379
N81d
No. 2363

A GRAPHICAL, DATABASE-QUERYING INTERFACE
FOR CASUAL, NAIVE COMPUTER USERS

DISSERTATION

Presented to the Graduate Council of the
North Texas State University in Partial
Fulfillment of the Requirements

For the Degree of

DOCTOR OF PHILOSOPHY

By

Clifford G. Burgess, B.A.(Honours), P.G.C.E., M.S.

Denton, Texas

August, 1985

K.M.S.

Burgess, Clifford G., A Graphical, Database-Querying Interface for Casual, Naive Computer Users. Doctor of Philosophy (Computer Science), August, 1985, 89 pp., 1 table, 19 illustrations, bibliography, 51 titles.

This research is concerned with some aspects of the retrieval of information from database systems by casual, naive computer users. A "casual user" is defined as an individual who only wishes to execute queries perhaps once or twice a month, and a "naive user" is someone who has little or no expertise in operating a computer and, more specifically for the purposes of this study, is not practiced at querying a database.

The research initially focuses on a specific group of casual, naive users, namely a group of clinicians, and analyzes their characteristics as they pertain to the retrieval of information from a computer database. The characteristics thus elicited are then used to create the requirements for a database interface that would, potentially, be acceptable to this group. An interface having the desired requirements is then proposed. This interface consists, from a user's perspective, of three basic components. A graphical model gives a picture of the database structure. Windows give the ability to view different areas of the database, physically group together items that come under one logical heading and provide the user with

immediate access to the data item names used by the system. Finally, a natural language query language provides a means of entering a query in a syntax (that of ordinary English) which is familiar to the user.

The graphical model is a logical abstraction of the database. Unlike other database interfaces, it is not constrained by the model (relational, hierarchical, network) underlying the database management system, with the one caveat that the graphical model should not imply any connections which cannot be supported by the management system.

Versions of the interface are implemented on both eight-bit and sixteen-bit microcomputers, and testing is conducted in order to validate the acceptability of the interface and to discover the level of graphical model which the users find most acceptable. The results of this testing are reported and further areas for research suggested.

TABLE OF CONTENTS

	Page
LIST OF TABLES	iii
LIST OF ILLUSTRATIONS	iv
Chapter	
I. INTRODUCTION	1
Introductory Statement	
General Rationale	
Specific Rationale	
Significance of this Study	
II. LITERATURE REVIEW	13
Introduction	
Graphical Database Interfaces	
Database Models	
Query Languages	
Summary	
III. INTERFACE DEVELOPMENT	32
Introduction	
User Characteristics	
Derivation of Requirements	
Graphical Model	
Lower Levels Within the Model	
Query Language	
Implementation	
IV. EVALUATION	52
Introduction	
First Study	
Second Study	
Concurrence & Divergence of the Studies	
V. SUMMARY & DISCUSSION	73
Summary	
Future Research	

	Page
APPENDIX	79
BIBLIOGRAPHY	85

LIST OF TABLES

Table	Page
I. Study Groups	61

LIST OF ILLUSTRATIONS

Figure	Page
1. Factors in Man-Computer Interactions	4
2. Second Study, Medical Network Model with Abstract Relations	18
3. Derivation of Requirements from User Characteristics	35
4. First Study, Top Level Model	40
5. First Study, One Level Down	41
6. First Study, Two Levels Down	41
7. ATN for the Language Analyzer	44
8. Interface Components	45
9. First Study, Ordered Model	55
10. First Study, Scattered Model	55
11. Second Study, Medical Hierarchical Model	59
12. Second Study, Medical Hierarchical Model with Relations	59
13. Second Study, Medical Hierarchical Model with Order of Relations	60
14. Second Study, Medical Hierarchical Model with Abstract Relations	60
15. Second Study, Medical Network Model	62
16. Second Study, Teacher Network Model	62
17. Second Study, Teacher Hierarchical Model	63

Figure		Page
18.	Top Level of Interface Using a Pointing Device	76
19.	Three levels Down in Interface Using a Pointing Device	76

CHAPTER I

INTRODUCTION

Introductory Statement

This dissertation is concerned with some aspects of database interfaces for casual, naive users. "Casual user" is defined as an individual who wishes to execute queries once or twice a month, and a "naive user" is someone who has little or no expertise in operating computers, and, more specifically for the purposes of this study, is not practiced at querying databases.

Over recent years the composition of the computer-user population has changed drastically. Only a decade ago virtually all computer users had some knowledge of computer programming, and it was reasonable to expect some technical expertise of anyone using a computer. Today there are increasing numbers of computer users (18) who neither have, nor desire to have, any technical knowledge of computers, but who only wish to use the computer as a tool--a means to an end. On the other hand, computer hardware is becoming much more sophisticated and supportive of a wide range of programming constructs and techniques. Software today is not as constrained by the hardware as was once the case. It is also true that, whereas computing facilities once were

resident only in large organizations, being centered around one or more mainframe computers, with the attendant support facilities and staff, the current trend is to an increasing number of mini- and micro-computers being used autonomously, or networked, in locations geographically remote from any support facilities (15).

While greater demands are being made of computers than was the case a decade ago, the machines themselves are far more capable of meeting these demands. It is now possible that, instead of a user having to communicate with a computer in a manner dictated by the limitations of the machine, the interaction may now be oriented to the user's perspective and take a form that is wholly acceptable to the user (13), whatever technical expertise he or she may or may not possess.

Within this larger sphere is the area concerned with information storage and retrieval. Vast amounts of factual data are currently held in computer file systems and databases. These systems range from small home, to large public, corporate, and governmental databases. Unless this data is retrievable in an acceptable manner by its intended users, then, to some extent, its worth is lessened since the only reason for storing data is that it will be available for use at a later time. This latter issue is addressed in some measure by the current study, which, in the context of a database retrieval interface, considers aspects pertaining

to the acceptability of the interface by casual, naive users.

General Rationale

The problem addressed by this research is that database interfaces do not appear to be meeting the requirements of casual, naive users (3, 4, 6, 10, 12, 17). This is one facet of a much larger problem, as a number of significant computer systems are extant which are under-utilized by their intended user population. MYCIN (16), for example, is an expert system that has been shown to perform as well as human experts (20) at diagnosing blood disease, but it is virtually unused in any clinical setting.

Database interfaces for casual, naive users are of particular significance at the present time because of the very rapid growth in the numbers of these users at all levels in society, from the home to the executive's office. The enumerating of factors affecting the interaction of a casual, naive user and a computer database may be considered from a number of standpoints (see Figure 1). Purely from the relationship of user to computer, numerous questions arise. Is the initiative to be with the computer or with the user? Will the computer provide acceptable response times? Is there any logon-logoff procedure? Is the program robust (i.e., tolerant of user errors)? Are help facilities provided where necessary? Does the screen inflict

Fig. 1--Factors in man-computer interactions

information overload on the user? Is the computer easily accessible? Is the input device appropriate to the application? Is the input device appropriate to the user (10, 15)?

For a casual, naive user, several questions may be added. Is the learning time short enough to be acceptable? Is the amount of memorization negligible? Does the user perceive a need to employ a computer (3)? When the interaction is to query a database, there are the further, compounding factors of the database model and the query language to be used.

The type of database model also raises certain questions. Is the model easy to comprehend? Can the model be depicted in a meaningful graphical manner? Is the terminology associated with the model easy to assimilate? The query language itself presents a number of variable factors? Is a small set of concepts required in order to get started? Is there a small number of syntax constraints? Is the language consistent? Is the language flexible enough to cover all possible queries? Is the language robust (i.e., can it handle words not in its dictionary)? Is it easy to extend and modify? Is there a small number of exception rules? Can the language processor easily detect errors (14, 15)?

This research does not attempt to analyze all of the possible combinations of the above factors. Rather, it focuses on a specific group of casual, naive users in a

particular situation and concentrates on their characteristics in an attempt to elicit the necessary requirements for a suitable database interface. These requirements are then used as the basis for developing an interface appropriate to the user group. In order to validate the utility of the resulting interface, some testing was undertaken with members of the original user group and with other subjects from groups with similar characteristics. While the main objective of the testing was to assess user comfort with the interface, the opportunity was also taken to measure user efficiency, as indicated both by the time taken to enter a query and the number of mistakes made while entering a query.

Specific Rationale

The research was initiated as a response to the needs of a group of physicians at the University of Texas Health Science Center at Dallas. These physicians have a large research database that is used in attempts to determine trends in illness, to discover best indicators of particular diseases, and to aid in other long-term studies. It is not used to assist in clinical decision making.

At the present time, when a physician has a query he does not enter the query to the database directly; all the database queries are coded and entered by two nurses. By the nature of the pressures of their vocation, the

physicians do not wish to expend the time and energy involved in learning a query language. Moreover, the physicians, typically, do not wish to be personally involved in the use of a computer. Since the nurses have a number of other duties, there is considerable delay in processing a query. Thus it would be much more efficient if a physician could go directly to the database and retrieve the required data.

