
N A S A

N A S A

THE MAN/MACHINE INTERFACE
IN
INFORMATION RETRIEVAL:
PROVIDING ACCESS
TO THE
CASUAL USER

Martin Granier

Computer Science Department
University of Southwestern Louisiana
P. O. Box 44330
Lafayette, Louisiana 70504

May 16, 1984

DEMS.NASA/RECON-4

- 1 -

MAN/MACHINE

ABSTRACT

This study is concerned with the difficulties encountered by casual users wishing to use Information Storage and Retrieval Systems. A casual user is defined as a professional who does not have the time nor desire to pursue in depth study of the numerous and varied retrieval systems. His needs for online searching are only occasional, and not limited to a particular system.

The paper takes a close look at the state of the art of research concerned with aiding casual users of Information Storage and Retrieval Systems. Current experiments such as CONIT, IIDA, CITE and CCL are presented and discussed. Comments and proposals are offered, specifically in the areas of training, learning and cost as experienced by the casual user. An extensive bibliography of recent works on the subject follows the text.

KEYWORDS

Man/Machine Interface, Information Storage and Retrieval Systems, Casual User, User/System Interaction, CAI, CAT, Assistance, Help, CCL, CITE, CONIT, IIDA, LEXIS, TRAIN.

-
TABLE OF CONTENTS

	<u>Page</u>
1. INTRODUCTION	6
2. STATING THE PROBLEM	6
2.1 The Intermediary	7
2.2 Problems and Challenges	8
2.3 The Casual User	10
2.4 "Leave the Casual User Alone"	11
3. EDUCATION	13
3.1 CAI/CAL	14
3.2 TRAINER	15
4. STATE OF THE ART	18
4.1 LEXIS	55
4.1.1 Presentation	19
4.1.2 Searching LEXIS	20
4.1.3 LEXIS Conclusion	22
4.2 CONIT	25
4.2.1 Presentation	26
4.2.2 Study of CONIT	27
4.2.3 Searching with CONIT	28
4.2.4 Comparing Languages	33
4.2.5 The Problems	35

4.2.6	CONIT Conclusion	37
4.3	IIDA	37
4.3.1	Presentation	37
4.3.2	Study of IIDA	38
4.3.3	Two modes available	38
4.3.4	Information Gathered	40
4.3.5	Monitoring User	42
4.3.6	Diagnoses	43
4.3.7	IIDA Conclusion	47
4.4	CITE	47
4.4.1	Presentation	48
4.4.2	Study of CITE	49
4.4.3	CITE Conclusion	52
4.5	CCL	52
4.5.1	Question and Standard	52
4.5.2	Presentation	53
4.5.3	Problem in Standard	54
5.	REMAINING QUESTIONS	55
5.1	New Technology	55
5.1.1	Micro Computers and Smart Terminals	56
5.1.2	Graphics Interface	59
5.1.3	Other Ways	60

5.2 Cost Factor 60
 5.2.1 Development Cost 61
 5.2.2 Usage Cost 61
5.3 Problems 62
6. SUMMARY. 63
REFERENCES 66

THE MAN/MACHINE INTERFACE

IN

INFORMATION RETRIEVAL:

PROVIDING ACCESS

TO THE

CASUAL USER

1. INTRODUCTION

This age we are living in has been labeled by many as the "Information Age." Since the 1960's, we have seen a tremendous growth in technology, which has made possible the creation of online retrieval systems. It came right on time, because, simultaneously, the quantity of documents and publications available has experienced an expansion never seen before. There would be no way for the old structures - libraries and published indices - to keep up with such a growth, if it were not for the development of technology. However, all is not beautiful for the user: indeed, data is available much faster and more complete, but so far it has been necessary for the user to retrieve information through an intermediary, such as a librarian or an information specialist. This paper will study what are the challenges to the untrained users of online retrieval systems, as well as the options for the future.

2. STATING THE PROBLEM

Online retrieval systems have been very well accepted by the user community, since their early debuts, in the late 60's. The tremendous growth in technology which has happened in the computer field, has been reflected in all areas of data retrieval; research in networking, reliability, memory size and speed, all had important and constructive consequences in the information retrieval area. During the same period the number of databases has multiplied, as well as the number of records in every one of them. By the end of the 1970's, the number of online searches per year, increased from 1 million to 2 million and since then, the growth has kept the same fantastic progression.

With the rapid proliferation of data available on line, and the general acceptance by the general user community of this way of searching, new problems have appeared. One of them is the difficulty for an occasional user to search a database effectively, without human help such as from a professional searcher or a librarian. In order to access commercial systems, the searcher must follow a rigid code: logging in procedures, manipulation language, error messages and help online differ widely from one system to another. All claim, to a certain extent, to be "user friendly", (or even better "ergonomic!"), but

that is seldom the opinion of the end-user!

2.1 The Intermediary

To remedy these difficulties, a quite general approach in business and university environments is for the person in need of information to go to a professional searcher, well trained for online searching, and to use him as an "interface"! Generally, it is thought that only between 10% and 15% of the searches being performed are performed by the end-users themselves [Wanger, 79].

However, for reasons which are explained in this report, it can be interesting for users to perform the searches by themselves, instead of through an intermediary, as the case normally is. The advantages of having a professional searcher working online are tremendous, there are no doubts about that; but some difficulties exist, such as communication between user and searcher, time frame, availability of each party, and other incompatibilities. Furthermore, the user could very well benefit from hand-on experience, and even retrieve information "accidentally", perhaps by discovering new keywords while online or modifying search strategies while online.

2.2 Problems and Challenges

It is also important for the reader to realize that the success and the growth of interactive bibliographic retrieval systems have been such that users, in recent years, have found themselves facing huge challenges. The difference between the systems available, their organizations, indices, thesauri, retrieval languages and procedures make occasional searching quite an enterprise. Some kind of standardization is clearly overdue, but, in the meantime, what should be done?

Recently, many efforts have been made to make online bibliographic retrieval systems easier to use by the end user. In fact, since the early days of computers, designers have attempted to befriend the users: from user training and online help, to the use of a "mouse," the promises are many. However, problems remain, and we will examine the ones users of online retrieval systems encounter.

In the retrieval systems, the problem is twofold:

-
- (1) On one hand, it is extremely difficult, if at all possible, for the end user, to perform searches well and efficiently, if he has not acquired some kind of practice, and even expertise in the system he wishes

-
to use.

- (2) On the other hand, how is he to acquire this experience if his searching needs are not only occasional, but also require the use of various systems?

Here is a perfect "Catch 22", which many have tried to circumvent. This paper will study the different approaches possible, in order to resolve the problems facing a casual user desiring to search information online by himself.

2.3 The Casual User

A user, let's say a scientist, whose searching needs are only occasional does not wish to spend long and repetitive sessions learning how to use a specific system. He could very well do it, as his intellectual faculties are not in question, but he does not have the time - nor the desire - to study query languages he will use only one or two times a year.

--
For the purpose of this paper, a person whose knowledge in online retrieval is limited, and whose extent of experience in this form of search is only minimal and occasional, will be defined as a "casual user". The casual user's understanding of

the system is limited to the generalized concepts of records, indexes or keywords. Throughout this paper, and unless specified otherwise, a "user" will mean a "casual user" as defined above.

Let's observe a casual user wishing to have access to a retrieval system. He is unfamiliar with the search procedures, the command language, the database. Even more, he could be a novice in the use of a terminal or logging in procedures. He needs help! The most likely option for him, at this point, is to go to a professional "searcher" (1) and formulate his wishes. However, a problem of communication will rapidly take place if the professional searcher does not have some knowledge in the scientist's field. The best solution, in order to remedy this specific difficulty, is for the two of them to work and search together. Of course, the solution is far from ideal (2) : a gap still exists between the two, and frustration is quite likely if errors or delays result. What is more, the user always needs his information "immediately", when the professional has some "other urgent matters to attend to, before proceeding with this request..."

