BRIAN C. VICKERY

Tome Associates, Ltd.
London, England

Intelligent Interfaces to Online Databases

ABSTRACT

The possible functions of intelligent interfaces are summarized. Five examples of recent or current European projects on the development of interfaces are described: INSERM, CIRT, EURISKO, ERLI/MINITEL, and IMIS. A number of the problems of interface design and implementation are reviewed.

INTRODUCTION

There has been an enormous investment in publicly available and corporate databases—bibliographic, numerical, directory, full-text, and so on. Despite all the aids provided by search services (text retrieval and database management systems), online access to databases remains difficult for many potential users. The user may need to know a variety of communications protocols, host command languages, search techniques, database file structures, and subject terminologies. In Europe, the natural language of the database may not be that of the user.

The aim of an intelligent interface is to make access easier by building some of the needed knowledge into front-end software used to interrogate the online search system. This aim does not coincide with that of creating an intelligent retrieval system. An interface accesses existing online systems, with all their constraints and deficiencies, so it can only be as successful as the online search system allows it to be. An interface does not address the problem of restructuring the database or the search system itself to make retrieval more intelligent.

In Europe, we are well aware of the pioneering work on intelligent retrieval that has been carried out in the United States—names such as Doszkocs and Croft come immediately to mind. But this article presents some of the work on intelligent interfaces that has been and is currently being carried out in Europe.

The Commission of the European Community has a division entitled "Telecommunications, Information Industries and Innovation"; a section of this has been particularly active in promoting interface work. During 1988-89, this section funded two "state-of-the-art" reviews by Cognitec (1988) and Vickery (1989). In 1989, it awarded two major contracts:

1. DISNET: an intelligent interface to online information to be implemented partly on a personal computer and partly on a host or network. The system will provide some general interface functions but also some specific to particular subject domains—agriculture and microbiology.
2. MITI: an intelligent multilingual interface (IMIS) on personal computers, to access a number of hosts. More will be said about this later in this paper.

FUNCTIONS OF AN INTELLIGENT INTERFACE

Online search of a database might be aided in a number of ways, aiming to:

- choose appropriate databases and hosts;
- permit the enquirer to state an information want in his/her own words;
- assist in clarifying the expression of the want;
- establish the level (introductory? advanced?) and approach (practical? theoretical?) of the information required;
- adjust the scope of the want (now become a query) so that the volume of retrievable information and the cost of the search are acceptable;
- formulate the query in the vocabulary used in the chosen databases;
- express the query as a search statement in the required format (e.g., using Boolean operators);
- handle the "housekeeping" activities of dialup, logon, file selection, downloading, and document ordering;
- transmit the search statement to the host using the appropriate command language, and, if necessary, switch between hosts and command languages;
- in search amendment, change the Boolean or other search operators,

and/or change search terms by various means, including relevance feedback; and

- present the search output in a helpful form, e.g., by ranking in order of probable relevance.

In a European context, a further important function is to aid the user whose natural language is not that of the database. Four levels of multilingual facility may be envisaged:

1. Multilanguage screen messages but query input in the language of the database(s) to be searched.
2. Input of search terms in one language and their immediate translation into the language of the database. (In these two options, if the user is not familiar with the language of the database, there can be little intelligent interaction with her/him in formulating and modifying the search strategy.)
3. Full processing of queries in more than one language, with translation of the final search statement into the language of the database. (In this option, full interaction can be achieved.)
4. Translation of search output into the language of the user. (This facility can in principle be added after any of the first three options, though it goes beyond interface functions into full-text translations.)

Following are five examples of European work on the development of interfaces exhibiting various degrees of "intelligence."

INSERM Interface

The French *Instit National de la Sante et de la Recherche Medicale* have developed an interface for searching MEDLINE on TELESYSTEMES-QUESTEL via the videotex system MINITEL (Halpern & Sargeant, 1988). The system uses the standard Minitel terminal and the user is prompted via menus.