Significance of this Study

With the increase in the numbers of naive computer users, there has been a growing interest in producing computer systems that can be used by this group of people. In particular, there are today a considerable number of database management systems available, but the amount of research conducted into the many aspects of the user-database interface is not proportionately large. A few systems (e.g., 1, 5, 6, 7, 8, 9, 21, 22) use a graphical display in order to assist in the querying process. However, even with these extant systems, little effort has been made (2) to judge which graphical models are most appropriate for differing classes of users. Thus the current systems rely almost entirely on the preferences of their designers rather than on a solid research foundation. This study aims at initiating a redress of this imbalance by first employing the user's characteristics to define the

requirements for the interface and then going on to provide a graduation of the database model abstraction from systems that display little or no model [e.g., QBE (22)] to those employing a complex network model [e.g., gq1/ER (21)].

It is perhaps apposite at this point to consider the concept of a model. A model, as defined by Webster, is "a description or an analogy used to visualise something" (19, p. 544). A model, therefore, should provide an appropriate representation of some target system (in the current study-- a database). Norman, talking about mental models, notes that "These models provide predictive and explanatory power for understanding the interaction" (11, p. 7). Mental models give an individual a greater understanding of the world about him. A model is appropriate if it is accurate, consistent, and complete. The meaning of these factors will probably vary with the individual; for example, a "complete" model for a casual, naive database user will normally differ from a "complete" model for a database administrator.

In the majority of the database literature, the use of the term "model" refers to the logical data structure used by the database management system to store the data which are in the database. This data structure in many ways defines the database, in that it has a significant impact on the efficiency of data storage and retrieval, and, frequently, on the query language(s) used in conjunction with the database. This view of "model" is, however, not of

concern in this research. The work here is concerned with a computer user's mental model of the structure that holds the data and, furthermore, attempts to influence this mental model by presenting, graphically, on the computer display screen a "picture" or model of the database. This graphical model may be of a completely different type (hierarchical, network, relational) to the database model employed by the database management system.

To summarize, this study is concerned with generating a database interface for a specific group of casual, naive computer users. The main thrust of the work is toward the acceptability of the interface by the target user group, but an ancillary goal is that of improved efficiency. This latter goal will be attained if the interface is acceptable to the casual, naive users as their interaction with the computer will, in their situation, improve the efficiency of the data retrieval system.

CHAPTER BIBLIOGRAPHY

1. Burgess, Clifford G., "A Database Interface Aid," Proceedings of the 1984 IEEE Computer Society Workshop on Visual Languages, Hiroshima, Japan, December, 1984, pp. 72-76.
2. Burgess, Clifford G. and Kathleen M. Swigger, "A Database Memory Aid," unpublished paper, North Texas State University, Denton Texas, 1984.
3. Cuff, R. N., "On Casual Users," International Journal of Man-Machine Studies, 12 (February, 1980), 163-187.
4. Hayes, Phil, Eugene Ball, and Raj Reddy, "Breaking the Man-Machine Communications Barrier," Computer, 14 (March, 1981), 19-30.
5. Herot, Christopher F., "Spatial Management of Data," ACM Transactions on Database Systems, 5 (December, 1980), 493-514.
6. Larson, James A. and Jennifer B. Wallick, "An Interface for Novice and Infrequent Database Management System Users," American Federation of Information Processing Societies Conference Proceedings: 1984 National Computer Conference, edited by Dennis J. Frailey, Las Vegas, July, 1984, pp. 523-529.
7. McDonald, Nancy H., "Video Graphic Query Facility," Proceedings of the 2nd International Conference on Databases, edited by Eli Melenbaum and Vincent Stone, Jerusalem, Israel, November, 1982, pp. 205-217.
8. McDonald, Nancy H. and M. Stonebraker, "CUPID: The Friendly Query Language," Proceedings of the ACM Pacific Conference, edited by Jacques Loeckx, San Francisco, April, 1975, pp. 619-623.
9. Michard, A., "Graphical Presentation of Boolean Expressions in a Database Query Language: Design Notes and an Ergonomic Evaluation," Behaviour and Information Technology, 2 (July-September, 1982), 94-97.

10. Nickerson, Raymond S., "Why Interactive Computer Systems are Sometimes not Used by People who Might Benefit from them," International Journal of Man-Machine Studies, 15 (November, 1981), 469-483.
11. Norman, Donald A., "Some Observations on Mental Models," Mental Models, edited by Dedre Gentner and Albert L. Stevens, Hillsdale, Lawrence Erlbaum Associates, 1983, 7-14.
12. Ogden, William C. and Susan R. Brooks, "Query Languages for the Casual User: Exploring the Middle Ground Between Formal and Natural Languages," Human Factors in Computer Systems, ACM CHI'83 Proceedings, edited by Ann Janda, December, 1983, pp. 161-165.
13. Pal, S., "Improving the Human Factors Aspects of Software Products," Computer Bulletin, 39 (March, 1984), 16-18, 19.
14. Reisner, Phyllis, "Use of Psychological Experimentation as an Aid to the Development of a Query Language," IEEE Transactions on Software Engineering, 3 (May, 1977), 218-229.
15. Shneiderman, Ben, "The Future of Interactive Systems and the Emergence of Direct Manipulation," Human Factors and Interactive Computer Systems, edited by Yannis Vassiliou, Norwood, Ablex Publishing Corporation, 1982, pp. 1-27.
16. Shortliffe, E., Computer-Based Medical Consultation: MYCIN, B. V. (North Holland), Elsevier Science Publishers, 1976.
17. Tennant, Harry R., Kenneth M. Ross, and Craig W. Thompson, "Usable Natural Language Interfaces Through Menu-Based Natural Language Understanding," Human Factors in Computer Systems, ACM CHI'83 Proceedings, edited by Ann Janda, Boston, December, 1983, pp. 154-160.
18. Vassiliou, Yannis and Matthias Jarke, "Query Languages--A Taxonomy," Human Factors and Interactive Computer Systems, edited by Yannis Vassiliou, Norwood, Ablex Publishing Corporation, 1982, pp.47-82.
19. Webster's Seventh New Collegiate Dictionary, Springfield, G. and C. Merriman Co., 1963.

20. Yu, V. L., B. Buchanan and E. Shortliffe, "Evaluating the Performance of a Computer-Based Consultant," Computer Programs in Biomedicine, 9 (September, 1979), 95-102.
21. Zhang, Zhi-Qian and Alberto O. Mendelzon, "A Graphical Query Language for Entity-Relationship Databases," Entity-Relationship Approach to Software Engineering, edited by C. G. Davis and others, B. V. (North Holland), Elsevier Science Publishers, 1983, pp. 441-448.
22. Zloof, Moshe M., "Query-by-Example," American Federation of Information Processing Societies Conference Proceedings: 1975 National Computer Conference, edited by Andries Van Dam and John Stankovic, Anaheim, May, 1975, pp. 431-437.

CHAPTER II

LITERATURE REVIEW

Introduction

A large corpus of literature exists that is concerned with the many and varied aspects of human-computer interaction in general and with database querying in particular. Therefore, this review concentrates on the literature relevant to the user population of this research effort and to the interface developed during this work. The sections of this chapter reflect the different sections of the final interface.

Each section addresses both a differing aspect of the final interface and a number of the interface requirements. The first section is concerned with graphical database systems. These systems differ widely both in the mode by which they employ graphics and in the objectives they attempt to attain with their graphics. This initial section considers a selection of systems, highlighting those features or deficiencies that are relevant to the current research. The second section deals with database models, taking cognizance of the divergence between the view of "model" in much of the literature and that of this study. This section then goes on to discuss two database interfaces

that employ graphical models and to consider other issues related to the presentation of these models. The penultimate section first enumerates the two basic categories of query languages with their characteristics. It then itemizes the pros and cons of these two approaches to query specification before finally considering a language that may be thought of as occupying the middle way between the extremes of the two main categories. Hence, this final language combines the desiderata of each of the previous languages while excluding the inexpedient features. Finally, the summary draws together the preceding sections to present a unified whole rather than a collection of disparate parts.

Graphical Database Interfaces

Graphical representations of information are particularly apposite for use with naive users because, as Murch says, "Almost all cognitive manipulations are visual, since people think in images rather than in words" (20, p. 53). The ease with which pictorial (as opposed to written) information is internalized finds emphasis in other places in the literature, notably in the works of Glenn (5) and Reilly-Smith and Roach (23). The literature also contains studies (3, 14, 15) that stress the advantages of presenting the user with a realistic model of the system being used. Marrying these two threads of thought indicates that not

only should casual, naive database users be given a model of the database but also this model would be best presented in a pictorial fashion.