--

(1) In a university environment, a librarian would have the appropriate training.

(2) Even if it is the most recommended approach.

2.4 "Leave the Casual User Alone!"

Thus, there exist many cases when the user, the "scientist" from above, would like to perform his searches by himself. By doing so, he will have more freedom and he will get the feeling of moving in "terra cognita": after all, those formulas, those scientific names, if they have no meanings to the librarians, they should have some for him... Above all, performing his own search, the user has the opportunity to "browse" through the records, like he would do about library shelves. It is because of all these points, that it has become interesting to investigate the feasibility of systems which would allow casual users to search the databases without external help.

However, as we will see, the options and the problems are numerous. The options are, for example, to train the occasional user in a quick and easy way, but the problems there are too clear: which shortcuts are acceptable? Others options are to simplify the language(s) needed to access a system. As long as some "standard languages" do not exist, would it not be good to have a language allowing access to different systems? It would also be valuable to have interfaces "counseling" the users, before sending the queries to the commercial system: users would save time, money and much irritation. Those solutions have been investigated already, and this paper will detail them to the

reader, with their good and their bad points. Other ways are open, as will also be shown.

2.5 Goal of the Paper

The author of this paper firmly believes that online system searches benefit from user involvement. Thus this paper is going to study ways to guide an inexperienced user through the maze of the searching world. It is first going to study the question of training the users, and then different systems which have been realized during the past few years, all in the hope of curing the unfriendliness of retrieval systems toward inexperienced users. Finally, the author will offer some personal remarks, comments, predictions for the near future, as well as some guidelines for user-friendly systems.

3. EDUCATION

The easiest way to have users able to fully use a system, is to give them some kind of formal training. Most commercial system's vendors will be pleased to send representatives of their organizations, in order to train the future users. For several days, within a classroom or online using some "canned example," the naive users will slowly lose their innocence!

Manuals are also of great value, of course, and should always be available. They describe, hopefully in laymen's terms the how - and sometimes the why - of each command. If the instruction manuals are considered insufficient or, more often, too complex for beginners, simpler guides are available, explaining commands step by step. (1)

This paper stated earlier, however, that a casual user does not have the time nor the desire to sit in a formal classroom in order to be taught all the tricks and short cuts of a system he will use only once or twice a year. In this section, the reader will find only a quick overview of some of the most original approaches.

3.1 CAI/CAL or Computer Assisted Instruction/Computer Aided Learning

In this section, the reader is invited to consider that some form of training seems indispensable for any user: the question is how little is enough, and how in depth can it be without bothering the casual user. For these reasons, the formal type of training is not analyzed in this study. The literature on the

(1) For example, Robert Laurence wrote a "Self Teaching Exercise" for LEXIS, which is a textbook at the University of Illinois Law School [Laurence, 78].

subject refers to Computer Assisted Instruction (CAI), to describe training offered online. There are two types of instruction which can be differentiated:

- (1) The first type is the instruction as offered by an online system. TRAIN, for example, is offered online by DIALOG. This category has not been specifically designed for casual users, and it is often time consuming and expensive. Thus it will only be mentioned here.

- (2) The second type is CAI as a "canned exercise". In this case, the student is instructed offline, by examples and exercises, using simulation, emulation of a system, or a subset of a system's databases. This solution has the double advantage of being cheaper and, potentially, more individualized (ie: designed for a specific class of users). In this class, one of the most interesting and most successful cases is the TRAINER, described below.

3.2 TRAINER

The University of Pittsburgh has a CAI system, called the TRAINER system, which has been in operation since 1978 [Caruso,

78]. It provides both instruction and training for online searching, and teaches users, by emulation, how to use a retrieval system. That is, it simulates the functions of retrieval systems, and allows users to obtain training, without using expensive commercial service connect time and telephone connections.

TRAINER was developed by Elaine Caruso, under an NSF Grant [Caruso, 77]. It helps users to learn how to access a system and operate searches in an economical way, with feedback, and no time pressure, because the trainee is not connected to commercial systems while experimenting. The TRIANER system teaches users to operate retrieval systems by emulating them on a stand alone computer.

The user can use DIALOG-like commands, or ORBIT-like commands in order to emulate actual searches. The searches are performed on a subset of Lockheed (for DIALOG) and SDC (for ORBIT), accessing 3 data files, and the entire session looks like real DIALOG and/or ORBIT.

A user who does not wish to, or can not spend time in a classroom, but would benefit from a session of training, would be a good candidate for TRAINER. At his own pace, he could choose any of the seven instruction modules available, or directly go to

the emulation modules, and start familiarizing himself with them.

The advantages are reduced cost, because no online connections with the commercial systems are required (1) and convenience, as the user gets trained on two widely used systems, at his own speed and when he wishes. What is more, a casual user can easily select the TRAINER's exercises he needs, and get information and training on, and only on, a particular subset. The capabilities offered by individualized training are one of the requirements needed in order to please a casual user.

However, there exists some important limitations, the major one being that the training is done on a very restrictive subset of the databases. Also, the code is written in ANSI FORTRAN, certainly not the best choice for data and string processing. Also, the portability of the system is far from being adequate: the TRAINER runs at University of Pittsburgh, on a PDP 11/40 and many attempts to export the system have only shown that TRAINER likes DEC best [Caruso, 81]!

After this very rapid overview of what kind of training is available online to the casual user, the next chapter of this paper is going to take a close look at the way some systems have

(1) TRAINER is available for dial-up access through the educational network EDUNET/TELENET.

approached the problem of casual users desiring hands-on experience.

4. STATE OF THE ART

In the next few sections, some systems, research and experiments will be presented. Each one of them has been selected because its particular approach in handling casual users is original and of importance. The selection was also based upon the availability of literature and published material. (1)

First, LEXIS will be presented as a commercial system intended for a casual user community. Then CONIT, a research project at MIT, will be explained and will show how it is possible for a user to access many retrieval systems using a single language and a set of procedures. IIDA, from Drexel University will give the example of a system helping the searcher to search "well". Finally CITE will give an example of a natural language approach.

(1) Which explain the number of NSF grants studied in this report

4.1 LEXIS

4.1.1 Presentation

The first system this paper has chosen to study is LEXIS. There are many reasons for this choice, the most important one is the fact that LEXIS has, indeed, been created for a "casual user" community, as defined earlier. Another reason is the wide acceptance of this system in this country and overseas. Finally the ease of use, and the little training required makes the experience worth being studied.

One of the most successful examples of online retrieval system for casual users is given by Mead Data Central's LEXIS, and the newer NEXIS and LEXPAT, (however, for the present study, only LEXIS will be discussed, as the 3 systems are quite similar, and LEXIS is by far both better known, and more heavily used).

LEXIS was created in 1967 (1) and started its nationwide expansion in 1973. It was designed to be an interactive time-sharing system, specially molded for lawyers' use. Its databases contain full text of all federal cases, as well as state cases in dozen of states, and other specialized libraries.

(1) Back then, LEXIS was known as OBAR ("Ohio Bar Automated Research").

The user has also access, with special billing, to DIALOG, New York Times Information Bank, Encyclopaedia Britannica and NEXIS. The hardware and retrieval procedures have been designed with the legal community in mind, and the legal community is a perfect example of a community of "casual users" as described above. Lawyers are professionals, who certainly feel that their time is valuable -at least, it is expensive!-. They do know how to search, as they are familiar with the concepts of keywords, indices and abstracts which they use in their libraries, but most of them are computer-illiterates. LEXIS answers most of the legal community's needs.