A menu asks for entry of the major search criterion: French keyword, English keyword, English textwords, Author or Journal. When a criterion has been chosen, the user is prompted to enter a search term. If this is a textword, it is immediately searched. If it is a keyword, a listing of the MeSH keywords alphabetically surrounding the chosen term is displayed (in English or French), from which the user makes a selection. If the user initially enters a term that is a nonpreferred synonym in MeSH, then the preferred synonym and its alphabetical neighbors are displayed.

Once a keyword has been chosen from the alphabetical display, the user has the option of selecting more specific terms from the hierarchical MeSH thesaurus. The keywords one level lower in the hierarchy are displayed, from which the user may choose one; the process may be repeated down to lower levels until the user considers that his/her topic has been precisely expressed. The system now displays the MeSH subheadings that are valid for the search term, and the user is invited to select one or more of these.

From the keyword finally chosen, a subset is formed: the term is automatically ORed to all the more specific keywords derived from "exploding" that section of the MeSH hierarchy, the whole subset being linked to the chosen subheadings.

An initial search is now carried out. The number of references retrieved is displayed, and the user may either inspect them or narrow the search. To narrow the search, the user is presented with a further menu that asks for secondary search criteria, which may be of the same kind as the first or may be a limitation of the search. If the choice is to select a second criterion comparable to those in the first menu, the same procedure is followed as before, ending up with the choice of another search term. This term (or a subset derived from it) is then ANDed with the search based on the first criterion. This process of narrowing the search statement can be repeated.

Alternatively, a search may be limited to French language items or to clinical articles, or a search term may be required to be present in the "major keyword" field of a database record.

CIRT

CIRT is an experimental microcomputer front-end for searching certain databases, particularly MEDLINE, on the DATASTAR host (Robertson & Thompson, 1987). The user logs in to DATASTAR (using stored user i.d., password, and database name) and is then asked to specify limits (year, language, MEDLINE check tags such as human/animal, female/male, etc.). Subsequent interaction makes use of CIRT command language.

The user enters query terms which can be natural language words or MeSH terms. Any MeSH search facility can be used (e.g., explosion) or DATASTAR facility (e.g., truncation, adjacency). For example, suppose that three search terms A, B, C are entered. The system carries out the following searches:

```
1. A OR B OR C
2.     A AND B
3.            A AND B AND C
4.            A AND B AND NOT C
5.      A AND NOT B
6.            A AND C AND NOT B
7.            A AND NOT B AND NOT C
8.      B AND NOT A
9.            B AND C AND NOT A
```

```
10.          B AND NOT A AND NOT C
11.     C AND A
12.          C AND NOT A AND NOT B
```

If a second-level search such as A AND B is reported by DATASTAR as retrieving nothing, then subsearches 3 and 4 are not made; the same would apply at searches 5 and 8. Weights are calculated for each of the third-level searches carried out, and up to fifteen records are downloaded from each set. The weight of each search term is inversely proportional to its postings in the database, and the weight for a retrieved set is the sum of the weights of its matching terms.

The retrieved sets are now ranked in decreasing weight order, and set details are presented to the user. The user can inspect items in each set in sequence from the (fifteen or less) items downloaded and, if desired, mark some items as relevant. When inspection of a particular set is completed, set weights are recalculated using a new estimate for term weight. A new term can be added with the effect that the necessary additional searches are carried out and set weights recalculated. A search term can be deleted, which has the effect of setting the term weight to zero. Individual items checked as relevant, or complete sets, can be selected for printing out record details offline.

EURISKO

This system has been implemented on microcomputer for an intelligent search interface, operating at present for searches of thirty databases on the TELESYSTEMES and CEDOCAR hosts (Barthes & Glize, 1988). The user is asked to choose a subject area of interest from a menu and to enter a subject query in French (English terms may also be used). A semantic grammar of fifty rules then analyzes the query, extracting data on the type of document requested, on any author name or language, and on the subset terms present. The system then tries to acquire further search-specific information from the user, asking, for example:

—If you wish to truncate "dyadic," give me the root.
—How many characters should be sought after truncation?
—Is the word "and" in "dyadic functions and piezoelectricity" a link between two concepts (y/n)?

Based on the subject area of interest and on the type of document requested, a list of databases is displayed to the user in order of probable relevance. Several databases may be selected by the user. Connection to the selected hosts and databases is automatically established, and the system prepares to transmit a search request.

