Proteus: a concept browsing interface towards conventional Information Retrieval Systems

O. Signore - A.M. Garibaldi - M. Greco

CNUCE - Institute of CNR - via S. Maria, 36, 56126 Pisa (Italy)
Phone: +39 (50) 593201 - E.mail: oreste@ICNUCEVM.CNUCE.CNR.IT

ABSTRACT

In the access to the unstructured information, the representation of the relevant concepts is a fundamental issue for the sharing of the knowledge between indexers and users. The browsing of the concepts can play an important role in the user interface. We present an implementation of an interface that will give the possibility of interacting with a conceptual structure of the documents, and of making a graphical navigation on the thesauri available on the different fields.

INTRODUCTION

The growth and the spread of the online information retrieval services has been much less impressive than what was expected at the beginning of the '80s. This has been due in part to technological problems (reliability and speed of networks, logon procedures, etc.), and in part to intrinsic limitations of the Information Retrieval Systems.
As a matter of fact, the commercially available systems are not very different from the functional point of view, even if we can notice major or minor differences as far as the query language is concerned. This peculiar aspect has been considered many years ago by the European Community, who proposed ([8], [9]) a Common Command Language(or CCL) that was really conceived as a superset of the most used query languages. This language was successfully implemented by several hosts ([1], [13]) and became afterwards the basis for the ISO standard.

The Information Retrieval Systems ([11], [12], [19]) manage documents which have a flat structure, being subdivided into smaller units, named paragraph or fields. Therefore, when the user has to retrieve documents that contain some specific concepts, he/she has to search for the relevant concept into the document considered as a whole, or he/she can issue a more sharp query looking at specific sections of the document, so reducing the "noise" effect. It is evident that in this approach the user must be well aware of the content and the structure of the documents ([3]). To be more specific, this means that the user has to know the names and the semantics of the different fields. Even more complex is the case that occurs when, for some peculiar reasons, the information contained in a specific field has been coded (i.e. abbreviations, splitting into several subfields, etc.).
An *online help* is normally available, but we cannot affirm that this is satisfactory for a non skilled user.
These considerations make evident that it would be very effective for the user to have access to the documents using a "conceptual representation" of the document itself (see [2], even if the peculiar implementation is somewhat simpler than the proposed model), following in some way the paradigm used in the database environment, where we can define a conceptual schema and a set of operations acting on this schema.

Going back to the features of the Information Retrieval systems, we have to remember that the Precision and Recall factors are strongly influenced by the availability of lists of terms organised according to their semantic relationships of hierarchy, synonymy and preference, that is, by the availability of structured thesauri, which can be of invaluable help in the identification of the relevant concepts. In passing, we have to remember that the thesauri may be considered the representation of the knowledge of a specific domain . It is customary to use a hierarchical representation of the concepts, as, according to [17], moving on a hierarchical representation of the concepts is desirable *"because it seems to replicate the structure of human thought processes most closely, thus allowing the simplest, most direct transfer of the man's problem into the structure and the vocabulary of the system"*. Term relators (NT, BT, LT, USE, etc.) have been standardized at

international level [6], and commercial systems give the user the possibility of issuing more specific or more generic queries, using the available thesauri. However, when the thesaurus is constituted by thousands of terms, the mechanisms provided by the commercial systems may confuse the user, instead of helping him/her. This is due to the fact that all these mechanisms suppose that the user has a good knowledge of the specific domain and that if the user followed a different path, it would be difficult for him/her to restart the navigation on the concepts.
Finally, as it was pointed out in [16], some users of the information retrieval systems often need support to define or refine the topics which interest them, others should improve their knowledge of the subject, in order to decide exactly which are the interesting elements, others have a clear idea about the concepts to be found out, but they are not able to express such concepts. Others, at last, start from clearly defined subjects, but they realize that the starting concept is either too much indefinite (getting therefore too many documents), or too much specific (causing the finding of a too much scanty number of documents). In any case, a refining of the required concept is necessary.