By the end of 1983, LEXIS functioned on two Amdahl 5860 mainframes, located in the Mead Center, in Dayton, Ohio. The network is the Med-Net Network and the terminals needed by the users are owned by LEXIS (however, because competition is now shaking the market (1), LEXIS is now also available to the owners of IBM PCs, IBM 3101, IBM Display-Writer and Televideo 950 terminals).

4.1.2 Searching LEXIS

(1) Note: In the US, competition to LEXIS is composed mainly of WESTLAW, JURIS (Justice Retrieval and Inquiry Systems), AUTO-CITE (Automated Citation Testing Service) and FLITE (Federal Legal Information Through Electronics).

A LEXIS user will typically search words or sentences within a text; Boolean logic is used, and a few common words are eliminated. It is a full-text system, with no pre-indexing, and therefore the user can search the entire text for a word or a sentence. But what makes LEXIS especially interesting, and that is why it is discussed in this paper, is its ability to communicate in plain English. The function keys are numerous and clearly marked. For example, some are labelled: NEXT PAGE, NEXT CASE, HELP, CHANGE FILE, CHANGE LIBRARY...

The following function keys are worth being mentioned:

- (1) FULL displays the full text.
- (2) CITE gives title, date and formal citation
- (3) KWIC shows searchword in context (25 words)
- (4) THESAURUS will suggest synonymous, related terms.
- (5) CLIENT will help billing client to relevant searches.

LEXIS allows features which are quite interesting, all very easy to use, such as a search in a range, where the user can find a keyword within "n" words of another keyword.

For example: 'louisiana w/5 university' will retrieve all occurrences of the citations mentioning both words within 5 words of each other, with the searchwords highlighted. (1)

According to MEAD's recent publicity, the average session online lasts 15 minutes, and the average time for retrieval is typically 15 seconds.

4.1.3 LEXIS Conclusion

In its limited area, legal searches, LEXIS has triumphantly resolved the problem of a casual user accessing an online system without external help. It is possible for a user to sit at a terminal, without prior experience, and to retrieve meaningful, and complete information in a reasonable amount of time. Thus, LEXIS has seen, since 1975, the creation of many associations of users, in local law libraries and bar association. In those environments, users are clearly occasional searchers: it is because of their rare need for online retrieval, that those lawyers could not justify a personal subscription to the MEAD's system. LEXIS does seem to satisfy the casual users in the legal world [Larson, 80].

(1) In "LEXIS Legal Research", published and distributed by Mead Data Central to prospective clients. Copyright 1982 MDC (pp. 11).

LEXIS has also resolved the problems inherent to the most traditional type of searching such as the time lag before hard copy is available and the subjectivity brought by indexing: the time lag is negligible (1) , and the subjectivity is absent, as indexing has disappeared. And since indexing has disappeared, all data is searchable.

But, non-indexing has also its drawbacks. The main argument against it is, of course, the fact that many irrelevant documents are retrieved. It is also frequent, for somebody not familiar with the laws and the legal vocabulary, to miss some cases, maybe some important ones. In other words, it can be said that the role of the indexer has always to be assumed by somebody. In full-text, the searcher himself has to play the role of the indexer.

Because of those points, the use of LEXIS is incompatible with users who do not know well the world, and the vocabulary of the legal environment. It is also impossible for a searcher to perform well as long as he has not identified relevant terms and searchwords. Thus, LEXIS has gone a long way from more traditional retrieval systems: a LEXIS' searcher will perform

(1) The time lag could eventually completely disappear, if and when the court reports and laws are entered directly in the databases, a possibility not completely utopian.

well as long as he knows his domain. His expertise in law is primordial, and knowledge in other conventional systems would not help him very much.

The domain of legal research will change quickly in the mid 80's. The concept of full-text retrieval is clearly favored among lawyers, but, as the size and number of files increase, it is clear that more research is needed in order to keep retrieval speed and efficiency at the high standards which are known today. (1) As Users become more numerous, and require more searches, they are likely to ask more from these systems which have performed so well in the past.

There is also considerable competition on the legal market. Westlaw, LEXIS' major competitor has made great progress in the past five years. It has added full text search capability, once unique to LEXIS, while keeping its "key-number" system, so familiar to lawyers. (2) Also, Westlaw is compatible with almost any hardware on the market, an issue of interest to small firms, and which LEXIS will have to accept.

(1) Many authors argue that the growth has attained some maximum, and that increases are likely to slow down. However, as more and more cases are being argued, and retrospective material is being added, the size of legal databases will continue to grow for the next few years.

(2) West Publication is the major legal publisher in the U.S.

Thus, the casual user seems to be the winner in the legal profession: the quality of past service and the new competition have forced LEXIS to improve, and users can expect extended services in the years to come. Finally, it should be noted that LEXIS is also remarkable by the fact that even if it allows easy use for the infrequent user, it does so without inhibiting the experienced searchers: very "fancy" searches are possible on the system. Thus, the two class of users - casual and experienced - are united under the same interface, which is a rare achievement.

4.2 CONIT

CONIT, for "COnnector for Networked Information Transfer," is one of the best examples of an attempt to help the casual user of information retrieval. It is an experimental computer interface which was developed at MIT, by the Electronic Systems Laboratory, under Richard S. Marcus. The system was designed to provide a translating tool between the users and various retrieval systems. CONIT's success has been demonstrated experimentally, and it has been used as a basis for other projects. (1)

(1) For example, see IIDA, further in this paper.

4.2.1 Prēsentation

CONIT allows the user to access many different retrieval systems, using a common language, during a continuous session. Usually, the heterogeneity offered by a group of systems creates obstacles at different levels: the user must know different access procedures, in order to log-in and to exit the system. He must also know different languages in order to perform searches and request outputs. Finally, he must be aware, even if only slightly, of the indexing vocabularies and the retrieval capacities of the system. As a matter of fact, even within a given system, it is possible to find differences in indexing methods and other inconsistencies, like difference in catalog record fields.

The approach used by CONIT is quite original: it can be described as an attempt to present to the user a single "virtual system," with a single manipulation language. The virtual system consists of many commercial retrieval systems, with all their complexity and originality preserved. However, to the user, all those system "look like" a unique and homogeneous system.

CONIT works on the MIT's MULTICS system. Currently it can access four commercial retrieval systems: Lockheed DIALOG, SDC ORBIT, NLM and SUNY Medline, as well as MIT's own "INTREX."

MULTICS accesses those systems through a device called the "autocall." Autocall takes care of dialing the system's number, and informing the user of the reasons for delay, if they occur.

The CONIT interface provides a common access to a network of different online bibliographic retrieval systems. The user views a global or virtual system, which he accesses through a common language. That offers a flexible and dynamic means for handling the interconnection between the searcher and the database.

4.2.2 Study of CONIT

The language used by CONIT was designed in such a way that all of the functions needed for information retrieval operations can be expressed. Thus, each language $L(i)$ from the original systems, has been broken down to its most elementary pieces: each one of the elementary pieces is unique, that is, two different languages will not have two similar elementary pieces, unless their meaning is equivalent to each other. From those unique parts, a common language function is built as a MACRO of the common language.

The structure of the language is always:

- (1) VERB (which is a Command).
- (2) A space.
- (3) An argument (where an argument can be a list of arguments).

The verbs can be abbreviated, and space and arguments are not always required.