A query on "chemical composition of the leaf of sweet corns and of the stalk of sorgho" would be analyzed into components which are tagged as follows.

root	truncation	operator
1 - chemical	N	PROX
2 - composition	N	AND
3 - lea???	Y	AND
4 - sweet	N	PROX
5 - corn?	Y	OR
6 - stalk?	Y	PROX
7 - sorgho	N	NIL

The operators indicate the relation of a term to the following term in the sequence. The PROX operator implies that a proximity operator will be needed between the two terms. The search terms and appropriate operators are transmitted to the host system one by one under the control of a set of sixty rules. In this case, the following search statements would be generated, and the number of postings would be returned by the host at each stage:

1 - chemical PROX composition
2 - lea???
3 - sweet PROX corn?
4 - 2 AND 3
5 - stalk?
6 - sorgho
7 - 5 AND 6
8 - 4 OR 7
9 - 1 AND 8

At each step, errors can be recognized that need correction. For example, at any step the number of postings might be zero. Rules control the actions that the system then takes, for example, to ask the user for a synonym of a zero-posted term, which is then used for search. If the overall search retrieves zero postings, the search must be broadened in consultation with the user.

When the search has retrieved some items, these are displayed to the user for a relevance judgment, which may be that the results are too general, or too specific, or relevant but insufficient, or off-focus. Appropriate action is then taken. For example, to narrow a search, the system interrogates the user in turn about:

- amending truncation to be more specific,
- eliminating ORed terms,
- adding new ANDed terms,
- altering the operators, e.g., changing AND to PROX,

- restricting search to named fields, or to specified dates, or to types of documents, or to language.

After each amendment, a fresh search and evaluation takes place.

ERLI/MINITEL

The firm ERLI has developed an interface via MINITEL terminals to the professional headings of the French Yellow Pages directory (Clemencin, 1988). The system naturally uses French, but in the description below, the examples will be mainly in English.

The Yellow Pages are normally accessed through about 2,500 headings, e.g., lampshades (manufacturing and trade), estate managers and co-ownership trustees, rubber products for sanitary use (manufacturing); domestic vacuum cleaners and floor polishers; or typewriter, accounting and invoicing machine hire.

Headings may have subheadings chosen from a standard set or assigned by the agency listed in the directory. The technique normally used to access the directory is by keywords: a user query such as "I would like to book seats on a holiday tour" is analyzed to eliminate "empty" words and a Boolean expression is created: AND (book, holiday, seat, tour). This is then used to search for headings containing the ANDed words. If no output is obtained, the expression OR (book, holiday, seat, tour) is searched; this all too often results in a match with many headings.

The ERLI interface differs in two ways: the headings are indexed as described below and are approached via the index, not directly, and queries are handled by language processor. A study of the headings used in the Yellow Pages indicated that they contained three kinds of words:

1. so-called "predicates" expressing the activity of an agency in the directory, e.g., sales, manufacture, repair, hire, retail;
2. "empty" words such as supplies, equipment, contractor; and
3. "primary" words—the main bulk of words referring to objects such as furniture, cars, etc., or names of professions such as printer, surgeon, architect.

The index contains 20,000 entries. Each entry consists of a single or compound word. Rules allow the recognition of a word through all its inflectional variants (e.g., *social, sociale, sociales, sociaux; sport, sportif; fabriquer, fabrication*). Compounds may be of various kinds, e.g., *salle de bain, pomme de terre, train electrique (miniature)*.

To each entry is attached a grammatical category, links to terms that are semantically related, and pointers to the headings which it indexes. Each heading is indexed either by its primary term alone or by a compound of an empty word and a primary term. Predicates are indicated in the form of a relation between the index entry and the heading; there are twelve such relations, their English equivalents being: retail, wholesale, manufacture, repair, renting, transport, design, medical care, reservation, lessons, training, and custom-made contracts. An entry may be linked via several predicate relations to a number of headings. The semantic links are to synonyms and to broader terms.