The importance of a suitable user interface is an aspect that has been emphasized in the last years by many authors ([4], [18]). As a matter of fact, even the most sophisticated systems give services which depend on the effectiveness of the user/system concept communication, that is, the ability by which the user can identify the descriptors which allow to compare the user's query with the content of the documents.
An approach based upon knowledge assures a consistency in the representation between indexers and users, consequently the consistency can be obtained without using automatic indexing mechanisms, provided that a semantic system of concepts accessible to indexers and users is available ([7]). As a consequence, it is possible to implement intelligent interfaces which allow the user to make significant searches using suitable descriptors, by creating a mechanism of communication that fits the key concepts.
One of the most promising approaches is consituted by the graphic browsing interfaces, where the browsing represents an informal or euristic research that, through a collection of well connected documents, allows to find the important information which is needed. One of the tasks that a browsing interface has to fulfill, is to allow the users, particularly when they start from not well defined queries, to prepare or to refine their necessities([15]), giving the possible alternatives of research. The fundamental condition to implement a browsing interface is the availability of a high quality thesaurus.
These considerations led to the decision of implementing a general purpose interface, which has been named Proteus because it is adaptable to any information retrieval system.

THE PROTEUS INTERFACE

As it has been pointed out before, the idea of implementing the Proteus interface came from the consideration that, as noted in [18], while a lot of effort has been put in the area of the representation of documents, automatic indexing, probabilistic models, etc., other aspects, like the implementation of tools for the support of the query formulation, have been of little concern.
The basic idea is to prepare the query in a sort of "esperanto" (using, in reality, the ISO standard syntax), and then to translate it in the query language of the system available on the host to which the user is connected. A possible solution appeared to be the splitting of the interface and query formulation support functions from the data bank query phase.
The architecture of Proteus is depicted in fig. 1. It is evident that the two basic functions (graphic interface and submission of the query to the host) are seen and managed as two separate tasks.

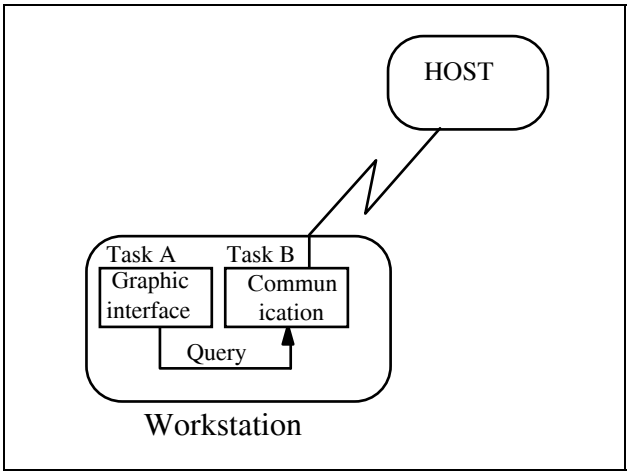

Fig. 1 - The PROTEUS Architecture

The data coming from the host site are not currently intercepted, leaving therefore unchanged the communication protocol and the data formatting provided by the host itself.
The interface has been conceived as being adaptable to every Information Retrieval System and:
- it is available on widespread personal computers;
- it permits the user/system interaction via an abstract representation of the document;
- it implements a graphic concept browsing.

The interaction via an abstract representation of the documents offers some advantages, namely:

- the independence from the name constraints imposed by the IRS;
- the possibility of interacting having the perception of the content of the fields according to an "external syntax" ([14]);
- a query formulation according to a uniform language, independent on the specific system, and therefore the implementation of a simple and consistent paradigm of interaction.

In the implemented solution, the browsing is used as a support to the query formulation, and not as a document search mechanism.
This choice postulates the availability of high quality classification schemas and/or thesauri, and it is very similar to the approach adopted by CALIBAN ([5]), CoalSORT ([7]) and CANSEARCH ([10]).

The supporting database

The supporting database is formed by three groups of tables, which contain the description of the hosts and of the available databases, the query tokens and the information needed for the navigation on the thesaurus.
As far as the description of the hosts and databases is concerned, it is worthwhile to note that, in case we have different systems, possibly on different hosts, managing the same (as far as the semantics of the various fields is concerned) database, the characteristics of the fields are stored once.
In more detail, a single field can be of four different types:

Thes when a thesaurus is available, that is, when the field values are semantically arranged according to their preference, equivalence or hierarchical relationships;

Dict when the field can take values from a definite list of values;

Text when the field is a free text field;

Cod when the field takes values that are a coded representation of the perceived values. This is the case when a single field is internally splitted into a set of fields, but is perceived by the user as a set of fields ([14]).