Existing graphical database interfaces can be divided into two categories. One category deals with graphics more as a means for formulating the query per se than in giving a representation of the database itself; the other category both gives the user some "picture" of the whole database and provides a means for data retrieval. Thus systems such as CUPID (17), and Michard's (18) Venn Diagrams are concerned only with employing graphics to ease the formulation of queries. CUPID employs plane figures, hexagons, rectangles, etc., that contain the names or values of the data items (a query is composed of a structure of such figures). Michard's Venn Diagrams give a representation of the query that can be interpreted in terms of Boolean algebra so that queries are formulated in mathematically rigorous terms.

One of the earlier members of the second category is Zloof's (40) Query-by-Example which initially displays a number of table skeletons as a representation of a relational database. The user enters key letters or values into the tables in order to indicate which data items are to be printed and the conditions to be applied in their selection.

An example of other more recent interfaces which display the complete database is the Spatial Database Management System [SDMS] (8). One implementation of SDMS is used

with a database of oil and gas wells in the United States. Initially a map of the U.S.A. is displayed. The user is able to move a cursor around the map by means of a joy stick. When the cursor is over an area of interest, the user twists the joystick, which has the effect of "zooming" in on the chosen area. As the user closes in, more information (within the area of the "zoom") becomes visible. Hence, in the SDMS system, information is retrieved purely by closing in on an area of choice.

The SDMS interfaces are, then, pictures of the actual data of the database, with the level of detail dependant upon the level of "zoom" the user applies at a particular point in time. They do not give the user any aid in building a model of the database structure. Yet another variant on the pictorial presentation of the actual data is the Video Graphic Query Facility [VGQF] (16). The VGQF uses multiple CRTs and video discs to give users video pictures of data items in order to help them find the particular item they require. However, as with SDMS, this system does not attempt to provide any pictorial model of the database structure to the user.

The literature clearly indicates that the use of a graphical interface is particularly appropriate for casual, naive users. A number of systems have attempted to capitalize on the use of computer graphics in one form or another. However, few have followed the indications in the literature

that a model of the database structure would facilitate the man-machine interaction. Rather, these graphical interfaces have been oriented to other uses for the graphical capabilities of the machine.

Database Models

As indicated above, the literature seems to intimate that it is advantageous for a database user to have a model of the system being queried. On the surface, it appears that there are already a number of studies that consider the effect of database models on the querying process (1, 9, 11, 12, 13, 14, 15, 22, 27, 29). There remain, however, some major questions about the appropriateness and ease of use regarding selection and learning of various logical database models. Nevertheless, all these studies are concerned with the complete logical model, which is directly tied to the data structure used to store the data and, generally, how this affects the user's ability to produce queries correctly and efficiently. Although these studies provided some guidelines for the present research, they have no direct relation to this paper's definition of model. This previous work, thus, has little bearing on the current research in which the underlying logical model of the database is largely irrelevant. For example, the display of the first study (2) presented the users with a hierarchical model of the database, while the data were actually held in a

relational database. The model type of the database need have no bearing on the model displayed to the casual user, except that the display should not imply any relations which are not supported by the database. The displayed model of the database is purely to increase the comfort of the casual, naive user with the system. This is very different from the literature cited above, in which the database model is an integral part of the database management system and, frequently, has ramifications on the formulation of database queries.

The literature does contain one or two examples of database interfaces that give the user a graphical model of the database. The interface of Zhang and Mendelzon (39) gives the user a model similar to that of Figure 2,

Fig. 2--Second study, medical network model with abstract relations.

employing headings abstract relation names in diamonds, and the order of relations. The query is constructed by selecting keywords from menus which are displayed on the right-hand-side of the display screen, in an area separated from the graphical display. The graphical model of the database is static throughout the user-computer interaction. In order to view particular data item names under each heading, the user must switch to the menu. As a result, Zhang and Mendelzon's interface does not use the spatial organization of the display to associate a particular set of data items with a specific top level heading. One solution to the spatial inconstancy problem would be to window out headings and allow the user to view the names of data items which lie underneath each heading. Such an approach is used in this particular study.

The interface designed by Larson and Wallick (10) uses a graphical illustration of a syntax diagram as the means by which a user initiates a query. When the user reaches a point in the query which requires the use of the database schema, a graphical model of the database is displayed, and the user can employ a mouse to select the required data item names. Again, the graphical model used in this interface is similar to that of Figure 2. However, Larson and Wallick's interface gives only intermittent views of the graphical model, since its display is subservient to the syntax diagrams.

Unlike Zhang and Mendelzon's (39) design, Larson and Wallick's (10) interface does have a spatial connection between a heading and its associated data items. Associations are made by drawing "bubbles," each of which contains a data item name. Each of the bubbles then has a line joining it to the appropriate heading. However, displaying all data information at all levels and at one time has significant drawbacks, viz., the screen quickly becomes cluttered. This is compounded in the Larson and Wallick (10) interface by the additional concurrent display of all or part of a number of levels of syntax diagram. In their paper, Larson and Wallick attempt to reduce the "cluttering" effect by showing a model with only three headings and six data items. Such a representation would hardly suffice for a practical system. This is in contrast to the implementations of the interface designed during this research, which represent practical systems and use graphical models developed with the co-operation of individuals working in the areas of interest.

The work of Ross and Moran (26) is relevant also to the current research because it underscores once again the need for consistency. A definition of consistency, as it relates to this study, is information on the same topic, presented in the same area of the screen, no matter what the level of abstraction, so that a user will, subconsciously, associate a particular area of the screen with information on a

particular topic. This idea is in marked contrast to Zhang and Mendelzon's (39) design. Ross and Moran demonstrated what they call "reminders" whereby users of a text editor are able to apply learned commands more readily in a setting that is similar to the one in which the original learning took place (i.e., similar type of text) than in a context which is significantly different from the original.

Other advantages to be accrued from the adherence to spatial consistency are discussed by Teitelbaum and Granda (31). Their study is concerned with menu selection, and they compared how fast their subjects were able to find and identify a particular piece of information on the screen, both when its position was constant and when it varied. They found that positional constancy significantly improved user response time. In the first trial block, subjects with positionally constant information performed significantly better than those without positional constancy. They also found that with positionally constant information there was a learning curve. Over a number of trial blocks the disparity between the groups enlarged as those with a positionally constant display were able to build on their previous experience while those with a positionally nonconstant display were not.

"Chunking" is the concept advanced by Miller (19) whereby people decompose a problem or system into a number (seven, plus or minus two) of "chunks." These "chunks"

represent a logical division by the individual of a larger problem into smaller groupings which are then easier for the individual to process.

In the context of the system developed for this study, the "reminders" effect acts as an adjunct to the user's "chunking" in locating the data item names required for a particular query. The "reminders" reinforce the association of information with position.

A differing view of spatial constancy is that users should be able to see the current detailed information which they are viewing within the overall context (4), so that they maintain a grasp of the larger world while investigating a particular area of it. This is not supported by either the Larson and Wallick (10) or Zhang and Mendelzon (39) interfaces. Larson and Wallick only display the graphical model when it is required by the user, and all the information pertaining to it is displayed at one time. There is no materialization discrimination between the differing levels of information. Zhang and Mendelzon do display the overall graphical model continuously but do not give the position of detailed information within that model.

Since the majority of the database literature referring to database models is concerned with the model employed by the database management system, it is of little moment to the current research. The few database interfaces that do attempt to provide a graphical model of the database

structure seem to have overlooked a number of criterion in the literature as to the desirable features of such a model. The interface which is the result of this research attempts, at least partially, to fill this void.

Query Languages

Languages used for querying databases can be considered as forming two groups, those that are procedural and those that are non-procedural. A procedural language may be characterized both by its having a strict syntax and requiring the user to have some knowledge of the database design in order to formulate a query correctly. A non-procedural language, such as natural language, has a freer syntax and is not tied to the underlying database design.

The literature contains a number of studies of procedural languages (6, 7, 22, 24, 25, 28, 30, 33, 34, 35, 36, 37, 39). All of these studies involved testing novices or non-programmers. The studies are mostly concerned with comparing various features of one or more languages and with the ease of learning and memorizing particular languages. The majority of the studies found that a significant investment of time and effort was required to learn the procedural language(s) under investigation.

On the other hand, non-procedural natural languages have been studied from various perspectives. Combining the results of previous studies, Ogden and Brooks (21) and

Turner and others (35) indicate that if the user is allowed to employ unrestricted natural language, while reducing learning time, it can lead to queries outside the scope of the natural language processor.

Both the Ogden and Brooks (21) paper and the Turner and others (35) chapter suggest that a restricted natural language is the most appropriate mode for a query language. Turner references Shneiderman (29) to underline the appropriateness of employing natural language with users who are familiar with the content of a database, i.e., where the users will be constrained in the queries they form by their knowledge of the constraints implicit in the information held by the computer. The utility of this type of restriction is demonstrated in the lunar rocks expert system (38) where it was found that the geologists who queried the system produced a very high rate of acceptable queries, whereas a group of psychologists, who had undergone a short training session on the data held by the system, still had a very low success rate. An alternative approach to a restricted natural language is proposed by Ogden and Brooks (21), who suggest that an analysis of user queries (for a particular system) will reveal a comparatively small number of syntactic constructs being used and the language can be restricted to recognizing only these constructs.

