A Software Base For Environmental Studies

G. GUARISO[1]* AND H. WERTHNER[2]

[1] *Centro Teoria dei Sistemi CNR, Politecnico di Milano, Milano 20133, Italy*

[2] *Institut für Statistik und Informatik, Universität Wien, Austria*

A prototype of an on-line inventory of computer programs for environmental studies available at the Laboratorio di Informatica Territoriale ed Ambientale (LITA), Politecnico di Milano, is presented. The main purpose of this software base is to overcome some of the difficulties encountered in disseminating to a relatively wide range of potential users all the informations concerning programs already available. Each program description is stored in a structured, but easy to read format in order to allow a fast screening. The prototype software base has been implemented on a personal computer, using a commercial relational DBMS. Furthermore, a user-friendly interface, which has some intelligent features, has been designed in order to ease the use of the system.

Received October 1986, revised April 1987

1. INTRODUCTION

Teaching and research activities in environmental sciences are increasingly relying on the availability of computers and adequate software and thus many educational institutions, training and research centres dealing with this domain are now equipped with various hardware and software facilities. However, only in a very limited number of cases (see for instance Refs 1 and 2) has the software available in these institutions been catalogued in such a way that it may be easily found, understood and utilized by a number of people outside the group who developed and/or implemented the program. As a consequence, potential users do not know if any software is available for their particular problem or where it can be found or how it can be used.

In a university context, an additional problem arises: expertise in computer science and in the application domain varies widely among users. For instance, the Computer Laboratory for Environmental Sciences (Laboratorio di Informatica Territoriale e Ambientale, LITA), Politecnico di Milano, was created for three different purposes: teaching the most modern environmental planning and management techniques to university students (about 200 per year); improving the know-how of engineers working in regional and municipal authorities (50–100); and supporting the research activities of the Department of Electronics in this area (10–15 scientists).

With such a wide range of users, it is essential to allow an easy search and access to all the software developed in the laboratory or obtained by other sources. That is why the software base presented here has been developed. It supports the process of locating and choosing a software tool by describing the program, its limitation, data requirement, techniques and methods used, but does not guide during the use of a specific package as in ref. 3.

The basic requirements and purposes of the system are briefly discussed in Section 2. Section 3 describes the structure of the software base at LITA, while the fourth section presents the relevant characteristics of the user interface.

2. DESIGN OF THE SYSTEM

The design of an acceptable retrieval system has to solve the following two problems.

(*a*) Store a relatively large amount of non-numerical data, which describe the programs (about 200 at present).

(*b*) Support an easy access to this information by a user who does not know and does not have to know a formal query language.

These problems are often conflicting.[4] The information should be highly structured in order for the computer to access it easily, and should be flexible enough to allow an easy interaction with the non-computer-oriented user. The software description must be as accurate as possible so that the user gets at least some idea of its suitability for a specific application, but should be short enough to allow a fast screening process. In order to free the user from heavy formal restrictions and to guide him during each phase of the retrieval, a system component has to support the man–machine communication.

3. THE DATABASE STRUCTURE

The system developed at LITA is based on a relational database structure with the following characteristics.

(*a*) The information is stored in the form of tables (relations), where each row (tuple) represents one program and each column represents one of the program's specifications (attribute).

(*b*) Several relations are used to describe the programs, since each one represents a particular point of view according to which the program can be catalogued or examined.[5] This makes the form in which the program descriptions are stored in the software base relatively close to the user's more intuitive representation. Furthermore, there will be a main relation containing a number of keywords which give a condensed form of description of the program purposes and main characteristics. The screening process will work only on this last relation.

(*c*) Software retrieval takes place only through the use of a series of keywords identifying the program's main features.

(*d*) The software base is implemented on a PC system of the LITA LAN in order to be accessible to a larger number of users and thus has to take into account some hardware restrictions.

The single relations are the following.

* To whom correspondence should be addressed.

PROGRAM-KEYS

This relation contains in the form of keywords the basic information which describes a program in a condensed form and shows how to locate it.