Analysis of user input begins by recognition of single words, and each word is looked up, in turn, in the index. If it occurs more than once (i.e., it is ambiguous), rules are invoked that take context into account to resolve the ambiguity. If the word as entered is not in the index, stemming rules derive a standard form and variants of this are sought. If still not found, a spelling correction procedure is invoked that creates a phonological representation of the word; this is compared with the phonological representations of single index words. If still not identified, the word is treated as unknown. Compounds occurring in the input are identified next. Terms which are synonyms for a preferred term (as used in headings) are replaced by the heading terms.

The treated input is now processed by sets of grammatical rules, which identify elements as conjunctions, standard subheadings, predicates, and primary terms, and take appropriate actions to transform the input into a query that can be matched against Yellow Pages headings. An example of this process is the treatment of the input query (here given in English):

"steel rim for car wheel"

The string contains only primary terms. It does not occur as it stands in the index, although the individual words are known. The system tries to generate variants by using broader terms:

"steel accessory for car"
"steel rim for wheel of vehicle"
"metal rim for car wheel"

These are not found in the index so the system simplifies the input by dropping terms, to give the searches:

"car wheel"
"steel rim"

and then again tries broader terms, achieving an index match with:

"car accessory"

which points to a Yellow Pages heading:

automobiles (detached components and accessories).

This is searched together with the other original primary terms ("rim" and "steel") as subheadings.

IMIS

This year the European Commission awarded a contract to a consortium to develop an intelligent multilingual interface to databases, mounted on an IBM PC and accessing, in the first instance, a number of European hosts. The consortium partners include Tome Associates (UK) who have previously developed the commercial software TOME SEARCHER and TOME SELECTOR; the University Paul Sabatier, developers of EURISKO; and Softex GmbH, a German firm specializing in multilingual text processing. A new interface is to be constructed, building on the products and techniques already existing among the partners.

The functional scope of the proposed interface can be seen from the figure titled IMIS in Action. The user will be able to choose one of four languages in which to interact with the system: English, French, German, and Spanish. She/he will then be asked to indicate the general subject area of the query—if necessary, being guided down a hierarchical menu of subjects. The system will display descriptions of databases in the chosen subject area, and the user will select one or more of these and, if necessary, the preferred host. This part of the new interface will be based on the existing TOME SELECTOR.

At this point, several alternatives will be available:

1. It may be that the user wishes to access a host "not known to IMIS." The system will then be used simply as a communications package, and the user him/herself must dial up, provide identifier and passwords, logon, and carry out a normal search.
2. The host may be "known to IMIS" but the user does not want "aided search." In this case, the system will provide automatic dial-up and logon, but the user must input a search using the command language, Boolean operators, and other search techniques of the chosen host.
3. The user wants "aided search" but in a subject area not covered by the IMIS dictionaries. She will in this case be guided in query development along the "user-based" path: essentially, the system will use the procedures described in EURISKO.
4. The user wants "aided search" in a subject area covered by the IMIS dictionaries (the subject areas to be covered are technology in general and environmental information). In these areas, the user will be guided along the "thesaurus-based" path, which will be a development of the existing TOME SEARCHER procedures now to be described (Vickery, 1988).