4.2.3 Searching with CONIT

The stream of input coming from a user, or from a system in response to a user's request, is matched against a set of rules. The advantage of matching strings against a table is that the number of rules can be varied over time, and the rules modified if necessary. Also, the rules are organized in such a way that longest matches occur first: thus, a rule NNNN will match first an incoming string NNNN. But, if a match does not happen, a rule NNNX would be checked for matching, and the process repeated until the end of the table is reached. This smart approach has helped keep the number of rules to a very small number. (1)

An example of a rule can be given by:

-
- (1) In CONIT 3, less than 80 (basic) rules were necessary

--> RLE--3/TELENET//Telenet Responding/RLE::4/Send

1 / 2 // 4 / 5 / 6

where:

1 is the Context String (the State at Start).

2 is the Match String (the Incoming String to be Matched).

3 is the Host Message (the Message for the System, if any).

4 is the User Message (the Message for the User, if any).

5 is the Next String (the Next State).

6 is the Special Action (if Action is Required).

The meaning of the rule in this case, would be:

1

R for Retrieval System : the message comes from system.

L for Logging Procedure : we are in the Logging stage.

E for Telenet : the network is Telenet.

- for "do not care" : any character would match.

3 for step 3 of Logging.

2 TELENET is the string to match.

3 no message for the host, since host is the sender.

4 "Telenet Responding" message is sent to user.

5 Next Context is R, L, E, : (do not care) and step 4.

6 If match occurs ==> action.

As a further example of commands, and capabilities of the system, the reader is invited to consider the following examples.

A user entering the command:

--> PICK SYSTEM-NAME

would initiate the following procedures, which will occur without any further commands from the user's part:

- (1) Send message(s) informing user of "what's going on."
- (2) Dial the correct number of the network
--(Tymnet/Telenet), via the autocal.
- (3) Inform the user of how to leave the system
(disconnect).

(4) - And send the following messages, as soon as appropriated:

- (a) Phone connection made with [Tymnet/Telenet].
- (b) [Tymnet/Telenet] responding.
- (c) Logging into [System-Name].
- (d) You are now connected to the [System-Name] retrieval system.

>>> At this point, the user will choose a particular database: (i.e.: "SOCIAL SCIENCE (No 51)")

- (e) You are now connected to the [SOCSCI (NUMBER 51)] database.
- (f) For explanation of how to find a document, type: 'e find'.

and finally, the user is logged in. In the previous example, the user would be logged in, ready to use the database of his choice.

In most of the cases, however, the casual user will not have to bother about selecting a specific system. After a:

----> SHOW DATA

command, CONIT will indicate a set of available "areas of interest". From this set, the user will select one group. Let's

assume for the sake of example, that the user wants to investigate the area of Social Sciences (No 51). He will type:

---> PICK 51

Note, that all he had to do, was to select an area of interest: very often the casual user is not interested in knowing which system brings him the information. CONIT will select a system according to system selection rules, which can be overridden by the user at his choice. Two of the most important rules are:

- (1) If a system is already online, use that system.
- (2) If MEDLINE is desired, use SUNY, instead of NLM, since SUNY is cheaper and often less busy.

Note also, that in order to switch from SYSTEM-NAME to SYSTEM-2, the user will only need to type:

--> PICK SYSTEM-2

and all of the necessary logging out from SYSTEM-NAME, dialing and logging in to SYSTEM-2 will be taken care of by CONIT.

It is now possible to search, change database, consult the index, request outputs and keep the results of searches in an

out-file for later consultation.

4.2.4 Comparing the Languages

On page 34, the reader will find some examples of the CONIT requests. They have been compared to the DIALOG and ORBIT requests they simulate. A column of MADAM equivalent commands has been added for references (MADAM is a retrieval system which has been developed at USL, Lafayette, LA, where this paper was also written). SUNY and NLM Medline examples have not been given, because their syntax is very similar to the language used by ORBIT.

Note that translations are necessarily approximate, since exact translations are impossible, as will be shown later in this paper.

CONIT	DIALOG	ORBIT	MADAM
PICK system	[logout/login]	[logout/login]	MADAM / QUIT
PICK DATA file	* FILE file no.	FILE file name	change db
SHOW INDEX term	EXPAND term	NEIGHBOR term	(see footnote)
FIND topic	SELECT topic	FIND topic	topic
FIND AUTHOR name	SELECT AU = name	FIND name (AU)	SE AU = name
FIND a AND b	SELECT a (C) b	FIND a AND b	SELECT a AND b
COMBINE SET m AND SET n	COMBINE m * n	m AND n	SE SET m AND SET n
COMBINE SET m OR SET n	COMBINE m + n	m OR n	SE SET m OR SET n
SHOW	TYPE x/2	PRINT	DISPLAY
SHOW TITLE	TYPE x/6	PRINT TI	DISPLAY [ti]
SHOW OFFLINE	PRINT x/2	OUTPUT OFFLINE	PRINT
SHOW SET n	TYPE n/2	Sn/OUTPUT	[on] DIS. [n]
SHOW NEWS	?NEWS	NEWS	* db [or] sb
EXIT	LOGOFF	* STOP	QUIT

Note: Index Browsing exists in MADAM as Interface, not yet fully implemented.

TABLE 1 Comparison Between CONIT and Three Other Systems

4.2.5 The Problems

CONIT has demonstrated, after 7 years of research and use, that it was indeed a successful tool in helping casual users retrieve relevant information from different databases, and CONIT did so without causing too great an increase of time online. However, the system has also shown some difficulties inherent to such an approach: some are clearly due to the implementors' choices, while others are somewhat accidental; these points will be presented and analyzed here.

The problems with CONIT, at least CONIT such as described by the literature available to the author of this review, come mainly from the imperfection of the translation. The question of "how exact must a translation be?" is one which can be argued for quite a long time, because, just like for human languages, a perfect translation is probably impossible to obtain. So, "how good is good enough?" is up to the designers to decide. But the following points are worth noting:

- (1) --By simplifying the language, many valuable options have been lost from the original, - even if those options were available in ALL of the original systems. For example, the systems retained by CONIT all have the capacities of "Search History", and to repeat on

- equivalent search on different databases, but CONIT does not allow those facilities.

(2) Some particularities of systems have been preserved, which is clearly a choice of the designers. But the global view of the virtual system suffers from such exceptions.

--> PICK DATA FILE

is a number in DIALOG (for .FILE FILE no), but a name for ORBIT (for FILE FILE NAME).

An other example is the

--> SHOWING NEWS

which is not sufficient for all systems. Thus, for the MEDLINE news, the following is needed:

--> PICK NLM NEWS

Of course, there are many difficulties in constructing
- a consistent translation, and this paper points only
to those weaknesses, even if reasons are recognized.

4.2.6 CONIT Conclusion

In conclusion, CONIT has clearly shown that it was possible for a computer intermediary system, to assist users with no previous experience, in retrieving information from dozens of heterogeneous databases, coming from 4 different systems. It has also proved to be a useful tool, and a valuable help to casual users; however, its efficiency and performance are clearly inferior to "the retrieval effectiveness achievable by expert human intermediaries working in conjunction with the end user", as Marcus himself puts it [Marcus, 81]. The question of "how exact a translation should be" remains, and it is one which become more important all the time, with natural languages being studied more intensely.

4.3 IIDA

An example of a computer system serving as intermediary between a user of online bibliographic systems and the system itself, is the IIDA (Individualized Instruction for Data Access).

4.3.1 Presentation

IIDA has been developed at Drexel University, with the financial help of the National Science Foundation. Professor

Charles T. Meadow started the project in 1976. The Software used in IIDA, is a direct descendant of MIT's CONIT and Caruso's TRAINER, both already described. The MULTICS system is the computing environment used for the project, and the Data Base Search System is Lockheed DIALOG System.

