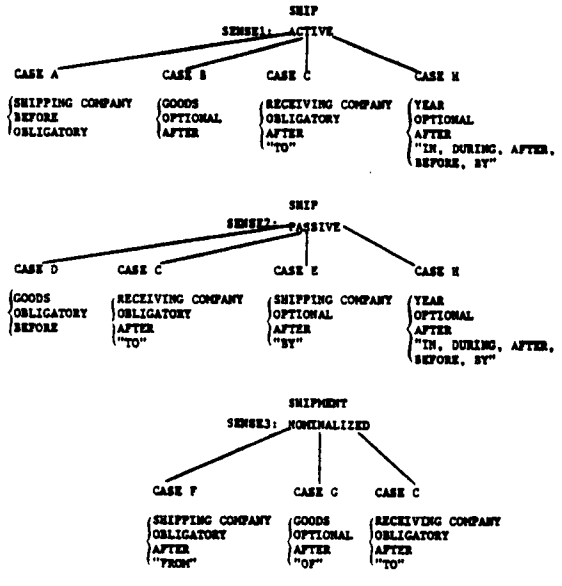

Figure 1.

The first sense of "ship" accounts for active voice verb phrases with the pattern "Companies ship goods to companies in year."

Examples are:

What companies ship to Ajax?
In 1976, who shipped light freight to Colonial?

This sense of "ship" has two obligatory cases, A and C, and two optional cases B and H. The fact that the "year" case can be moved optionally within the phrase is not represented within the case structure, but is recognized by the Analyzer, which assigns a structure to the phrase.

The second sense of "ship" accounts for the passive construction of the type "Goods are shipped to company by company."

Examples are:

Was light freight shipped to Ajax in 1978?
What goods were shipped to Ajax by Colonial?
By what companies in 1975 was heavy freight shipped to Colonial?

Case D has the same filler as case B, but precedes "ship" and is obligatory. Case E has the same filler as case A, but follows "ship", has a different connector, and is optional. That is, sense 1 of "ship" is defined as the association of "ship" with cases A,B,C. Sense 2 is the association of "ship" with cases C,D,E. Sense 3 of "ship" describes the nominalized form "shipment" and explicitly captures the information that shipments involve goods and reflect transactions between companies.

An example is:

"What is the transaction number for the shipment of bolts from Colonial to Ajax?"

3.2 DATA BASE TABLES

The data base tables describe the data base as viewed by the data management system. Since all data management systems deal with data items organized into groups that are related through links, it is possible to have a common table format for any data management system.

The data base tables actually consist of two tables. The CAN table contains information about groups and data items. A group (also called entity or record in other systems) is identified by the group name. A data item in the CAN table consists of the data item name, the group to which it belongs, a unit code, an output identifier, and some field type information. Notably missing is anything about the byte within the record or the number of bytes. EUFID accesses the data base through a data management system. Therefore, the data can be reorganized without changing the EUFID tables as long as the data items retain their names and their groupings.

The second data base table is the REL table which contains an entry for each group with its links to other groups. For network data bases, the link is the chain name for the primary chain that connects master and detail records. For relational data bases, every data item pair in the two groups that can have the same value is a potential link.

3.3 MAPPING TABLES

The mapping tables tell the program how to get from the semantic node, as found in the semantic dictionary, to the data base field names. Each entry in the mapping table has a node name followed by two parts. The first part describes the pattern of cases and their fillers for that node name. The second part is called a production and it gives the mapping for each case filler. A node may map to a node higher in the sentence tree before it maps to a data base item. For example, "company name" in the question "What companies are located in Los Angeles?" may map to a group containing general company information. However, "company name" in the question "What companies ship to Los Angeles?" may map to a group containing shipping company information.

Therefore, it is necessary to first map "company name" up to a higher node that determines the meaning. At the point where a unique node is determined, the mapping is made to a data item name via the CAN table. This data item name is used in the generation of the query to the data management system.

4. INTERMEDIATE LANGUAGE

EUFID is adaptable to most data management systems without changes to the central modules. This is accomplished by using an intermediate language (IL). The main parts of EUFID analyze the question, map it to data items, and then express the query in a standard language (IL). A translator is written for each data management system in order to rephrase the IL query into the language of the data management system. This is an extra step, but it greatly enhances EUFID's flexibility and portability.