In the query formulation phase, some tables are used in order to store the tokens which constitute the query (fig. 2), or the acceptable values for Dict fields. In this case, the user may get a list of the acceptable values, and make a selection, possibly multiple, of the required values. It is needless to say that the access to the dictionary is not mandatory, and all the query values are in any case checked against the dictionary, avoiding, in this way, the submission of useless queries.

The thesaurus modelling tables give account of the semantic relationships between the terms. In the pilot test case, the structure has been tested against two large thesauri, consituted by several thousands of terms, namely ICONCLASS and GEODOC.
ICONCLASS (Iconographic Classification System) is a classification system set up by the Leiden University (Holland) between 1944 and 1984, and it is used for the description of the iconography of the western art. A typical example of a ICONCLASS hierarchy is in fig. 3. Keywords are attached to the different subjects, and the code supplies a good linguistic independence.
GEODOC is a multilingual thesaurus agreed by the Commission on Geological Documentation (COGEODOC) of the IUGS (International Union of Geological Sciences) as a common terminological thesaurus to be adopted by different documentation centres.

Field	Value
Author	Michelangelo OR Caravaggio
Subject	'11 A' OR '11 D 2'

Fig. 2 - The query tokens table for the query:
(Author : Michelangelo OR Caravaggio) AND
(Subject : '11 A' OR '11 D 2')

25	earth
25 H	landscapes
25 H 2	landscapes with water, waterscapes
25 H 21	water course
25 H 21 3	river personified, if wanted with NAME between brackets

Fig. 3 - A typical ICONCLASS hierarchy

The software architecture

The application is made up by two different programs (A and B) whose executing instances are the process A and the process B.
First of all, the application invokes the A process. This process, in turn, may invoke the B process (the "child" process) and *both can continue in parallel* (fig. 4).
The B process is an instance of the program that implements the Thesaurus manager.
For each thesaurus a thesaurus manager has been coded, but they have the same basic structure.
Multitasking has been implemented both at the process level and at the thread level. In fact, each process is made by two execution threads:

- a ***first thread*** which is in charge of receiving the user input and of passing the time consuming tasks to the second thread;
- a ***second thread*** which is in charge of executing the time consuming tasks (as a series of queries to the support database) avoiding locking the system waiting for the completion of them.

Processes are syncronized by semaphores, threads are syncronized by an exchange of messages.

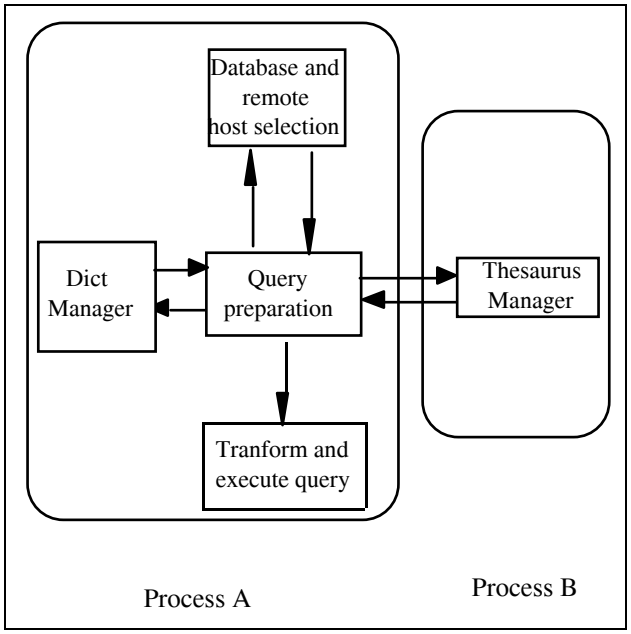

Fig. 4 - The processes architecture

Proteus at work

We can distinguish three groups of functionalities, each for a well defined phase of the interaction with the database.
The *first group* contains the functionalities that permit the user to select the remote host and the database he/she is interested in. The *second group* is necessary for the formulation of the query. In this phase the user may use the Dict_Mgr and the Thesaurus Manager, which, in fact, constitutes the functionalities of the *third group*.
The first and the second group are absolutely general, while the thesaurus management presents implementation aspects that depend on the organization of the thesaurus itself.
As it can be seen from Fig. 4, the interaction wiIl take place in three different phases, and the navigation on the thesaurus will take place only if the user intends to refine his/her query. Otherwise, if the user is certain of the terms to use, he/she can enter directly the terms, that will simply be checked in the thesaurus to make sure of their existence.
It is worthwhile to note that *the interaction with the Thesaurus manager can take place in parallel with the phase of the setting up of the query on the other fields of the document.*

Let us now have a quick look at the way in which the interface works.