4.3.2 Study of IIDA

The IIDA was designed to offer online instruction and assistance to casual users of bibliographic database systems. This extensive help is available to the relatively inexperienced searcher without human intervention. IIDA offers the example of a system where major procedural errors are detected by the computer, and where inexperienced searchers are offered assistance in order to complete their searches. This kind of intermediary facility is known as an "expert system."

4.3.3 Two Modes Available

IIDA is an interface lying between the terminal of the searcher and the retrieval system of the search service. (1) IIDA acts as a "screen," and provides instructions and diagnostic capabilities. That is, it will monitor the user-system exchange,

(1) At this time, Lockheed's DIALOG is the retrieval system used by IIDA.

it will keep track of all data transfers, record history and perform analysis of each and every search. It will also give all this information to the user, if requested to do so. In some cases, IIDA will even take the lead, and signal to the user some errors, inconsistency or messages related to the state of the system (errors, disconnection, line over-use, shut down ...)

IIDA works in two distinct modes, which the user will select himself: the "exercise mode" and the "assistance mode", which are explained below.

Exercise mode. A first time user, or a user who has not searched for quite a long time, will probably favor this type of monitoring; here, IIDA exercises extensive monitoring, and the user is closely "watched" by the system. This mode is to be used mainly for training, and for improving search skills. It can be described as an introduction to the commands and to the development of search strategies.

Three different types of exercises compose the exercise mode. The user has the choice of which exercise to go through. The order represents an order of increasing difficulty. During each of the sections, the user has access to the "help" library.

- (1) Exercise 1 is a "canned exercise". The user answers IIDA suggestions (menu driven or prompting). This

way, the user learn the "basics" and the syntax of the language. Each presentation of a concept is followed by examples that the user completes. The system does not intervene if the answer is not correct: it is a very basic introduction to the system, but enough to get started.

- (2) Exercise 2, which uses only a subset of the language, allows limited search with only: BEGIN, EXPAND, SELECT, COMBINE, PAGE and TYPE which were introduced in exercise 1. The search is also restricted in its sequencing (i.e.: a combination is necessary before any output can take place).
- (3) Exercise 3 utilizes the full language, but there is still a restriction in the logical sequencing of the instructions.

Assistance mode. The assistance mode is the normal mode to be used when performing searches. The IIDA monitoring is very discrete, intervening only when difficulties appear, or when specifically requested by the user. In order for IIDA to perform its counseling role, it must gather enough information from the search, as the reader will see below.

4.3.2 Information Gathered

During the exchange between the user and the system, IIDA gathers the history of the following data:

- (1) Command: The text of each command is stored as entered, and then parsed, in order to have each element of the command stored as independently addressable data.
- (2) Set: All sets created are stored (just as the DIALOG "ds (display set)"), with additional pointers and descriptors which allow further comparisons. Also, those sets are clustered according to similarity of defining terms.
- (3) Descriptor: All the descriptors used in search commands are present in a descriptor table. The descriptors which have been already viewed by the users are specially marked.
- (4) Sets retrieved: A table of all sets viewed is also formed. This table indicates for each record retrieved, its relevance and its ties with a set.
- (5) Error: Each "ERROR" is noted in its context, and thus

available for further analysis.

All this data is gathered for further analysis of the user-behavior. For the sake of keeping statistics and general system evaluation, IIDA keeps also track of all the features searchers use on the system. Mainly, the HELP session are kept in a separate table.

4.3.5 Monitoring User's Activity

As was said earlier, the main purpose of an expert system is to help the user in his necessary exchange with the computer. Thus, IIDA will monitor this conversation, and pick up some (if not all) problems. A list of most of the categories of problems identified by IIDA follows: it is not exhaustive, but represents the framework behind such a system.

(1) ERRORS:

Syntactic Errors: such as invalid commands or abbreviations, invalid characters, bad format in command, invalid operators, parenthesis missing ...

(2) POOR USAGE:

(a) Under use of facilities:

- Failure to EXPAND, especially after creating

several null sets.

- Failure to TYPE or DISPLAY before issuing a PRINT command, or format not correct.

(b) Over use of facilities:

- Excessive TYPE
- Excessive EXPAND
- Excessive time spent in between commands.

(c) Correct, but poor use of facilities:

- unnecessary repetition of commands.
- "Excessive" null set generation.

4.3.6 Diagnoses

As was explained above, IIDA will "screen" the user's input, and diagnose his behavior. There exist two levels of such analysis:

- (1) At the first level, IIDA will recognize a syntactic error, that is, a query that DIALOG would not be able to recognize. The query is not sent, and the users is informed by IIDA of the error.
- (2) At the next level, IIDA will make a further analysis into the context of the input: for example, the strategy used during the search is observed, and if

"judged" nonproductive or inefficient by the system, IIDA will suggest variations. That is, IIDA will take an "intelligent look" at a query of the type: "Correct, but poor use of facilities", as described above. The system, once it has recognized the strategic errors, will offer some solutions in order to improve performances.

Here we can see that IIDA does not stop its services by issuing an "error message", but goes one step further by giving advice, indeed a unique approach. Meadow and his team had to define first what a "good strategy" was. From this definition, they gave rules adaptable to the system [Meadow, 79b].

The strategy of a typical search is derived from Penniman's cycle. David Penniman in his Ph.D. dissertation [Penniman, 75] describes one cycle as a pattern of repetitive commands such as: BEGIN, EXPAND/SELECT, COMBINE and PRINT in that order.

The number of each step varies, according to numerous factors, but the order is always respected within a cycle. If a search needs more than one cycle for completion, the set of "n" successive cycles determining a search, is called a string.

All the diagnostic capabilities in IIDA are based upon this

observation. (1) Most of the errors and warnings come from the observation of a broken cycle or from a cycle/string going over some predefined limits (size and number).

When IIDA receives an input, it first determines its validity. If the syntax is found correct (that is, DIALOG would recognize the input as a valid command), IIDA takes a look at the "strategy" used. A strategy found incorrect or inefficient will generate actions from the Warning Control Program (WCP), an important piece of software within the system. If called, the WCP will take one of the following actions:

- (1) It will send an "error message" to the user. In the choice of wording, Meadow's team was very careful about using "neutral" messages, that is, messages which would not adversely strike the user. Indeed, an important consideration for casual users, when the reader realizes that most of the messages by "computers" are accusatory: the user did something wrong. Not a "friendly approach"!
- (2) The WCP can defer the message. That is, if the user has just been warned of some actions, and he repeats

(1) Further developed by Oldrich Standera [Standera, 75].

it, IIDA will leave him alone. It is hoped that the user knows what he is doing. The first implementations of IIDA were going further in this direction, allowing the user to stop messages (type "/SLACK"), but such fancy requests are far from the casual users' casual needs.

(3) A message can also be suppressed. For example, if an error generates a set of errors, only the "most important" will be selected, in order to simplify the corrective action.

(4) A message can also be "enhanced", that is, complemented with relevant information. For example, if a message has been deferred "n" times, and finally released, the user could be informed that the message had been hidden from him "n" times.

Another important piece of software is the "help" library already mentioned. It allows quick advice (type "QA") as well as return to the exercise sessions. In fact, IIDA has the capacity of moving the user back to instruction level, if it is judged needed.

4.3.7 IIDA Conclusion

In this short presentation of IIDA, the author of this paper has tried to show some of the most interesting points offered by the system. The reader can see that IIDA presents much originality. However, the philosophy behind it, as well as many of the approaches are not new. On the contrary, Meadow obviously knew how to advance his system by using state-of-the-art work: thus Penniman's understanding of a cycle, CONIT's tables and parsing, as well as Caruso's TRAINER, all have been used to complete this remarkable tool, which has been successfully tested outside of "academia", by EXXON for example [Landsberg, 80].