The intermediate language looks like a relational retrieval language. Translating it into QUEL is straightforward, but translating it to a procedural language such as WWDMS is very difficult. The example below shows a question with its QUEL and WWDMS equivalent.

QUESTION: WHAT ARE THE NAMES AND ADDRESSES OF THE EXECUTIVE SECRETARIES IN R&D?

INGRES IL:

```
RETRIEVE [JOB.EMPLOYEE,JOB.ADDRESS]
  WHERE (DIV.NAME = "R&D")
  AND (DIV.JOB = JOB.NAME)
  AND (JOB.NAME = "SECRETARY")
  AND (JOB.CLASS = "EXECUTIVE")
```

QUEL:

```
range of div is div
range of job is job
retrieve (job.employee,job.address)
  where div.name = "R&D")
  and div.job = job.name
  and job.name = "secretary"
  and job.class = "executive"
```

WWDMS IL:

```
RETRIEVE [JOB.EMPLOYEE,JOB.ADDRESS]
  WHERE (DIV.DNAME = "R&D")
  AND (DIV.DIV_JOB_CH = JOB.DIV_JOB_CH)
  AND (JOB.JNAME = "SECRETARY")
  AND (JOB.CLASS = "EXECUTIVE")
```

WWDMS QUERY:

```
INVOKE 'WWDMS/PERSONNEL/ADF'
REPORT EUFID-1 ON FILE 'USER/PASSWD/EUFID'
  FOR TTY
Q1.  LINE "EMPLOYEE NAME =",EMPLOYEE
Q2.  LINE "ADDRESS =",ADDRESS
R1.  RETRIEVE E-DIV
       WHERE DNAME = "R&D"
     WHEN R1.
R2.  RETRIEVE E-JOB
       WHERE JNAME = "SECRETARY"
       AND CLASS = "EXECUTIVE"
     WHEN R2
     PRINT Q1
     PRINT Q2
     END
```

5. SECURITY

EUFID protects the data base by removing the user from direct access to the data management system and data base. At the most general level, EUFID will only allow users to ask questions within the semantics that are defined and stored in the dictionary. Some data items or views of the data could be omitted from the dictionary.

At a more specific level, EUFID controls access through a user profile table. Before a user can use EUFID, a system person must define the user profile. This table states which applications or subsets of applications are available to the user. One user may be allowed to query everything that is covered by the semantic dictionary. Another user may be restricted in his access.

The profile table is built by a concept graph editor. When a new login id is established for EUFID, the system person gives the application name of each application that the user may access. Associated with an application name is a set of file names of the tables for the application. If access is to be restricted, a copy of the CAN and mapping function tables is made. The copies are changed to delete the data items which the user is not to know about. The names of the restricted tables are then stored in the user's profile record. EUFID will still be able to find the words that are used to talk about the data item, but when EUFID maps the word to a removed data item it responds to the user as though the sentence could not be understood.

6. CONCLUSION

EUFID is a system that makes data base access easy and direct for an end user so that he does not need to go through a specialist or learn a language to query his own data base. It is modular and table driven so that it can be interfaced with different data management systems and different applications. It is written in high-level transportable languages to run on a small computer for maximum transportability. The case grammar that it uses allows flexibility in sentence syntax, ungrammatical syntax, and fast, accurate parsing.

If the reader wants more detail he is referred to references 2-4.

7. REFERENCES

1. Burger, J., Leal, A., and Shoshani, A. "Semantic Based Parsing and a Natural-Language Interface for Interactive Data Management," AJCL Microfiche 32, 1975, 58-71.

2. Burger, John F. "Data Base Semantics in the EUFID System," presented at the Second Berkeley Workshop on Distributed Data Management and Computer Networks, May 25-27 1977, Berkeley, CA.

3. Weiner, J. L. "Deriving Data Base Specifications from User Queries," presented at the Second Berkeley Workshop on Distributed Data Management and Computer Networks, May 25-27, 1977, Berkeley, CA.

4. Kameny, I., Weiner, J., Crilley, M., Burger, J., Gates, R., and Brill, D. "EUFID: The End User Friendly Interface to Data Management Systems," SDC, September 1978.