A completely different approach to the use of natural language is that taken by the Texas Instruments (TI)

database interface (32) which allows the user to compose a natural language query by making selections from a number of menus. This type of system can be considered as having a grammar, but the program does not need to parse natural language. When a particular implementation is produced, a tree-like structure can be generated with menu options and the formal syntax of their associated query segment stored at the nodes. This structure is tree-like in that the flow is from parent to child and the flow does not loop. This system diverges from a tree structure because one child may have a number of parents. This "tree" can be traversed by the program during the inputting of a query, and the static information held in the system can be used to produce the desired query. The TI interface requires so many menus on the screen at one time that there is little room for anything else; no attempt is made to give the user any model of the database being queried.

The two major classes of query languages, procedural and non-procedural, were discussed with their possible trade-off of specificity for ease of use. A compromise, restricted natural language, was considered in which ease of use may be maintained but not at the cost of a complete loss of specificity. The various factors involved with the user population of this study would seem to indicate that a restricted natural language should be the language of choice for this interface.

Summary

A graphical interface is an appropriate vehicle for presenting a casual naive computer user with a model of the database structure, while a restricted natural language is an appropriate means for query entry. These are the two major conclusions to be drawn from the review of the literature. They are, further, the major features of the database interface designed during this research effort. Ancillary conclusions relate to some of the more detailed aspects of the major features. That the display should maintain spatial consistency as information at different levels is viewed, and that this information should be partitioned so as to correspond to the user's "chunking", are, perhaps, generalizations of the other conclusions. The interface described in the following chapters attempts to embody these ideas in a consistent, coherent manner.

CHAPTER BIBLIOGRAPHY

1. Brosey, M. and Ben Shneiderman, "Two Experimental Comparisons of Relational and Hierarchical Database Models," International Journal of Man-Machine Studies, 10 (November, 1978), 625-632.
2. Burgess, Clifford G. and Kathleen M. Swigger, "A Database Memory Aid," unpublished paper, North Texas State University, Denton, Texas, 1984.
3. Du Boulay, B., T. O'Shea and J. Monk, The Black Box Inside the Glass Box: Presenting Computing Concepts to Novices, Paper 133, Department of Artificial Intelligence, University of Edinburgh, Edinburgh, Scotland, University of Edinburgh Press, 1980.
4. Foley, James D. and Andries Van Dam, Fundamentals of Interactive Computer Graphics, Reading, Addison-Wesley Publishing Company, 1983.
5. Glenn, Bernice T., "Alphabet and Text in Presentation Graphics," Computer Graphics World, 7 (January, 1984), 10-22.
6. Gould, J. D. and R. N. Ascher, Use of an IQF-Like Query Language by Non-Programmers, IBM Research Report RC 5279, Yorktown Heights, IBM Thomas J. Watson Research Center, February, 1975.
7. Greenblatt, D. and J. Waxman, "A Study of Three Database Query Languages," Databases: Improving Usability and Responsiveness, IEEE Workshop on Databases, edited by Ben Shneiderman, Haifa, Israel, October, 1978, pp. 426-431.
8. Herot, Christopher F., "Spatial Management of Data," ACM Transactions on Database Systems, 5 (December, 1980), 493-514.
9. Hoffer, Jeffrey A., "An Empirical Investigation into Individual Differences in Database Models," Proceedings of the Third International Conference on Information Systems, edited by D. C. Collins, London, England, December, 1982, pp. 153-167.

10. Larson, James A. and Jennifer B. Wallick, "An Interface for Novice and Infrequent Database Management System Users," American Federation of Information Processing Societies Conference Proceedings: 1984 National Computer Conference, edited by Dennis J. Frailey, Las Vegas, July, 1984, pp. 523-529.
11. Lien, Edmund Y., "On the Equivalence of Database Models," Journal of the Association for Computing Machinery, 29 (April, 1982), 333-362.
12. Lochovsky, F. H., "Database Management Systems User Performance," unpublished doctoral dissertation, University of Toronto, Toronto, Ontario, Canada, 1978.
13. Lochovsky, F. H. and D. C. Tsichritzis, "User Performance Considerations in DBMS Selection," Proceedings of ACM SIGMOD, Volume 2, edited by David K. Hsiao, Munchen, Germany, March, 1974, pp. 128-134.
14. Mayer, Richard E., "Different Problem-Solving Competencies Established in Learning Computer Programming With and Without Meaningful Models," Journal of Educational Psychology, 67 (December, 1973), 725-734.
15. _____, "The Psychology of How Novices Learn Computer Programming," ACM Computing Surveys, 13 (March, 1981), 121-141.
16. McDonald, Nancy H., "Video Graphic Query Facility," Proceedings of the 2nd International Conference on Databases, edited by Eli Melenbaum and Vincent Stone, Jerusalem, Israel, November, 1982, pp. 205-217.
17. McDonald, Nancy H. and M. Stonebraker, "CUPID: The Friendly Query Language," Proceedings of the ACM Pacific Conference, edited by Jacques Loeckx, San Francisco, April, 1975, pp. 619-623.
18. Michard, A., "Graphical Presentation of Boolean Expressions in a Database Query Language: Design Notes and an Ergonomic Evaluation," Behaviour and Information Technology, 2 (July-September, 1982), 94-97.

19. Miller, G. A., "The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information," Psychological Review, 63 (March, 1956), 81-97.
20. Murch, Gerald, "Jerry Murch Discusses Perceptual Ergonomics," Computer Graphics World, 7 (April, 1984), 52-56, 119.
21. Ogden, William C. and Susan R. Brooks, "Query Languages for the Casual User: Exploring the Middle Ground Between Formal and Natural Languages," Human Factors in Computer Systems, ACM CHI'83 Proceedings, edited by Ann Janda, Boston, December, 1983, pp. 161-165.
22. Ray, H., "An Empirical Investigation of the Effects of Individual Differences and Data Models on the Ease-of-Use of Database Query Facilities by Casual Users," unpublished doctoral dissertation, North Texas State University, Denton, Texas, 1984.
23. Reilly-Smith, Susan and John W. Roach, "Improved Design for Graphics Displays," IEEE Computer Graphics and Applications, 4 (February, 1984), 42-51.
24. Reisner, Phyllis, Raymond F. Boyce and Donald D. Chamberlin, "Human Factors Evaluation of Two Database Query Languages: Square and Sequel," American Federation of Information Processing Societies Conference Proceedings: 1975 National Computer Conference, edited by Andries Van Dam and John Stankovic, Anaheim, May, 1975, pp. 447-452.
25. Reisner, Phyllis, "Human Factors Studies of Database Query Languages: A Survey and Assessment," ACM Computing Surveys, 13 (March, 1981), pp. 13-31.
26. Ross, Brian H. and Thomas P. Moran, "Reminders and their Effects in Learning a Text Editor," Human Factors in Computer Systems, ACM CHI'83 Proceedings, edited by Ann Janda, Boston, December, 1983, pp. 222-225.
27. Rustin, Randall, editor, Data Models: Data-Structure-Set versus Relational, Baltimore, Association for Computing Machinery Press, May, 1974.
28. Shneiderman, Ben, Software Psychology: Human Factors in Computer and Information Systems, Boston, Little, Brown and Company, 1980.

29. Shneiderman, Ben, "The Future of Interactive Systems and the Emergence of Direct Manipulation," Human Factors and Interactive Computer Systems, edited by Yannis Vassiliou, Norwood, Ablex Publishing Corporation, 1982, pp. 1-27.
30. Small, W. and L. J. Weldon, "The Efficiency of Retrieving Information from Computers Using Natural and Structured Query Languages," Rep. SAI-78-655-WA, Science Applications, 78 (September, 1977), 146-153.
31. Teitelbaum, Richard C. and Richard E. Granda, "The Effects of Positional Constancy on Searching Menus for Information," Human Factors in Computer Systems, ACM CHI'83 Proceedings, edited by Ann Janda, Boston, December, 1983, pp. 150-153.
32. Tennant, Harry R., Kenneth M. Ross and Craig W. Thompson, "Usable Natural Language Interfaces Through Menu-Based Natural Language Understanding," Human Factors in Computer Systems, ACM CHI'83 Proceedings, edited by Ann Janda, Boston, December, 1983, pp.154-160.
33. Thomas, John C., "Psychological Issues in the Design of Database Query Languages," Designing for Human-Computer Communication, edited by Max E. Sime and Margaret J. Coombs, London, England, Academic Press, 1983, pp. 338-361.
34. Thomas, John C. and John D. Gould, "A Psychological Study of Query-by-Example", American Federation of Information Processing Societies Conference Proceedings: 1975 National Computer Conference, edited by Andries Van Dam and John Stankovic, Anaheim, May, 1975, pp. 439-445.
35. Turner, Jon A. and others, "Using Restricted Natural Language for Data Retrieval - A Plan for Field Evaluation," Human Factors and Interactive Computer Systems, edited by Yannis Vassiliou, Norwood, Ablex Publishing Corporation, 1982, pp. 163-190.
36. Welty, Charles, "A Comparison of a Procedural and a Non-Procedural Query Language: Syntactic Metrics and Human Factors," unpublished doctoral dissertation, University of Massachusetts at Amhurst, Amhurst, Massachusetts, 1979.