4.4 CITE

The next example studied in this paper is an "Intelligent" approach to the same problem of a casual user wishing to access a retrieval system, with no external help.

CITE is an example of a system which can be queried in everyday English. The importance of such work can not be overemphasized, because if the efficiency and ease of use of such an interface can be demonstrated and extended to other systems, casual users could have their desires becoming reality.

4.4.1 Presentation

Since 1978, a user can access the National Library of Medicine's MEDLINE (Medical Literature Analysis and Retrieval Systems Online), using a natural language approach. The interface which allows this interesting approach is known as CITE, for "Correct Information Transfer in English". Queries are issued by the users under the original form of English sentences, paragraphs, groups (or lists) of terms or phrases which are compared to the set of titles, abstracts and controlled vocabulary of the system. The origin of CITE starts in 1978, with Doszkocs' designs of AID (Associative Interactive Dictionary), which allowed users to search the large bibliographic files of MEDLINE and TOXLINE, using queries in natural language. CITE evolved from this original approach [Doszkocs, 79].

In designing CITE, special attention was given to efficiency and quick response time. But, above all, the design of the language had to follow the following rules:

-
- (1) Assume that the user has no familiarity with the system.
 - (2) Information will be solicited in a natural manner.

- (3) Try to prevent "frustration" on the part of the searcher [Doszkocs, 79].

4.4.2 Study of CITE

CITE proceeds by realizing a number of successive operations, represented in Figure 1 (page 51).

The steps are as follows:

- (1) The casual user enter his query in plain English.
- (2) The system recognizes search terms in the incoming input by matching the incoming string with a stopword list of 600 words, and indexes are retrieved by comparison to the MEDLINE inverted file. At this point, synonyms, various spellings and other variations are recognized, and unified under the controlled vocabulary selection.
- (3) The search terms are processed, and each one of them is assigned a weight. Pointers are adjusted to each reference, in order to minimize the processing time, if the same term is called again.
- (4) The user gets a display of a set of titles with a

-
weighted value of terms.

- (5) Relevance feedback, where the user selects the titles which he found relevant to his search from the complete set of titles retrieved is employed.
- (6) With PRINT, the user gets a full listing of the titles he is retrieving, and thus is given an idea of the path he is following.
- (7) Modification of the query in light of the items the user has judged "relevant" is performed.

1/ ENTER

ENTER QUERY

2/ IDENTIFY

RECOGNIZE
SEARCH TERM

3/ COMBINE

PROCESS
SEARCH TERM

4/ WEIGHT & RANK

DISPLAY
RANKED TITLE

5/ RELEVANCE
FEEDBACK

SELECT TITLE

6/ OUTPUT

PRINT
CITATIONS

OR

MODIFY
SEARCH

END

FIGURE 1 The CITE Cycle

4.4.3 CITE Conclusion

The approach of CITE is quite original. It is however difficult for the author of this paper to really judge the value of such an interface, because of the lack of literature available. For example, it is of first importance, for the critic, to have an idea of what the stoplist looks like, as well as the way the records are treated once they have been recognized.

If the system were to prove successful, nothing stops it from further expansion, as nothing in its design makes it exclusively designed for MEDLINE.

4.5 CCL

A Standard for Euronet-DIANE. A solution which seems important consider is the possibility of having some type of a "standard language" which would allow retrieval from different systems. If this Esperanto of Information Systems were to exist, the casual_user would have to learn only one language.

4.5.1 Question of Standard

The problem with implementing a standard language is certainly not technological: it has already been done, and can

even be done quite efficiently. The most famous example, so far, is the Euronet-DIANE's Common Command Language (CCL). Euronet-DIANE is the result of a European effort to put together the wealth in information of the different European countries: the resulting databases would also gain if they were accessible through a unified language, on a single network. Many governmental and commercial organizations have accepted the proposal. Euronet has been operational since 1980. In 1983, 5 different European systems had accepted to implement the language CCL on their individual retrieval systems. They often choose to utilize "front-end" translators to modify their command languages, instead of separate computer interfaces.

4.5.2 Presentation

Unfortunately all that CCL has from "Common" is the first initial! And that is where the problem of standard language resides: those 5 systems have agreed to run a "common language," which, after only 2 years, has already evolved in two distinct subsets (1) [Verheijen-Voogd, 81]. The divergence between the two languages are quite small (mainly punctuation and use of symbols), but it is certainly enough to give a bad name to "Common Language" and headaches to the casual users.

(1) Those 2 languages are "IRS/ESA" and "DIMDI"

4.5.3 Problems in Standard

How Euronet-DIANE let itself be cornered in such a problem can easily be explained if the reader agrees to consider the following points, which are traditional difficulties in implementing standards: (1)

- (1) A standard will always be AGAINST current or past methods. In the case of languages, it is difficult to convince implementors and users that the new one will be better.
- (2) A standard is difficult to sell because it attacks some economic interests. Neither the commercial nor the governmental agencies see with pleasure the increased workload and cost resulting from implementing standards.
- (3) How can one be convinced which one is "better"? What is better, anyway? Better for one user, does not necessarily means better for all.
- (4) The size and advanced state of the current implementations make change difficult.

(1) Reference the social tragical comedy of the implementation of metric standards in the US!

In the case of Euronet, this last point should not have counted, but the other interests were too strong, and the "Community" now has at least 2 new languages to cope with!

5. REMAINING QUESTIONS

So far, this paper has shown different examples of realizations tending to simplify the access of retrieval systems to the casual users. Of course, there are many other approaches possible. In the following pages, the reader is invited to consider some aspects of the problem which have not been mentioned in the preceding pages.

5.1 New Technology

Studies in man/machine interface are taking increasing importance, since computer communication and information systems have been realized which not only seem to be getting increasingly more economical, but also more reliable and more responsive. This section is going to survey some of the interesting points which are being developed, and could, in a near future, simplify the casual user's access to retrieval systems.

5.1.1 Micro Computers and Smart Terminals

It is an easy guess to predict that within a near future much help will be directly given by the user's micro-computer. Already now, it is possible to access information retrieval systems through intelligent terminals or a micro-computer, and have those tools perform many redundant and annoying tasks. For example, there are many necessary "housekeeping" operations which are needed to access a system; it is necessary for the user to dial a network, issue an ID number, a password, an account number, the name of a database and a file ... All these operations are cumbersome and, because they are somewhat mechanical, they are very easily performed by a computer.

For example, a user could enter his name on his micro and call his favorite online system. Assuming that the modem is of the "smart" type (1), the user will have to wait patiently for the connection to be accomplished through the network to the system desired. Thus, it can be seen that micro-computers can take care of the login procedures, like the "Autocall" previously described⁻ did for CONIT's users. It makes sense to go one step further, and propose for the computer to allow a selection of

(1) Like the "Smartmodem" from Hayes, these modems transform character input into a phone number, and dial the number by themselves.

system. For the casual user, the only choice remaining would be to select the name of the desired system out of a menu.

Once online and ready to start, the user has to call a database. Some systems - like DIALOG - put the searcher, upon entering the system, within a default database; but the micro could easily, as a part of the login procedures, call a user-defined default. It is easy to imagine that the micro could insure some kind of interfacing, and help the casual user: in other word, a micro-IIDA!

Goldstein has shown [Goldstein, 78] how an interface could simplify the searching of a particular database (CATLINE) of a particular system (MEDLINE), by replacing the system language with a simpler one. His remarks can easily be generalized to other systems and other databases. Indeed, as this paper has previously shown, it would be interesting to implement some kind of "translators," where a user would be allowed to enter queries for any systems, using the system language he knows the best! (1)

The interface could also easily store repetitive queries and generate those streams upon requests. These mechanisms are now available in most systems. They can be implemented by the interface, giving the casual user an important advantage: he

(1) For example, accessing DIALOG, using ORBIT's queries

would have all the time he needs to build his queries and he would get to know "his" interface. Knowing the interface in this case could be more important than knowing the systems, since the queries would be directed to the interface.