PROGRAM	Program name and version; primary key
DATE	Date of completion of the program
KEY	General area of application of the program (e.g. ecology, water quality, air pollution ...)
OBJ	Keys for program objectives (management, forecasting ...)
TYPE	Keys for technique used
STUDY	Indicates if the program is isolated or part of a larger package (in this case related programs are quoted)
DOC	Level of documentation (minimal, detailed, ...)
DOC_LANG	Documentation language
COMPUTER	Type of computer
LANGUAGE	Computer language
SHORT_DES	Short description of the program (normally two lines long) used for screening purposes

PROGRAM-TECHNIQUE

Here the theoretical and technical basis of the program are stored in a plain form (normally 2–3 lines of text for each attribute in the relation).

PROGRAM	
OBJECTIVE	Summary of program objectives
TECHNIQUE	Summary of the theory and techniques on which the program is based
PROG_STRUCT	Definition of the structure of the program implementation
DATA	Necessary data
OTHER_PROG	Explains how the program can be connected with other programs (if any)
LIMITS	Conceptual limits of the program
APPL_CASE	Short reference to past cases on which the program has been used

PROGRAM-COMPUTER

Computer aspects are included, mainly using numerical description of hardware requirements.

PROGRAM	
COMP	Name and type of the computer
COMP_WORD	Length in bits of the computer word
COMP_CORE	Dimension of main memory (kbyte)
COMP_STORE	Dimension of mass storage (kbyte)
CODE_LINE	Length of source code
ADD_PACK	Specifies if any computer routine or library is necessary to run the program
SAMPL_APPL	Description of sample application where the following data have been evaluated
CORE_OCCUP	core occupation of the sample
RUNTIME	time required to run the sample

PROGRAM-LINKS

It refers to all persons and institutions connected with the program, and contains some indication of transfer and training estimates to implement new applications with the program.

PROGRAM	
DEVELOPER	Name of the original developer
DEV_AFFIL	Affiliation of the developer
DEV_ADDR	Address of developer
CONTACT	Name and address of person to contact for further information (if different from developer)
USER	Name of actual users
SUPPORT	Address of the supporting institution, if support exists outside the laboratory
TRANSF_EST	Estimation of transfer work for a new application
TRAIN_EST	Estimation of training time

PROGRAM-DOCUMENTATION

Descriptive texts are memorized in this relation. These are normally files of up to two typewritten pages on which no search is allowed.

PROGRAM	
DOC_REF	References on program documentation
HELP	Abstract of the documentation with essential information on how to run the program

4. THE INTERFACE

The system interface supports some understanding of natural-language input, thus freeing the standard user from learning a strongly formatted DB query language. A deep understanding of natural language, that means understanding of complete sentences with a broad range of words, as proposed for example in ref. 6, is not supported by our approach. The interface implemented for the software base is limited to the handling of very simple English expressions, which describe some characteristics of the programs and are translated into the keywords used for the locating process. This approach is easy to fulfil in the DB structure outlined above, where only keywords play an essential role in the screening process. Another circumstance makes the design of the interface easier: it does not have to deal with the whole set of English words, but only with a relatively small subset of the language concerning software for environmental studies.

The user is allowed to type in his descriptive words, use some standard abbreviation, and make some spelling mistakes, which the machine detects. Another characteristic of the interface is the capability of learning from each new user.

The interface of the retrieval system (see Fig. 1) consists of: (*a*) a component, which stores a dictionary of

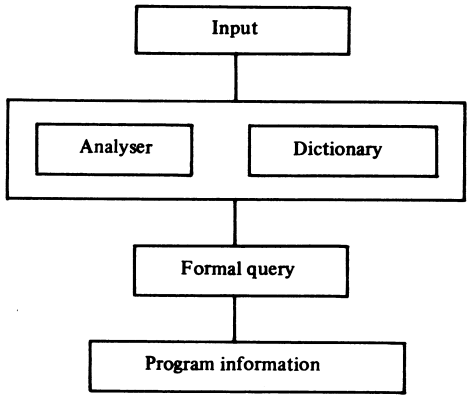

Figure 1. Structure of the interface.

usable words, here the keywords and their synonyms (see for instance refs 7 and 8), (*b*) an analysing mechanism to interpret the user input and to learn from a new input.