37. Welty, Charles and David W. Stemple, "Human Factors Comparison of a Procedural and a Non-Procedural Query Language," ACM Transactions on Database Systems, 6 (December, 1981), 626-649.
38. Woods, R. M., R. M. Kaplan and B. Nash-Webber, The Lunar Sciences Natural Language Information System, New York, Coward McCann, June, 1972.
39. Zhang, Zhi-Qian and Alberto O. Mendelzon, "A Graphical Query Language for Entity-Relationship Databases," Entity-Relationship Approach to Software Engineering, edited by C. G. Davis, and others, B. V. (North Holland), Elsevier Science Publishers, 1983, pp.441-448.
40. Zloof, Moshe M., "Query-by-Example," American Federation of Information Processing Societies Conference Proceedings: 1975 National Computer Conference, edited by Andries Van Dam and John Stankovic, Anaheim, May, 1975, pp. 431-437.

CHAPTER III

INTERFACE DEVELOPMENT

Introduction

This chapter traces the development of the interface from the initial enumeration of the user characteristics through to the implementation on microcomputers. It describes the various components of the system and highlights the user-computer interaction.

User Characteristics

In order to obtain an interface that would be acceptable, the characteristics of the specific user group, viz. doctors, were first analyzed. Recalling that the working definition of a casual, naive computer user is someone who has little or no knowledge of computers and may only want to use a system occasionally, then the relevant user characteristics (cf., 7, 8, 10) for the doctors are (a) is intelligent, (b) has computer naivete, (c) has little time to learn, (d) performs retrievals occasionally, (e) has antipathy toward computers, (f) has no model of database, (g) has good knowledge of information in database, and (h) has poor knowledge of data item names.

The first of these characteristics means that certain assumptions may be made that are not necessarily true for

all possible groups of naive users; for example, it is reasonable to expect that the users can read English, have a grasp of the fact that their actions at the keyboard will produce a reaction from the machine, etc. In other words, certain basic assumptions that are usually taken for granted, are in fact true in this case. The next characteristic, computer naivete, carries the implication that very little can be assumed as to the user having realistic expectations of the computer; the user may, not infrequently, have unrealistic expectations. This means that there is an onus on the computer program to ensure that the user is given a reasonable idea of both what is expected from him or her and what the computer is capable of doing. In this particular case, the doctors are, by virtue of their profession, very busy and have very little time available for acquiring new skills, particularly in an area they view as being of questionable benefit to their activities. The fact that a minimum of learning is required further emphasizes the need for the computer program to provide the user with sufficient information to carry on a useful dialogue with the machine. The doctors do not require the database for use in a clinical setting in which they would be accessing it many times each working day. The database holds research information which a particular doctor would need to reference only occasionally, perhaps once or twice a month, when he or she may perform statistical analyses on

selected groups of patients. While many doctors accept computers as a useful tool for technicians and nurses to use; they feel that they, personally, neither want nor need to use computers, no matter what potential benefits are proffered. It is felt that the lack of any explicit model of the database, i.e., representation of the underlying structure between different classes of data held by the database, is significant since other works (1, 4, 5,) indicate that when users have a realistic model of a computer structure, their efficiency in using the computer is enhanced. The doctors have no logical model of the structure holding the data because they have never been given one, either by formal (or informal) teaching or by a database interface. Finally, users have a good knowledge of the type of information held in the database since it is the type of information they are using in the normal execution of their duties. However, they are not conversant with the very specific division of the data within the database nor with the data item names assigned by the data description language (DDL).

Derivation of Requirements

Figure 3 gives an overview of the derivation of particular requirements from the enumerated user characteristics. The characteristics of "has little time to learn" and "performs retrievals occasionally" each force the requirement

Fig. 3--Derivation of requirements from user characteristics

that the interface be self-explanatory. The former means that any system that requires an investment of time in order to be successfully utilized will not be used (as is the case with the doctors and their current system). The latter requirement means that anything learned at one session with a system is likely to be forgotten by the time the system is next used, thus compounding the learning time associated with the system.

Having established the requirement that the system be self-explanatory, the characteristics "is intelligent" and "has computer naivete" then dictate the level of the explanations that are to be provided. The system messages and

instructions may assume an intelligent user but should not assume any knowledge of computers or computer terminology. Any message to the user must be in terms he or she can understand. This last point is of particular importance in view of the characteristic "has antipathy toward computers"; the user needs to feel "at home" with the system. It needs to put the user at ease and be clear in any messages given. Furthermore, it needs to be forgiving of any mistakes the user may make.

As the doctors do not have a logical model of the database, there is the requirement placed on the interface that it should provide a model. The model so provided should be sufficient to give the user an understanding of the logical structure of the database without becoming too complex so that confusion and information overload result.

The penultimate characteristic "has good knowledge of information in the database" carries the implication that the users, in all probability, will already have their own "chunking" of the information (4, 6, 9) and the interface needs to reflect these same divisions of the data as far as is practicable. Obviously, if different individuals within a particular user group employ drastically different "chunking" schemes, then one implementation of the interface cannot accommodate them all but should endeavor to present the most frequently used "chunking" scheme.

The final characteristic is "has poor knowledge of data item names." This means that the interface should be capable of either handling all possible synonyms for all data items and all aggregates of data items (e.g., "name" implies last name, first name, and middle initial) or of providing the user with easy access to the names of data items and aggregates that are recognized by the interface. Therefore, in summary, the requirements for a database interface to meet the needs of users with the characteristics delineated above are easy to learn, provide a database model, accommodate the user's need for "chunking", and naming of data items and aggregates.

Graphical Model

The use of a graphical model of the database not only helps fulfill a number of the requirements but also has other advantages. The graphical model aids in making the interface easy to comprehend since on the initial, surface level, it provides a "picture" of the database. This picture, in some ways, is familiar to the user, since the named headings correspond to the high level "chunking" employed by the user. The lines connecting the headings give expression to the relationships between one heading and another in a simple, clear manner. The graphical model has the advantage that it allows the user to employ his powers of parallel processing and peripheral visibility (9). The

user can take in the whole model at one time. It is not presented one step or section at a time but as a single coherent entity. Although the user may concentrate on a particular section of the graphical model at a particular point in time, nevertheless, because of peripheral visibility, he or she still has the whole model in view. The graphical model is a visuo-spatial structure so that much of the complexity, which would otherwise be inherent in an explanation of the relationships and elements that go to form the structure, is obviated and replaced by something which is easily comprehended by users with some background in the field addressed by the database. The graphical model represents a complex structure in a simple manner.

The graphical model is a model of the database, thus fulfilling this requirement. Although it may not have many features in common with the database schema, it serves to provide the user with a mental model of the database at a level of abstraction appropriate to the user's understanding.

Since any specific user does not work frequently with the database, the graphical model has the further advantage that it is more easily remembered than would be the case with a block of text that purported to accomplish the same task (9). Thus, employing a graphical model helps to make some use of a user's memory. A graphical model, then, addresses the interface requirements for ease of learning,

providing a database model and meeting the user's need for "chunking" at its most abstract level.

Lower Levels Within the Model

While the headings displayed as a part of the top level graphical model provide one degree of reminder as to the information held by the database, a more detailed set of information may also be viewed. At the lower levels, information such as data aggregate names, data item names, data item values give the content of the available information. The elements below the top level can be related to the heading through which they are accessed in a manner consistent with the user's "chunking" (9). The elements at one level will also maintain spatial continuity with the one at the prior higher level from which they were derived by being centered over the higher level element. As previously stated in the literature review, other methods (3, 12) are employed to display the data item names associated with the nodes of a graphical database model. However, both of the prior systems have deficiencies that are not shared by the current display. The Zhang and Mendelzon (12) system loses spatial constancy by not displaying the data item names in the same area of the screen as the associated higher level items. The Larson and Wallick (3) system (as they admit in the concluding section of their paper) quickly leads to an overcrowded display as each data item name requires its own

"bubble." The example graph given by Larson has only six data item names, which is not realistic for a working system.

Neither the Larson and Wallick nor the Zhang and Mendelzon system accommodate more than top level headings and data item names. The system proposed here has other levels of information to aid the user (see Figures 4, 5, 6). Under a top level heading, both data item names and aggregate names may be found. For example, under the heading "Patients" may be found the element "name" and under this may be found "last name", "first name" and "MI," these last three being data item names, while the element "name" is an aggregate. Where necessary, a hierarchy of aggregates could

Fig. 4--First study, top level model

Affective Disorders Unit Database

Press SHIFT and a numeric key to view the items stored under a particular heading. Press SHIFT 0 to back up.