Finally, and the most important point, the interface could "capture" the output. Instead of having the cycle:

Retrieve, Check, Print, Retrieve, Check, Print, Retrieve ...

until completion, or searcher exhaustion, a user's station could capture the data, copy the output into its memory (diskettes), and logoff. At this point, all the data is available to the user and ready for further processing. When the user is offline, he does not have the time pressure mentioned above. He will be able to edit, format, clean-up and select his output at his own convenience. If more information is needed, the user can renew the process. Note also that the exchange can easily be performed at 1200 baud or faster, instead of a slower output, required by reading online.

Another possibility is for the interface to transform the data. For example, a query to a database can give some output which the user would like to use as input for further queries. This transformation process could be realized by the interface,

with the double advantage that omissions and typographical errors are eliminated: all of the output can be transformed, and appear as a stream, in the same sequence as the output.

5.1.2 Graphics Interface

Even if it is not of first importance, as far as data retrieval is concerned, a report about man/machine interfaces can not be complete without a mention of graphical support. The exchange between user and machine has so far been restricted to conceptual and linguistic rules. However, the realization of such system as the LISA from Apple has shown the way toward new approaches.

Without pretending to judge the effectiveness or the necessity of such realization, the reader is invited to recognize the following trends, which are very likely to be further developed in the near future:

- (1) The concept of a user pointing to an image, a graphical representation of a concept instead of formulating rigid and/or lengthy queries.
- (2) The reduction of the amount of wordy dialogues: fewer terms for the user to memorize, type and correct.

- (3) Repetitive queries are widely used: the user is invited to transform existing modules instead of creating new dialogues from scratch.

All these attempts can help the casual user feel more comfortable in his search.

5.1.3 Other Ways

There exist many other examples of potential research, however, not at all necessarily applicable to data retrieval. The idea of moving a pointer across a screen in order to pin-point or select one item from a menu is certainly beneficial to the casual user. The "mouse" which helps the user in his path finding is also an issue which is well accepted, at least by the manufacturers... There are many other examples of new ideas: the US Air Force, for example, has equipped many of its bombers with devices on which the focusing of the pilot's eyes is translated into targets for its bombs... Thus the pilot's role is simplified to the visual selection from a screen.

5.2 Cost Factor

By building an interface, that is, by adding a new step to the process of retrieving data, a new factor is involved: the

cost factor. By this remark, reference is not only made to the cost of developing an interface, but also, the cost of executing it.

5.2.1 Development Cost

To be considered are the cost of writing the software, building the necessary hardware (like "friendly terminals"), the necessary extra storage (1) and the fact that the communication needs are often somewhat higher, sometimes doubled, since the communication is now not only between the user and the system, but between the user, the interface and the system. Also, by using an interface, many steps risk to be duplicated (i.e. parsing of commands, translation or interpretation of commands and messages, etc.).

Finally, if any "accessories" are used in order to help the user, such as a mouse or graphics, their cost will increase the global price for the user.

5.2.2 Usage Cost

On the other hand, the utilization of an interface can save a good amount of money, if the reader accepts the following

(1) For example, CONIT 3 requires at least 200K of storage.

points.

First, the original idea behind an interface was to eliminate, or at least to reduce the need of the professional searcher: an important savings on the payroll. Also, if the need for formal training can be diminished, and if the casual user can get a substantial satisfaction from the system, the interface can justify its role. It is the opinion of the author of this paper that if a user is satisfied by his search, he will use the search process more and more. By augmenting the usage, it is thought that the per-usage price of using retrieval systems will go down. Finally, it seems probable that, in the future, users of commercial systems will be billed not only for the time spent online, but also, or only, for the "usage time" (CPU time). This mode seems more "fair", and will probably appear in a near future. If, and when, this type of billing is installed, the casual user will get some financial advantages in using "counseling systems" such as IIDA.

5.3 Problems

Finally, the creation of an interface brings some problems which deserve further study. This paper has already noted the difficulties encountered by CONIT, which offers one interface for many classes of systems. An interface designed for one class of

user encounters the same problems, mainly because individual needs are different. An interface should be easily "by-passable". Also, the question of defining "how complete" or "how comprehensive" an interface should be, is a very difficult one to answer. By being absolutely perfect, and covering all cases, an interface runs the risk of being too slow for any user.

Other points worth considering are the facts that there exist some exceptional cases which are very important for a casual user, but so far have not been covered by any of the studies researched in this paper. Thus, if, during a search, a system failure occurs, it is likely that the casual user will be at a loss. The naive searcher is, without any doubts, going to think he did something wrong. And now what to do? That is a kind of traumatic experience which is worth being cured, before the user turns his back away from the retrieval process.

6. SUMMARY

This paper has studied some options available, in order to let casual users access multiple online retrieval systems with minimum difficulties.

One of the advantages of such an interface is that experimental research can (easily) be implemented at the

interface level. It is even possible to push the idea one step further, and recognize that modifications and enhancements of a retrieval system can be studied using the interface. The changes could be observed and tested on the interface as long as needed, and implemented on the main system only when fully satisfactory.

Users are very different in their needs, their behavior and their intellectual capacities, but the interface must be able to handle all of them. Thus:

- (1) The interface should be prepared to cope with any type of mistakes or any possible succession of mistakes.
- (2) Help should be available when necessary in precise and brief displays.
- (3) Users should be able to "fall back on their feet" after any important decision. For example, the system should come back to ask for confirmation before accepting any drastic changes.

It is important to keep in mind that users are human beings, and their expectations and their reactions should be well understood. For example, if the system is perceived once as unfriendly or difficult to use, it is very unlikely that the user

will come back to it, unless obliged to do so. As Lancaster puts it: 'It is all too easy for the inexperienced user to become frustrated, and the once-frustrated user tends not to return to the system' [Lancaster, 72].

It is the problem that the 1980's are facing: the availability of technology is a great thing, if, and only if, it can be used by the people who need it. And that is a challenge which must be met with no delay.

REFERENCES

- [Bayer, et al, 81]. A.E. Bayer and G. Jahoda, "Effects of Online Bibliographic Searching on Scientists' Information Style," Online Review, vol. 5, no. 4, pp. 323-330.
- [Buell, 81]. D.A Buell, "A General Model of Query Processing in Information Retrieval Systems," Information Processing & Management, vol. 17, no. 5, pp. 246-262, 1981.
- [Burns, 81]. C. Burns, "Information Storage and Display," Journal of the American Society for Information Science, pp. 141-147, July 1981.
- [Caruso, et al, 77]. D.E. Caruso and J. Griffiths, "A TRAINER for Online Systems," Online, vol. 2, no. 4, pp. 1-34, October 1977.
- [Caruso, 78]. D.E. Caruso, "Hands on Online: Bringing It Home," Online Review, vol. 2, no. 3, pp. 251-268, September 1978.
- [Caruso, 81]. D.E. Caruso, "TRAINER," Online, vol. 5, no. 1, pp. 36-38, January 1981.
- [Chapman, 81]. J.L. Chapman, "A State Transition Analysis of Online Information-Seeking Behavior," Journal of the American Society for Information Science, pp. 325-333, September 1981.
- [Clayton, 81]. A. Clayton, "Factors Affecting Future Online Services," Online Review, vol. 5, no. 4, pp. 287-300, 1981.
- [Cuff, 80]. R.N. Cuff, "On Casual Users," International Journal of Man-Machine Studies, vol. 12, pp. 163-187, 1980.
- [Dominick, 76]. W.D. Dominick, "The User Interface to Numeric Data Bases," Proceedings ASIS, vol. 13, p. 70, 1976.
- [Doszkocs, et al, 79]. T.E. Doszkocs and B.A. Rapp, "Searching MEDLINE in English: A Prototype User Interface with Natural Language Query, Ranked Output, and Relevance Feedback," Proceedings of the 42nd Annual Meeting of the American Society for Information Science, vol. 16, pp. 131-134, 1979.
- [Goldstein, et al, 78]. C.M. Goldstein and W.H. Ford, "The User Cordial Interface," Online Review, vol. 2, no. 3, pp. 269-275, 1978.