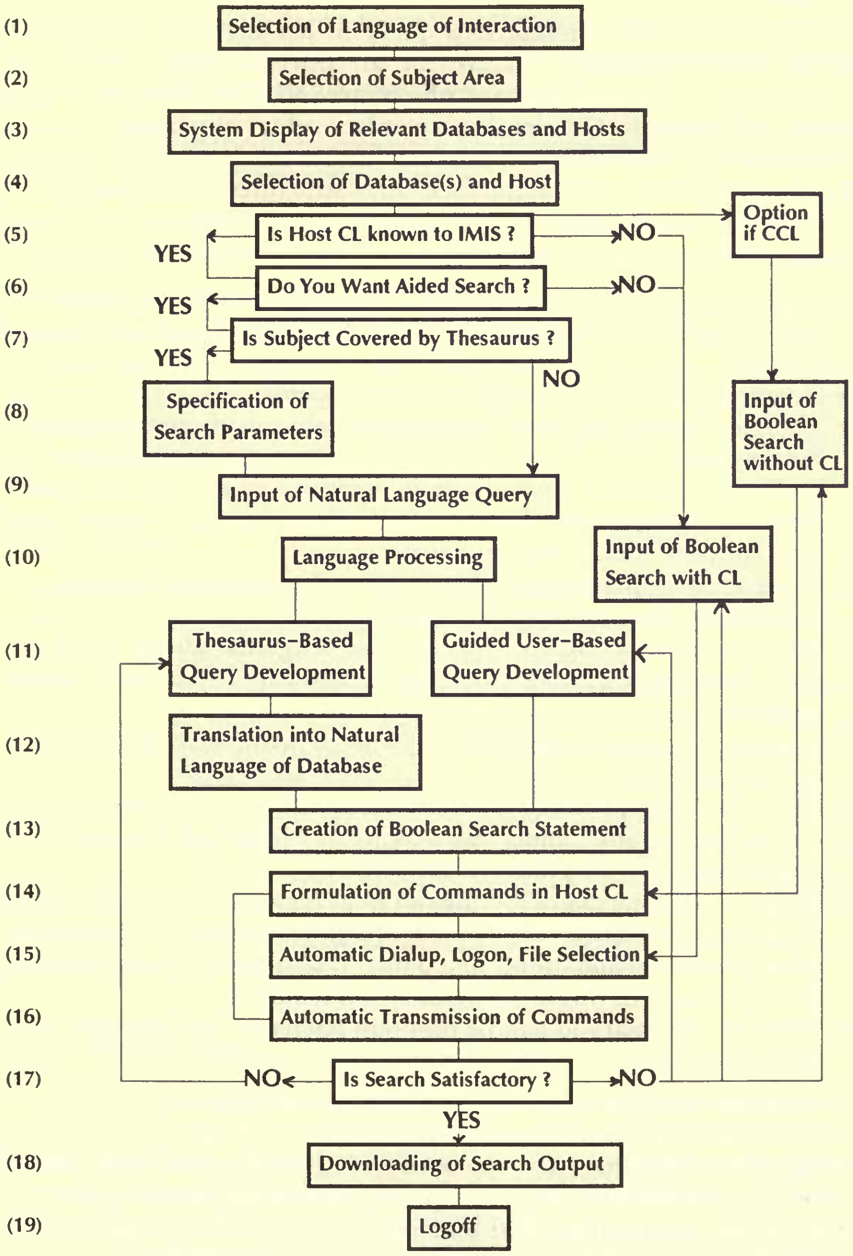
In "thesaurus-based query development" the user is asked to specify various "search parameters," e.g., Is the search to be by author or subject? Should it be precise? Should it be limited by date, language, or document type? What output format is required? How many output items should the search aim to produce? He/she then inputs a natural language query. Automatic language processing includes:

- separating the input into words;
- checking each word against an extensive stoplist;
- stemming each word not stopped;
- checking each stem against a dictionary (because stems with more than one meaning occur more than once in the dictionary, and for each stem there is recorded: a semantic category, a pointer to a position in a subject classification, and a pointer to any synonyms);
- clarifying the meaning of any stems not in the dictionary by interaction with the user (checking the spelling, assigning the word to a semantic category, and locating its position in the subject classification);
- checking successive stems in the input to see if they occur as compound phrases in the dictionary;
- disambiguating any remaining multimeaning stems;
- forming new compounds (not in the dictionary) from remaining successive stems in the input using rules on permissible combinations of semantic categories; and
- recognizing indicators of negation in the input.

Since the user is able to interact with the system in any one of four languages, these language-processing facilities have to be provided in each of those languages. If the search statement has been constructed in a language different from that of the chosen database, automatic procedures to translate search terms between English, French, German, and Spanish will be provided. A Boolean search statement in the language of the database will then be automatically constructed and transmitted to the host using the appropriate command language, and the search results will be automatically downloaded. If the initial search is not satisfactory, the system will return to an earlier stage to use thesaurus assistance in reformulating the search.

PROBLEMS OF INTERFACE CONSTRUCTION

The IMIS team is just completing its detailed design document, but we would be far from claiming that the problems of interface design have been resolved. Let us consider some of them.