As soon as the user has access to one of the databases he/she is registered to, he is faced with a conceptual representation of the document (fig. 5), independent from the features of the specific IRS.
It is needless to say that he/she can get a description of the content and the meaning of the various fields, or he/she can enter a value that will be saved for the query formulation (fig. 6)
If the user wants to enter a value for a controlled dictionary field, he can get a list of allowed terms, and perform a single or multiple choice in a list box, or just enter values that will be checked against the dictionary.

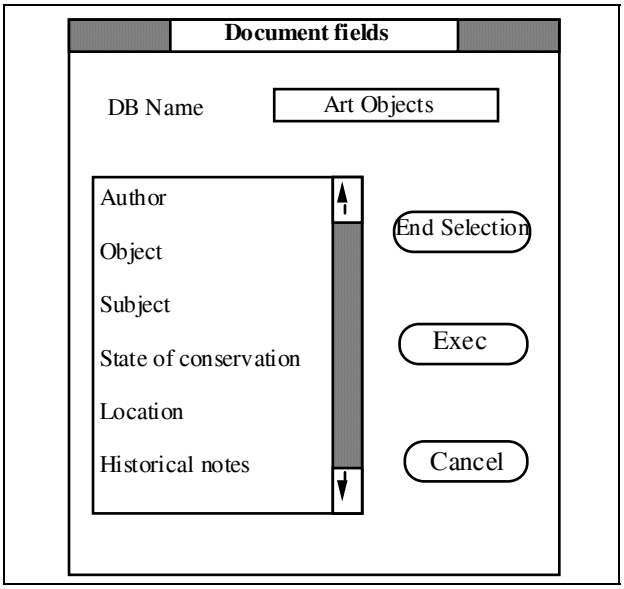

Fig. 5 . The conceptual document

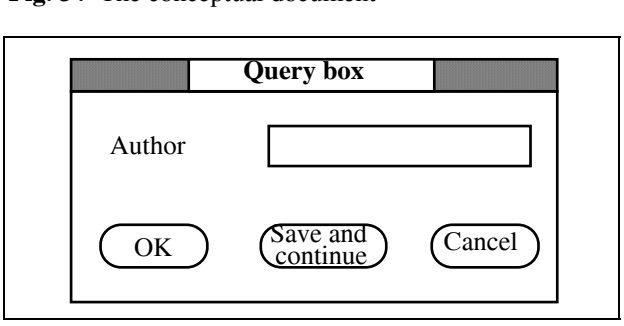

Fig. 6 - The query box for the field *Author*

The Thesaurus Manager

As it has already been pointed out, the Thesaurus Manager exhibits a different behaviour depending on the specific thesaurus, even if the basic functionalities are the same, that is:

- graphical representation of the thesaurus as a graph, where the nodes are the thesaurus terms, and the arcs are the connecting relationships;

• "point and click" interaction style.
Once the user has identified the kind of relationships he/she is interested in, he/she can select a term, either directly, or from a list box, and will have the display of a neighbour of the term, where only the selected relationships are shown (fig. 8).
The user can then use the various options available from the menu bar, or he/she can set some default actions to be taken as consequence of a "double click".
He/she can move around the structure (and ask for detailed explanation of the terms) using the scroll bar, and can extend the tree towards the root or towards the leaves.