Query: ■

Fig. 5--First study, one level down

Affective Disorders Unit Database

Press SHIFT and a numeric key to view the items stored under a particular heading. Press SHIFT 0 to back up.

Query: ■

Fig. 6--First study, two levels down

be employed. Another level below that of the data item names could be used to give the form (or possibly value or value ranges) of the data item. An example of form for the data item name "date of birth" would be "mm/dd/yy"; an example of the data item values is given in Figure 6 for the data item "RDC" (a psychiatric diagnostic method), while ranges of values would be appropriate for data items such as clinical and psychological test results. The values in these cases could indicate not only the acceptable range but also the range that is considered normal plus perhaps information on specific abnormal values. The lower level elements thus complete the requirements for an interface that accommodates user "chunking" and provides the user with a complete picture of the content of the database.

Query Language

Hayes, Ball, and Reddy say that "One of the most important causes of the man-machine communication barrier is that an interactive computer system typically responds only to commands phrased with total accuracy in a highly restricted artificial language" (2, p. 19). In view of this and the requirement that the interface be self-explanatory, it was determined that a procedural query language was not appropriate in this case. On the other hand the users are given a model of the database, and they have a good knowledge of the information stored by the database. All of these

considerations lead to the conclusion that natural language is the optimal choice in this situation. With natural language, there is no learning investment since the users know the language already and their knowledge of the domain, coupled with the model of the database, should ensure the use of a restricted natural language.

The ideal method for input to the interface would be for the user to talk to the computer. Unfortunately, funds for a speech recognition system were not available. This meant that it was necessary to use the keyboard to type queries. However, with speech recognition in mind, a natural language analyzer was designed that does not rely, in any way, on punctuation.

The design of the language analyzer employed the augmented transision network (ATN) approach [vide Winograd (11) for a full treatment of this methodology]. The ATN produced is shown in Figure 7. With reference to the requirements for the interface, in addition to those intimated above, the ATN is also robust in that it handles words which it has not recognised. This means that users could employ a much wider vocabulary in their queries than would otherwise be the case. The ATN refers to one group of tokens as "Adpreps." These are the group of adverbs and prepositions that appear in conditional clauses. They are grouped under one heading since they both act as modifiers. A simple partition of the ATN may be noted in that, to a large

extent, nodes "A" and "B" are concerned with tokens to be printed, while the remaining nodes are used to process conditional or qualifying clauses. The ATN makes minimal use of words not contained in the graphical model so as to be as flexible as possible and at the same time use only a small amount of computer memory for its dictionary.

Implementation

Each implementation of the interface consists of a suite of programs (see Figure 8). Each suite has utilities to generate graphical models and initialize data files and a main program which is the actual interface.

Fig. 8--Interface components

One utility is used to generate the graphical model to be employed by a particular implementation of the interface. It takes the person who is generating the model through elements of the graphical model: headers, joining lines, word relations, order of relations, abstract relations, data item names. For each of these, values are entered appropriate to the element. These are then added to the displayed model. Where applicable, the user is given the option of moving an element if the position selected by the program is not to the user's satisfaction. Once all this information has been entered, it is saved in a data file. A second utility is used to generate a file holding the name of the file containing the model data and information on the sequencing of the models during testing.

The main program is composed of three modules: the driver module, the display module, and the language analyzer. The driver module first obtains the name of the model data file and the sequence for the models with the current user. It updates and stores the updated information back in the file so that the next test subject will get the next sequence. The graphical model data is then loaded from its file. Once it has completed these initialization routines, the driver's main functions are to supervise user input and to pass control to the display and language analyzer modules as necessary. At the end of a session the driver saves in a data file the text of the user's query,

time taken for the query, and model accesses. The display module has a control routine that first validates the user input since not all commands to the graphical model are valid under all conditions: for example, a "back up" instruction is not valid if the top level model is displayed. If a command is valid then it is passed to either the routine to display the top level model or the routine to display lower level information. These routines access data on the current status of the display and use this, together with the user input, to update the display as desired by the user. Since the display module is separate from the other modules of the program, the display can be changed by the user at any point during the entering of a query without having any effect on the query itself.

The language analyzer is invoked at the end of a query. It parses the query, analyzes it, and then gives the user feedback on its understanding of the query by stating which item names and aggregates are to be printed and the conditions, if any, which are to be used in the selection process. An earlier pre-prototype version of the language analyzer processed the text of the query as the user was entering it. This approach is no longer followed as it was found that it slowed down the response to user input to an unacceptable level, and the overhead necessary for backtracking, should the user back space more than one word, led to a further degradation of system responsiveness.

The current implementations are on an eight-bit Otrona Attache microcomputer and a sixteen-bit Texas Instruments Professional microcomputer. The eight-bit machine was used in the first study, the sixteen-bit in the second. The two versions are essentially the same except for a few minor differences. The Attache has a monochrome (green) screen and highlighting is used with information below the top level to aid visual discrimination. The TI has a color screen; a blue background is used with white for the text characters and yellow for the graphics. The Attache has a standard "qwerty" keyboard; there are no function keys, so the graphical model was accessed by holding down the "shift" key and then pressing one of the numeric keys. An example of accessing lower levels of the model is given in Figures 4, 5 and 6. In the state of Figure 4, pressing "shift 3" gives the state of Figure 5; if "shift 1" is then pressed the state of Figure 6 results. From Figure 6, pressing the "shift 0" would cause the model to back up to Figure 5 and pressing "shift 0" again would restore the model to the state shown in Figure 4. As can be seen from Figure 5 only the access numbers relating to one level are visible at one time. This is in order to save any confusion over which numbers are operable. The TI has function keys and these are used to access the graphical model as this provides a demarcation between those keys used to enter the query and those used to access the graphical model. In both

implementations the only instructions displayed on the screen are those telling the user how to interact with the graphical model.

CHAPTER BIBLIOGRAPHY

1. Du Boulay, B., T. O'Shea, and J. Monk, The Black Box Inside the Glass Box: Presenting Computer Concepts to Novices, Paper 133, Department of Artificial Intelligence, University of Edinburgh, Edinburgh, Scotland, University of Edinburgh Press, 1980.
2. Hayes, Phil, Eugene Ball and Raj Reddy, "Breaking the Man-Machine Communications Barrier," Computer, 14 (March, 1981), 19-30.
3. Larson, James A. and Jennifer B. Wallick, "An Interface for Novice and Infrequent Database Management System Users," American Federation of Information Processing Societies Conference Proceedings: 1984 National Computer Conference, edited by Dennis J. Frailey, Las Vegas, July, 1984, pp. 523-529.
4. Mayer, Richard E., "Different Problem-Solving Competencies Established in Learning Computer Programming With and Without Meaningful Models," Journal of Educational Psychology, 67 (December, 1973), 725-734.
5. _____, "The Psychology of How Novices Learn Computer Programming," ACM Computing Surveys, 13 (March, 1981), 121-141.
6. Miller, G. A., "The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information," Psychological Review, 63 (March, 1956), 81-97.
7. Penniman, W. David, "Modeling and Evaluation of On-Line User Behavior," Proceedings of the 45th American Society for Information Sciences Annual Meeting, edited by Anita K. Jones, Columbus, October, 1982, pp. 231-235.
8. Rohr, Gabriele, "Mental Representations", Presented at the 1984 IEEE Computer Society Workshop on Visual Languages, Hiroshima, Japan, December, 1984.
9. Shneiderman, Ben, Software Psychology: Human Factors in Computer and Information Systems, Boston, Little, Brown and Company, 1980.

10. Winograd, Terry, Language as a Cognitive Process,
Volume I Syntax, Reading, Addison-Wesley
Publishing Company, 1981.
11. Zhang, Zhi-Qian and Alberto O. Mendelzon, "A Graphical
Query Language for Entity-Relationship Databases,"
Entity-Relationship Approach to Software Engi-
neering, edited by C. G. Davis and others, B. V.
(North Holland), Elsevier Science Publishers,
1983, pp. 441-448.

CHAPTER IV

EVALUATION

Introduction

The interface was evaluated with the original user population and two other similar populations in order to ascertain that it did indeed fulfill the requirements of these groups. Since the graphical model can be varied in many ways, the evaluation testing took the opportunity to compare the acceptability of differing models in order that an optimum model could be isolated, or that at least the models preferred by users could be employed in future versions of the interface.

First Study

Aims

The main aim of this study was to obtain feedback from the intended user population on the acceptability of the interface. To this end, the experimental framework was not as rigorous as in the second study but, rather, attempted to be more that of using the interface in a "normal" situation. Subsidiary aims were to compare different graphical models for the amount of help the user felt they gave and to collect data on the time taken to enter a query, the correctness of the queries and the display accesses made, so

that some idea of the efficiency of the interface could be gained.