(1)
Selection of Language of Interaction
(2)
Selection of Subject Area
(3)
System Display of Relevant Databases and Hosts
(4)
Selection of Database(s) and Host
Option if CCL
(5)
Is Host CL known to IMIS ?
YES
NO
(6)
Do You Want Aided Search ?
YES
NO
(7)
Is Subject Covered by Thesaurus ?
YES
NO
(8)
Specification of Search Parameters
Input of Boolean Search without CL
(9)
Input of Natural Language Query
(10)
Language Processing
Input of Boolean Search with CL
(11)
Thesaurus–Based Query Development
Guided User–Based Query Development
(12)
Translation into Natural Language of Database
(13)
Creation of Boolean Search Statement
(14)
Formulation of Commands in Host CL
(15)
Automatic Dialup, Logon, File Selection
(16)
Automatic Transmission of Commands
(17)
NO
Is Search Satisfactory ?
NO
YES
(18)
Downloading of Search Output
(19)
Logoff

IMIS in action

Analysis of Natural Language Query Input

To allow a user to express a query in her/his own words seems to be a very necessary feature for an interface that aims to be "intelligent." Natural language processing (NLP) is itself an art still under development. Queries display only a subset of natural language structures (e.g., they rarely contain verbs) and so are simpler syntactically. But analysis has to transform them into structures that represent the semantics of search statements, and this involves problems not always handled by conventional NLP. Here are a few of the issues:

- At the simplest level, how should misspelled words be recognized, and, if recognized, how handled? Should the system attempt to correct spelling?
- How will the system handle hyphenated words?
- There are unresolved problems in the recognition of compound terms. For example, how can we avoid forming a noun combination "cat food" in a sentence such as "It is necessary to give the cat food?" How is a long noun phrase to be broken up, e.g., "airport long term car park vehicle pickup point?"
- There are stock phrases and idioms such as "other things being equal"; how should they be recognized and handled?
- In some subject domains, there are also specialist phrase structures, e.g., dates such as "Monday March 24, 1989" that need special treatment.
- In general, how are numerals to be dealt with, such as "24 volts" or "Boeing 747?"
- How should enumerations be handled, e.g., "smog pollution control" (is "smog control" under discussion) or "hard and floppy disks" (should we form the phrase "hard disks")?
- Ellipsis is the practice of leaving out some data in a text string because it can be inferred from the context. A simple example is "the melting point of sulphur and the boiling point" ("of sulphur" is not explicitly stated). How will this be recognized and handled?
- Will it be necessary to handle pronoun reference? For example, what does "their" refer to in the following expressions?

  —the colors of dyestuffs and their chemical structures
  —the colors of dyestuffs and their fading
- Will the system have to cope both with grammatically well-formed sentences and with sentence fragments or ungrammatical input?
- Will the input make consistent use of capitals (for proper names, acronyms, etc.) so that they can be used to aid analysis?
- If a stoplist is used, how will the system handle a homonym such as the stopped word "and" and the expression "AND logic?"
- If a Boolean search statement is to be created, how will the system

know when to link search terms by AND, when by OR, when by AND NOT? For example, how should a query such as "Comparison of statistical and linguistic methods of indexing and abstracting" be represented as a search statement?

It is very difficult to construct a robust language processor that sensibly handles all types of user query.

Lexicons

By "lexicon" is meant *any kind of word file held within the interface system.* IMIS will contain monolingual dictionaries in four languages, pointers between language pairs, and pointers between words with thesaural associations (synonyms, broader and narrower terms). The creation of large lexicons presents many intellectual problems and is very labor intensive.

Subject Scope

Much experimental work on interfaces to information systems has been carried out within narrow subject limits. This clearly also limits the range of application of an interface and hence the number of potential users. For an interface to be commercially viable, it will have to handle a wide subject scope. This immediately increases the problems presented by lexicons; in particular, the problem of ambiguity—words with multiple meanings. There are few standard ways of resolving an ambiguity. A specific rule for each particular word must be constructed in most cases, and such rules are rarely foolproof. In a system of wide scope, one is no longer working with a subject domain that has a clearly defined semantic structure. It becomes more difficult for an interface to transform a query into a unique semantic representation that can be used in a search statement.