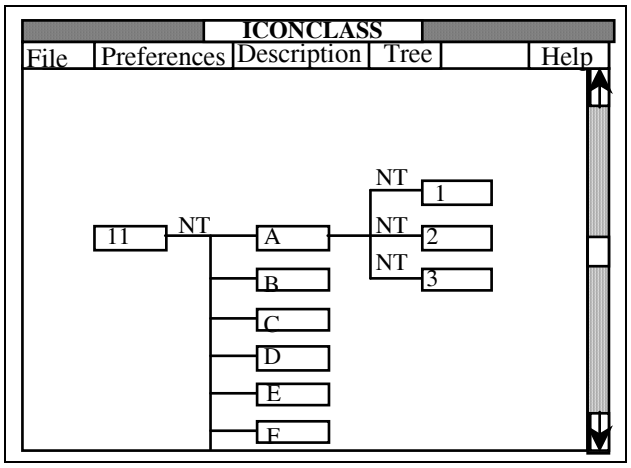

Fig. 8 - The neighbouring of the initial code '11 A'

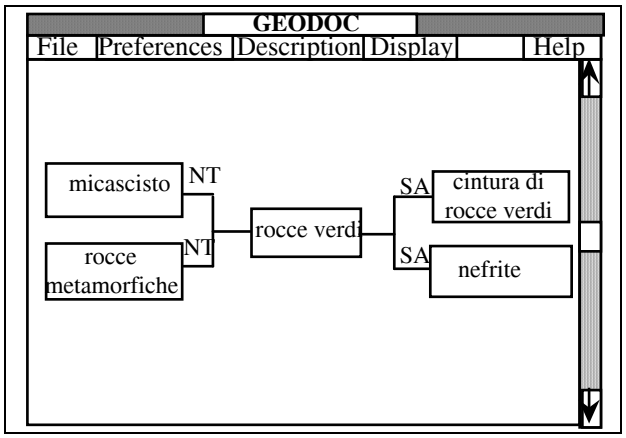

Fig. 9 - A "butterfly display" for the GEODOC thesaurus (italian version)

Any term can be selected in order to build a query to the database: all the terms which are selected in a single navigation will be OR'ed.
The colours emphasize the role of the nodes, and make possible to distinguish between the currently active node (*red*), the navigation starting node (*dark cyan*) and all the others (*green*).

The user can move around the displayed structure. As soon as he/she clicks on a node, the node will become active. Depending on the selected profile, which can be modified during the navigation, a double click will produce different effects.
When the "extension" profile is selected, a double click on a terminal node, either a leaf or the root, will cause an extension of the tree respectively towards the leaves or towards the root. The descritpion of the content of the active node can be obtained selecting the appropriate option from the menu bar.
Selecting the "description" profile will produce a symmetric effect: an extension by activation and menu, or a description by a double click.

Multi-tree thesauri are displayed as a "butterfly display" (fig. 9) where the central level is always constituted by a single node (at the beginning this is the navigation starting point). On the right we will find all the terms that constitute the domain of the central term in respect to the selected relators. On the left we will find all the terms that constitute the codomain of the central term in respect to the selected relators. Again, the arcs can be labelled with the names of the relators. All the operations previously described are possible.

CONCLUSIONS AND FUTURE WORK

As far as the implementaion environment is concerned (OS/2 and Presentation Manager) we have to notice that the used tools appeared to be quite complex and not completely reliable (but we were working with initial releases).
We consider that the aims have been achieved. More exactly:

- we succeded in the implementation of a tool that makes easier to formulate queries on a document, using a conceptual structure of the document itself. In addition, the user has the values which are specified in the query automatically tested against the available dictionaries and thesauri.
- The thesaurus manager gives a snapshot of the structure of the thesaurus in a neighbour of the term of interest, and the user is allowed to navigate in the structure. The navigation on the thesaurus may be accomplished in parallel with the formulation of the query on other fields of the document.
- The interface is totally independent from the databases, only the thesaurus manager has an intrinsic dependence on the thesaurus structure.
- The interface agrees with the SAA standards ([20]), and uses the languages ("C") and services (SQL languages, PM presentation interface, etc.) that are part of the CPI. In addition Proteus has an external aspect wich conforms to the CUA.

While the novice user can receive help in the formulation of the query, the expert user can use some shortcuts in order to speed up the process.
Some extensions are presently under study or development.
At present, the formulated query may be edited, but the only possibility given to the user is the modification of the boolean connectors between the tokens. However, the modification of the single token can be easily implemented.
The connection with the remote host will be implemented according to the CPI standards, using APPC (Application Program to Program Communication). The query will be saved for future use (establishment of a user profile)
In principle, it is possible to implement the multitasking navigation on several thesauri, using appropriate semaphores, but this enhancement is under consideration, because it seems not to be really useful for the user, who should follow several classification schemas in parallel.
We are considering the possibility of giving the user the ability to use the Thesaurus manager as a tool for the manipulation of structured thesauri (addition of terms, translation, moving of a set of terms, establishment of new relationships, etc.)
Finally, from the software implementation point of view, we are considering the possibility of having a parametric representation of the thesaurus structure, in order to implement a single thesaurus manager, which will produce a display that will vary according to the structure of the thesaurus it is processing.