- [Grignetti, et al, 75]. M.C. Grignetti, C. Hausmann and L. Gould, "An "Intelligent" On-line Assistant and Tutor -NLS-SCHOLAR," AFIPS Conference Proceedings 1975 National Computer Conference, pp. 775-781, May 1-22, 1975 Anaheim, CA.
- [Harris, 77]. L.R. Harris, "User Oriented Data Base Query with the ROBOT Natural Language Query System," International Journal of Man-Machine Studies, vol. 9, pp. 697-713, 1977.
- [Horowitz, et al, 81]. G.L. Horowitz and H.L. Bleich, "Paperchase: A Computer Program to Search the Medical Literature," New England Journal of Medicine, vol. 16, pp. 924-929, 15 October 1981.
- [Kennedy, 74]. T.C.S. Kennedy, "The Design of Interactive Procedures for Man-Machine Communication," International Journal of Man-Machine Studies, vol. 6, pp. 309-344, 1974.
- [Lancaster, 72]. F.W. Lancaster, "Evaluating the Effectiveness of Online Natural Language Retrieval System," Information Storage and Retrieval, vol. 8, pp. 223-245 October 1972.
- [Landsberg, et al, 80]. M.K. Landsberg, B. Lawrence, P.A. Lorenz and C.T. Meadow, "A Joint Industrial Academic Experiment: An Evaluation of the IIDA System," Proceedings of the 43d ASIS, vol 17, pp. 406-408, 1980.
- [Larson, et al, 80]. S.E. Larson and M.E. Williams, "Computer Assisted Legal Research," Annual Review of Information Science and Technology, vol. 15, pp. 251-286, 1980.
- [Laurence, 78]. R. Laurence, "Introducing Students to LEXIS: A Model Self Teaching Exercise," Law Library Journal, vol. 71, no. 3, pp. 467-470, August 1978.
- [Mann, 75]. W.C. Mann, "Why Things Are So Bad for the Computer-Naive User," AFIPS Conference Proceedings 1975 National Computer Conference, pp. 785-787, May 19-22, 1975 Anaheim, CA.
- [Marcus, et al, 81]. R.S. Marcus and F.J. Reintjes, "A Translating Computer Interface for End-User Operation of Heterogeneous Retrieval Systems I: Design," Journal of the American Society for Information Science, pp. 287-303, July 1981.

- [Marcus, et al, 81]. R.S. Marcus and F.J. Reintjes, "A Translating Computer Interface for End-User Operation of Heterogeneous Retrieval Systems II: Evaluation," Journal of the American Society for Information Science, pp. 304-316, July 1981.
- [Marcus 82]. R.S. Marcus, "Networking Information Retrieval Systems Using Computer Interface," Proceedings of ASIS, vol. 12, pp. 77-78.
- [Meadow, 79a]. C.T. Meadow, "Some Behavioral Similarities (Or Why the End User Will Eventually Take Over the Terminal)" Online, vol. 3, pp.49-52, January 79.
- [Meadow, 79b]. C.T. Meadow, "The Computer as a Search Intermediary," Online, vol. 3, pp. 54-59, July 79.
- [Miller, et al, 77]. L.A. Miller and J.C. Thomas, Jr., "Behavioral Issues in the Use of Interactive Systems," International Journal of Man-Machine Studies, vol. 9, pp. 509-536, 1977.
- [Oddy 77]. R.N. Oddy, "Information Retrieval Through Man-Machine Dialogue," Journal of Documentation, vol. 33, no. 1, pp. 1-14, March 1977.
- [Penniman, 75]. D.W. Penniman, "Rhythms of Dialogue in Human Computer Conversation," Ph.D. dissertation, The Ohio State University, Columbus, 1975.
- [Pollitt, 81]. A.S. Pollitt, "An Expert System as an Online Search Intermediary," Proceeding of the 1981 International Online Meeting, London, pp. 25-32, Dec 7, 1981.
- [Radhakrishnan, et al, 82]. T. Radhakrishnan, C. Grossner, and M. Benoliel, "Design of an Interactive Data Retrieval System for Casual Users" Information Processing & Management, vol. 18, no. 1, pp. 23-32, 1982.
- [Reisner, 77]. P. Reisner, "Use of Psychological Experimentation as an Aid to Development of a Query Language," IEEE Transactions on Software Engineering, vol. SE-3, no. 3, pp. 218-229, May 1977.
- [Standera, 75]. O. Standera, "Online Retrieval System: Some Observations on the User Systems Interface," Proceedings of the 38th ASIS Meeting, Washington, D.C., 1975.

- [Smith, 76]. L.C. Smith, "Artificial Intelligence in Information Retrieval Systems," Information Processing & Management, vol. 12, pp. 189-222.
- [Toliver, 81]. D. Toliver, "A Program for Machine-Mediated Searching," Information Processing & Management, vol. 17, pp. 61-68, 1981.
- [Verheijen-Voogd, 81]. C. Verheijen-Voogd, "Is EURONET CCL A 'Common' Command Language?" Online Review, vol. 5, no. 5, pp. 399-401, October 1981.
- [Wanger, 79]. J. Wanger. "Education and Training for Online Systems," Annual Review of Information Science and Technology, vol. 14, pp. 219-245, 1979.

4.5

1. Report No. <i>IN-82</i>		2. Government Accession No. <i>183550</i> 447385		3. Recipient's Catalog No.	
4. Title and Subtitle USL/NGT-19-010-900: THE MAN/MACHINE INTERFACE IN INFORMATION RETRIEVAL: PROVIDING ACCESS TO THE CASUAL USER				5. Report Date <i>DATE</i> May 16, 1984 <i>OVERRIDE</i>	
7. Author(s) MARTIN GRANIER				6. Performing Organization Code	
9. Performing Organization Name and Address University of Southwestern Louisiana The Center for Advanced Computer Studies P.O. Box 44330 Lafayette, LA 70504-4330				8. Performing Organization Report No.	
12. Sponsoring Agency Name and Address				10. Work Unit No.	
				11. Contract or Grant No. NGT-19-010-900	
				13. Type of Report and Period Covered FINAL; 07/01/85 - 12/31/87	
				14. Sponsoring Agency Code	
15. Supplementary Notes					
16. Abstract Working paper series report surveying the state-of-the-art in high level man/machine interfaces for supporting casual user access to interactive information storage and retrieval systems. Additionally, capabilities and characteristics of selected specific systems are addressed within the report, including LEXIS, CONIT, IIDA, CITE, and CCL. This report represents one of the 72 attachment reports to the University of Southwestern Louisiana's Final Report on NASA Grant NGT-19-010-900. Accordingly, appropriate care should be taken in using this report out of the context of the full Final Report.					
17. Key Words (Suggested by Author(s)) Man/Machine Interfaces for IS&R, Casual User Interfaces, Information Storage and Retrieval Systems			18. Distribution Statement		
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of Pages 69	22. Price*