Experimental Method

The subjects were eleven clinicians of the Affective Disorders Unit of the University of Texas Health Science Center at Dallas. A few days before the actual test, the prospective subjects were each given a short letter (see Appendix) explaining the objectives of the test and informing them that the test would require that they enter a few database queries on a computer. The subjects were asked to prepare about six of their own queries, covering the types of information they would expect to retrieve from the database. This approach was used because it provided a reasonable simulation of the normal user scenario, i.e., perceive an information need, go to the database, endeavour to retrieve the information. The free-format method was selected as apposed to methods that give the subjects a pre-written query (or a "fill-in-the blanks" method to complete a set of data) because it was felt that both the latter methodologies provide subjects with an unrealistic environment. The actual testing thus required the subjects to perform a more natural query task.

The experiment was conducted in an office setting. During the experiment each subject worked alone on their queries. Immediately prior to the experiment, each subject was given a short (five minute) individual introduction to

the system. The actions of the different keys on the keyboard were explained; in particular those used to communicate with the graphical model were differentiated from those used for the actual query. The method used to terminate a query was also emphasized. The subject was then given a pre-written query to enter along with some verbal instructions as to when, in this query, the graphical model should be utilized to confirm item names. The written query was used to help the subjects gain some measure of confidence in the system before they began the experiment. The written query and other information given to the subject are illustrated in the Appendix.

Once subjects had completed the pre-written query, they were free to enter their own queries. As a subject moved from one query to the next, the graphical model displayed on the CRT cycled through the three models of Figures 4, 9 and 10, thus ensuring that each model was given approximately the same exposure. With different subjects the initial model and the order of the models were also varied to obviate any biasing from this source.

Once a subject wished to terminate the experiment he or she merely informed the experimenter that he or she was finished. At this point the subject was asked to complete a questionnaire (see Appendix).

Fig. 9--First study, ordered model

Fig. 10--First study, scattered model

Data Analysis

In order to assist in comparing disparate queries, a time-complexity metric was employed. The metric values were obtained by dividing the time taken for a query by the number of valid display accesses during the query. This provided an equalizing effect for queries of differing length and complexity. As multiple measures (e.g., ranking three different models) were made for each subject, the testing could be viewed as a randomized block design; therefore Friedman's (2, p. 175) non-parametric, analysis of variance by ranks was employed. This statistic has the advantage that it may be applied to data that do not meet the analysis of variance assumptions of normality and homoscedasticity. Kendall's (2, p. 178) correction for tied ranks was used where appropriate. For the subjective rating of the graphical models, the two-tailed hypotheses employed are as follows:

H_0 : The models will be equally helpful;

H_1 : The models will not be equally helpful.

At a significance level of 0.05 the null hypothesis was rejected. In order to determine between which of the models a significant difference existed, a nonparametric multiple comparisons test, paralleling the Newman-Keuls test but using rank sums instead of means, was used (2, pp. 156, 176). This analysis showed that there was a significant difference between each pair of displays; Figure 4 was the

preferred choice while Figure 10 was considered the least helpful.

For the time-complexity metric, the two-tailed hypotheses employed are as follows:

H_0 : The models will be equally efficient;

H_1 : The models will not be equally efficient.

At a significance level of 0.05, the null hypothesis was accepted. There was no significant difference in user efficiency between any two of the three models.

First Study Conclusions

The ratings given to the preferred model were high and seemed to indicate that the subjects found the interface helpful. This was confirmed by the subjects' written and verbal comments. They expressed themselves happy with the system. The time taken to enter queries and the correctness of the queries would also imply that the database is an efficient tool. Although the subjects were not familiar with methods for accessing a computer database, their mean time for entering a query, including display accesses, was less than two minutes. This compares very favorably with their current system which, when accessed by competent users, takes at least fifteen minutes and normally longer. Also, out of the 221 aggregate, data item and data item value references in the queries, only 20 were entered incorrectly. Thus the interface provided a quick, accurate

and acceptable method for casual, naive users to enter their database queries.

Second Study

Aims

Having discovered in the first study that the interface is appropriate for casual, naive users, this study focused on the graphical model to be displayed to the user. To this end, a number of other features of the interface were held constant so as not to confound the main results.

This study aimed to answer the following research questions.

1. Given a hierarchical model of a database, how much information should the graphical model contain in order to ensure maximum comfort to a casual, naive user?
2. Given a network model of a database, how much information should the displayed model contain in order to ensure maximum comfort to a casual, naive user?
3. Is a casual, naive user's efficiency (based on each of time and number of errors) in querying a database affected by the database model displayed to the user?

Experimental Method

This experiment compared models at a number of levels of abstraction and different information content as shown in Figures 9 to 14, where Figure 11 is the preferred model type of the first study. Coupled with the results of the first

Press a Function key to view the items stored under a particular heading.
Press the F12 key to back up.

Query 1 :

Fig. 11--Second study, medical hierarchical model

Press a Function key to view the items stored under a particular heading.
Press the F12 key to back up.

Query 1 :

Fig. 12--Second study, medical hierarchical model with relations.

Press a Function key to view the items stored under a particular heading.
Press the F12 key to back up.

Query 1 :

Fig. 13--Second study, medical hierarchical model with order of relations.

Press a Function key to view the items stored under a particular heading.
Press the F12 key to back up.

Query 1 :

Fig. 14--Second study, medical hierarchical model with abstract relations.

study, this provides a graduation from "no model" to a complex model. In addition to the hierarchical models, the study also considered network models of a database, as is shown in Figures 15 and 16. These models were used with two groups of subjects: medical personnel and teachers. This gave six subject groupings: medical-weak model, medical-strong hierarchical model (Figure 11), medical-strong network model (Figure 15), teacher-weak model, teacher-strong hierarchical model (Figure 17), teacher-strong network model (Figure 16); also see Table I. The weak groups used the range of abstraction illustrated in Figures 9, 10, and 11, and "no model" in which the subjects used a simplified DDL instead of viewing a model displayed on the CRT. The strong groups used the range of abstraction illustrated in Figures 11, 12, 13, and 14.

TABLE I
STUDY GROUPS

Model	Medical Personnel	Teachers
weak model	first study	second study
strong hierarchical model	second study	second study
strong network model	second study	second study

Press a Function key to view the items stored under a particular heading.
Press the F12 key to back up.

Query 1 :

Fig. 15--Second study, medical network model

Press a Function key to view the items stored under a particular heading.
Press the F12 key to back up.

Query 1 :

Fig. 16--Second study, teacher network model

Press a Function key to view the items stored under a particular heading.
Press the F12 key to back up.

Query 1 :

Fig. 17--Second study, teacher hierarchical model

For the no model/DDL mode, the subjects were provided with a number of lists. Each list pertained to one heading and the information stored under that heading. This was easier than using a DDL listing and corresponded to the system used by the Affective Disorders Unit staff when querying the mainframe database.

There were five to twelve subjects in each of the test groups. The subjects were drawn from Texas College of Osteopathic Medicine (TCOM) and Denton area schools. The subjects had had little or no experience querying a database. Testing for the medical personnel was carried out in

an office at TCOM. The teachers were tested in groups, in a computer laboratory, at North Texas State University.

For the different groups each subject used the models described above. A "within groups" approach was followed, with the ordering of the models being varied so that as many combinations as possible could be utilized. As the ordering of the queries and the displayed information was the same within any group, any biasing due to a learning curve should have been obviated by the different sequences in which the models were presented. With the no model/DDL test, the experimenter ensured that a subject used the data element lists only at the correct time, while the computer controlled the model to be displayed at other times.

Each test consisted of a short (five to ten minute) introduction to the experiment for the subject. The subject was first shown a sheet with the four models he or she would be using (see Appendix) and told that the computer would be employing these pictures of the database. For the strong models, a brief explanation of the relations and orders of the relations was given. Then the experimenter went through the instruction sheet (see Appendix) with the subject. After this instructional period, the subject was given eight queries to enter into the computer. The queries were characterized by not using the correct aggregate and data item names as used by the computer, e.g., instead of referring to

patients, a query might refer to clients. Each subject within a particular group used the same list of queries.

The computer monitored the time taken for each query and saved the subject's queries so that they could be checked later for correctness of aggregate and data item names. Once each subject had finished the set of queries, he or she was asked to fill in a questionnaire (see Appendix).

Data Analysis

Since the queries were the same within any group, there was no need to apply any equalizing factor for differing queries as in the first study, thus the time to enter a query was used to gauge efficiency. The same statistical tests were applied within each group of the study as in the first study.

Friedman's statistic applied to the rating of the hierarchical models by doctors indicated that the null hypothesis should be rejected ($\chi^2 = 9.35$, d.f. = 3, $p = 0.05$). The within groups testing revealed that the model of the same type as Figure 12 was better than that of Figure 14 ($q = 3.63$, $p = 0.05$). No other significant differences were found at the 0.05 level. Using the rank sums gives an ordering of the model variants as: Figure 12, Figure 11, Figure 13, and Figure 14.