References

[1] Bartoli R., Romano G.A., Signore O.: Implementation of Common Command Language on STAIRS/VS-TLS, in Deontic Logic, Computational Linguistics and Legal Information Systems (A.A. Martino Ed.), North Holland (1982)

[2] Bertino E., Rabitti F., Gibbs S.: *Query processing in a multimedia document system*, ACM Transactions on Office Information Systems, Vol. 6, N. 1; pp 1-41 (1988)

[3] Blair D.C., Maron M.E.: *An Evaluation of Retrieval Effectiveness for a Full-Text Document Retrieval System*, Comm. of ACM, Vol. 28, N. 3 (March 1985), pp. 289-299

[**4**] Croft W.B.: *Advanced Information retrieval techniques*, HYPERTEXT'89 Course Notes, Pittsburgh, 1989.

[5] Frei H.P., Jauslin J.F.: *Graphical presentation of of information and services:a user-oriented interface*, Information Technology: Research and Development, N. 2, pp.23-42

[6] International Standard ISO 2788, *Documentation Guidelines for the establishement and development of monolingual thesauri,* International Organization for Standardization, Switzerland (1986)

[7] Monarch I, Carbonell J.: *CoalSORT: A Knowledge-Based Interface*, IEEE Expert (Spring 1987), pp.39-53

[8] Negus A.E.: *Euronet Guideline: Standard Commands for Retrieval Systems*, The Institution of Electrical Engineers, London (1977)

[9] Negus A.E., Snowden A.E.: *EURONET-DIANE User's Guide (Common Command Language),* Scicon Consultancy International Ltd. (1980)

[10] Pollitt A.S.: *End user touch searching for cancer therapy literature-a rule based approach*, in *Proceedings of the Sixth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Vol.17, N.4, (June 1983), pp.136-145

[**11**] Salton G., McGill M.J.: *Introduction to modern Information Retrieval*, McGraw-Hill, New York (1983)

[**12**] Salton G.**:** *Automatic text processing*, Addison-Wesley (1989), ISBN 0-201-12227-8

[13] Schreiber F.A., Difilippo C., Zagolin M.: *Un preprocessor per l' interfacciamento di sistemi eterogenei di information retrieval su una rete di calcolatori,* Proceedings del convegno annuale AICA '80 (29-31 Ottobre 1980), pp.1245-1263

[**14**] Signore O., Bartoli R. : *Implementation of a historical-geographical database with support of imprecise dates*, DEXA 90: Proceedings of the International Conference in Vienna, Austria, 1990 (Tjoa A.M., Wagner R., Eds.), Springer Verlag, Wien-New York

[15] Signore O.,Aulisi R., Ceccanti V.: *Hypertext for Hypertext: A Figured Thesaurus*, DEXA 91: Proceedings of the International Conference in Berlin, Germany, 1991 (D. Karagiannis., Ed.), Springer Verlag, Wien-New York, pp. 514-519

[**16**] Smith P.J., Shute S.J., Galdes D., Chignell M.H.: *Knowledge-Based Search Tactics for an Intelligent Intermediary System*, ACM TOIS, Vol. 7, N. 3 (July 1989)

[17] Thompson D.: *Interface design for an interactive information retrieval system: A literature survey and a research system description*, J. Am. Soc. Inf. Sci. (1971), pp. 361-373

[18] Thompson R.H., Croft W.B., *Support for browsing in an intelligent text retrieval system,* Int. J. Man-Machine Studies, Vol.30, 639-668,1989.

[**19**] Van Rijsbergen C.J.: *Information retrieval*, Second edition, Butterwoths, London(1979)

[**20**] Wheeler E.F., Ganek A.G.: *Introduction to System Application Architecture*, IBM Systems Journal, Vol. 27, No.3, 1988