DICTIONARY is a relation where the database keywords are stored one to one with their synonyms. Each keyword may have any number of synonyms or abbreviations, and both the keywords and their synonyms are stored in a standardised form (i.e. singular form, without special characters). The ANALYSER works on this separate relation DICTIONARY in a heuristic manner by looking for a synonym which matches the expression typed by the user.

The input may be composed of one or a few words (normally nouns and adjectives) which are often used together to indicate a specific concept (e.g. water pollution, partial differential equations ...). First, it is standardised and partitioned, if it consists of more than one word. The whole input, in this standardised form, is then searched for among the values of the attribute SYNONYM in the relation DICTIONARY. If the input matches an existing value, the search ends. If not, the system takes the first letter of the input (by this assuming that the probability of a correct first letter is very high) and a measure of the input length as basic criteria of similarity and tries to match all synonyms which fit these criteria. The comparison of the input and the synonym in the matching process allows also the detection of spelling errors (with the length of two) in the input. If conformity to a specified percentage is reached, the keyword is found and the user, after a consistency check, is asked if he wants to insert his input as a new synonym into DICTIONARY.

If the search on the whole input fails and the input has been parted the process continues for the single parts as described above. Each part, when found, is transformed into the related keyword. After locating all parts, the new transformed input (synonyms changed with the associate keywords) is searched for in DICTIONARY with the same procedure. In case of failure of the search, a list of all available keywords is displayed.

Finally the keywords are transformed into a formal query for locating the programs in the relation PROGRAM-KEY.

5. CONCLUSION

A software base for environmental studies has been implemented at the 'Computer Laboratory for Environmental Sciences (LITA)', Politecnico di Milano. A structured, but relatively complete description of all the programs available at the laboratory is stored into the software base in such a way that the users (students, engineers, researchers) can see if software for their specific application is available. An easy interaction with the user is made possible by an intelligent interface, which does not understand natural language, but allows the use of simple techical expressions in the user's more natural language. A simple implementation of the dictionary and a heuristic approach in analysing the input shows satisfactory results.

The software base is presently working on a PC system and as such is available to all the users of the LITA local area network. In the near future, it is planned to transport the software base to a LISP-machine where the intelligent features of the interface can be strengthened and some application programs can be started directly from inside the software base.

Acknowledgement

This work has been partially supported by a study contract of IBM Corp.

REFERENCES

1. IAHR (International Association for Hydraulic Research), *Information Exchange on Computer Programs*, 2nd edn (July 1978).
2. C. R. Shriner (ed), *Inventory of Databases, Graphics Packages and Models in the Department of Energy Laboratories*. U.S. Dept. of Commerce, National Technical Information Service, Oak Ridge, Tennessee (1978).
3. R. W. Blanning, A system for NL communication between a decision program and its user. In *Artificial Intelligence Economics and Management* (ed. L. F. Pau) (1985).
4. D. L. Waltz, An English language question answering system for large databases. *Comm. ACM* **21**, (1978).
5. S. I. Gass, Computer Program Documentation: A Review and an Approach. NBS Publ. 500-39 U.S. Dept. of Commerce, 1979.
6. G. G. Hendrix, E. D. Sacerdoti, D. Sagalowicz and J. Slocum, Developing a natural language interface to complex data. *ACM Transaction on Database Systems* **3** (1979).
7. L. Bolc, K. Kochut, A. Kowalski and M. Kozlowska, Natural language retrieval system with extension toward fuzzy reasoning in medicine. *Proceedings of the 6th European Conference on AI* (ed. Tim O'Shea). Pisa, Italy (1984).
8. G. Guida and C. Tasso, An expert intermediary system for interactive document retrieval. *Automatica*, **19** (6), 759–766 (1983).