Query Modification

This includes both the process of clarifying and adjusting a query before search, and the process of revising a query if first search results are not satisfactory. There are two aspects to query modification. First, what ways of amending a query are open to intelligent interface? Second, what is the best balance between man and machine, i.e., should the interface make modifications automatically or should it simply advise the user as to what modifications are possible and leave him/her to take action? Also, an interface cannot make a general recasting of a query. It can only operate in small, discrete steps such as:

- adding a term to a query (using AND, OR, AND NOT),
- removing a term from a query,
- replacing one term with another,
- altering a Boolean operator (e.g., AND to OR),
- altering a term (e.g., by truncation), or
- altering a search limitation (e.g., field, date, language).

Making or suggesting changes a step at a time can irritate the searcher. The sequence of changes offered may not be acceptable. The procedures cannot easily cope with the user who suddenly has an insight into the query she or he should have put. The problem is learning how to provide guidance while retaining flexibility.

The degree of user initiative offered will reflect the views of the interface designer on how capable the user is of making search decisions. The ERLI/MINITEL system described earlier was explicitly designed on the assumption that the bulk of users could not make effective use of the subject headings of the French Yellow Pages. The EURISKO systems expects the user to be able to supply the terminology of the chosen subject, offering guidance only as to the kind of actions that may profitably be undertaken. IMIS plans to offer a variety of alternative degrees of user involvement.

Interface and Database

The interfaces described in this paper have all been situated with the user, and the software incorporated in a microcomputer. In this situation, there arise the issues: How much search preparation can the interface provide before going online? Can the interface continue to provide search aid when the user is already connected with the database?

Alternative ways of providing intelligent search aid are:

1. to mount an interface on a gateway node in the telecommunications network, accessible from each user's terminal: the DISNET project mentioned earlier will be exploring this possibility;
2. to mount it within a host computer: the European Commission is funding work on its own host, ECHO, using this configuration; or
3. to mount it in a microcomputer that also contains software to search local CD-ROM. This is a configuration that needs to be actively explored, especially as the possibility then arises of the interface software making active use of the indexes and thesauri stored on the CD-ROM.

CONCLUSION

This paper has tried to present some of the achievements, possibilities, and problems of constructing intelligent interfaces to online databases, arising out of European experience. Despite the effort that has gone into—and is continuing to go into—the development of practical systems, there is still a feeling in Europe that more analysis of the problems and experimentation with possible solutions are needed. This feeling is reflected in the existence of another European Commission project in which Tome Associates is involved, a project known as SAINT: Simplification of Access to Information using Normalised Transfer. The project is designed to collect further information on interface design, to come up with a more refined modular architecture, and to suggest experiments for the testing of particular modules.

REFERENCES

Barthes, C., & Glize, P. (1988). A case study of planning in information retrieval. *Expert Systems for Information Management, 1*(1), 50-65.

Clemencin, G. (1988). Querying the French Yellow Pages: Natural language access to the directory. *Information Processing & Management, 24*(6), 633-649.

Cognitec. (1988). *Simplification of access to information using normalised transfer: State-of-the-art analysis.* Report presented to the Commission of the European Communities.

Halpern, J.; Sargeant, H. A.; & IMA Centre de Documentation de l'INSERM, France. (1988). A new end-user interface for bilingual searching of MEDLINE. In *Online Information 88: 12th International Online Information Meeting* (pp. 427-443). Oxford: Learned Information.

Robertson, S. E., & Thompson, C. L. (1987). *An operational evaluation of weighting, ranking and relevance feedback via a front-end system.* London: City University, Department of Information Science.

Vickery, A. (1988). The experience of building expert search systems. In *Online Information 88: 12th International Online Information Meeting* (pp. 301-313). Oxford: Learned Information.

Vickery, B. C. (1989). *Intelligent interfaces for user-friendly access to databases: State-of-the-art survey.* Report presented to the Commission of the European Communities.

Vickery, B. C., & Vickery, A. (1990). Intelligence and information systems. *Journal of Information Science: Principles & Practice, 16*(1), 65-70.