Friedman's statistic applied to the rating of the network models by doctors indicated that the null hypothesis should be rejected ($\chi^2 = 13.075$, d.f. = 3, $p = 0.05$). The within groups testing revealed that the model of the same type as Figure 11 was better than those of Figure 13 ($q = 3.91$, $p = 0.05$) and Figure 14 ($q = 4.58$, $p = 0.05$). No other significant differences were found at the 0.05 level. Using the rank sums gives an ordering of the model variants as: Figure 11, Figure 12, Figure 13, and Figure 14.

Friedman's statistic applied to the rating of the weak models by teachers indicated that the null hypothesis should be rejected ($\chi^2 = 12.375$, d.f. = 3, $p = 0.05$). The within groups testing revealed that the model of the same type as Figure 9 was better than both the DDL listing ($q = 3.690$, $p = 0.05$) and Figure 10 ($q = 4.025$, $p = 0.05$), while the model of the same type as Figure 11 was better than both the DDL listing ($q = 2.907$, $p = 0.05$) and Figure 10 ($q = 3.242$, $p = 0.05$). No other significant differences were found at the 0.05 level. Using the rank sums gives an ordering of the model variants as Figure 9, Figure 11, DDL, and Figure 10.

Friedman's statistic applied to the rating of the hierarchical models by teachers indicated that the null hypothesis should be rejected ($\chi^2 = 7.370$, d.f. = 3, $p = 0.10$). The within groups testing revealed that the model of the same type as Figure 12 was better than that of Figure 14

($q = 3.745$, $p = 0.10$). No other significant differences were found at the 0.10 level. Using the rank sums gives an ordering of the model variants as: Figure 12, Figure 13, Figure 11, and Figure 14. Friedman's statistic applied to the rating of the network models by teachers indicated that the null hypothesis should be accepted ($\chi^2 = 5.22$, d.f. = 3, $p = 0.10$).

Second Study Conclusions

The results were fairly consistent across the different groups of users tested. For the range of weaker models, Figure 4 was preferred by the doctors and Figures 9 and 11 by the teachers, i.e., the preferred models were those giving the most information. While for the stronger, hierarchical and network models, displays employing the level of Figures 11 and 12 were those favored, i.e., the users opted for the displays that gave the least information. Thus, viewing together the experimental results from both weak and strong sets of graphical models, it appears that the media via is the optimum choice.

The comments from the experimental subjects indicated the reasons for the rejection of the models at either end of the spectrum. The level of model shown in Figure 10 and in the DDL listings were rejected because they gave the user too little information so as not to be sufficiently helpful. The level of model in Figures 13 and 14 by contrast provided

too much information and were perceived as producing a cluttered display. There was some variance between the groups over the middle ground graphical models; this was due to individual differences rather than any major division of opinion. Thus a display using any of these models would probably be acceptable, though the model level of Figure 12 was preferred in the largest number of cases and so must be considered the most helpful level of information for a graphical model.

It is interesting that the doctors using the hierarchical model preferred the level of Figure 12 but those using the network model preferred the level of Figure 11. It may be that as the model structure increases in complexity, users wish to keep other sources of information (relation names, in this case) to a minimum; conversely, for a simple structure, the user is happier with a slightly higher level of other information. It may well be that the best solution to this variation in preference would be to have an interface that allowed the individual to choose which level of graphical model (Figures 9, 11 and 12) the computer should display; in this manner personal preferences would be accommodated.

The lack of a significant result with teachers using the network model is due in part to the small sample size, since of the five subjects in this group, four chose the

model level of Figure 12. Thus even in this case, Figure 12 is indicated as being the preferred choice.

It is noteworthy that the interfaces cited in the literature review as using a graphical model (1, 3) use that model which was most consistently rejected by the subjects in this experiment. This, perhaps more than anything else, underscores the need for more work in this area, so that the needs of the users can be met.

The queries took the subjects approximately four minutes each to enter, twice as long as with the first study. This was probably due more to changes in the experimental method than any other factors. This is also true of the error rate in data aggregate and item names which rose from less than 10 percent in the first study to nearly 20 percent in the second study. These differences in the studies will be discussed in the next section.

Concurrence and Divergence of the Studies

The two studies were somewhat similar, though with a number of notable differences. It is quite apposite at this point to consider what the areas of concurrence and those of divergence indicate--both for the interpretation of the results of these studies and for future work.

As noted in the conclusions for both studies, the group with a weak set of graphical models preferred those that gave the most information. A major difference between the

studies proved to be the fact that in the first study, the clinicians made up and entered their own queries, while in the second study, the subjects needed to re-word queries they were given. The latter is not such a normal operation as the former. Thus despite the fact that both the queries and the display of the first study were more complicated than those of the second study, the queries in this latter study took twice as long to enter and contained twice as many errors. The printed queries seemed to give the subjects of the second study an illusion of security, which those who had to think up their own queries did not possess. As a result, many of the aggregate and data item names were copied directly from the list of printed queries. Hence the form of the experiment would seem to have had a considerable effect on the apparent efficiency of the interface, but it does not seem to have had any effect on the subjects' choice of model types.

The size of the group of subjects performing the experiment at one time also had a detrimental effect on some individuals. In the group of eleven teachers who did the experiment together, two of the subjects merely copied the queries directly from the list on to the computer. This did not happen with any of the other subjects, all of whom were in smaller groups. None of the subjects seemed to have any trouble with the keyboard as a means of data entry, though it was noted that some of the poorer typists favoured a

"keyword" approach rather than entering complete sentences. Neither the shift-numeric key sequence, nor the use of function keys to access the different levels of the graphical model, seemed to have any effect on user performance. If anything, the former method was preferred, since some of the subjects of the second study did press the numeric keys instead of the function keys; they seemed naturally to associate the numbers in the boxes with the numeric keys. Despite the difference between and within the studies, the main findings with regard to the helpfulness of the graphical models and the ease of use of the interface remained constant.

CHAPTER BIBLIOGRAPHY

1. Larson, James A. and Jennifer B. Wallick, "An Interface for Novice and Infrequent Database Management System Users," American Federation of Information Processing Societies Conference Proceedings: 1984 National Computer Conference, edited by Dennis J. Frailey, Las Vegas, July, 1984, pp. 523-529.
2. Zar, Jerrold H., Biostatistical Analysis, Englewood Cliffs, Prentice Hall, 1974.
3. Zhang, Zhi-Qian and Alberto O. Mendelzon, "A Graphical Query Language for Entity-Relationship Databases," Entity-Relationship Approach to Software Engineering, edited by C. G. Davis, and others, B. V. (North Holland), Elsevier Science Publishers, 1983, pp. 441-448.

CHAPTER V

SUMMARY & DISCUSSION

Summary

The database interface designed as a result of this research is appropriate for use by casual, naive computer users. It meets the user's needs by being not only easy to use but also by giving information about the form and content of the database itself. The interface should also effect an improvement in user efficiency and enable users to enter queries quickly and accurately. In addition it should expand the user group by allowing casual, naive users to join this group. For the casual, naive user it provides a means of retrieving data in a manner that requires no learning but which is comprehended quickly and easily.

This work adds to the body of knowledge concerning the display of the logical model of a database; specifically it shows that a database schema representation commonly used among professionals in the database field is not appropriate for use with casual naive users. It further shows that casual, naive users do prefer to have a model of the database and that they feel that this improves the helpfulness of the system they are using.

Future Research

There are a number of areas for future work that can be based on the interface developed as a result of this research. The areas can be split into three sections; those that address accessing the information held in the graphical model, those that address the display characteristics of the graphical model, and those concerned with the input of the actual query.

The first study used the shift and numeric keys to access the model, while the second study used function keys in order to provide a demarcation between those keys used to access the model and those used for entering the query. This will also be the case if a pointing device is used to communicate with the model, but each of the various possible pointing devices offers further levels of variability. For example, a mouse with two or three buttons may result in a different method of integration than would a light pen.

Some work has already been initiated on aspects of the display characteristics, but there is considerable scope for more. The amount of data displayed at one time may be affected by the colors used, so that if all data pertaining to a particular level has its own background and foreground colors, then the amount of data which the user can easily assimilate is increased when compared to the amount when color differentiation is not employed. Further, different color combinations may well give different amounts of

acceptable data displayed. On the other hand it may be that the over use of color is counter productive, as is claimed by Tufte (1).

In addition to studying the different models presented to the user, there is a need to consider the different ways a particular model can be displayed. For example, Figure 4 shows a tree structure, but the normal method for displaying a structure of this sort would be to have the leaf nodes side by side instead of the way they are staggered in the graphical model, wherein the staggering was used to aid the user's visual discrimination.

As was intimated earlier, one alternative means for entering a query would be to use a voice recognition system. It would, perhaps, be instructive also to compare a more procedural language with the natural language system currently in use.

Were a pointing device employed, then there is the possibility of using the graphical model to specify the query itself. If, for example, a two-button mouse were to be used, the interface could resemble Figure 18, with one mouse button designated to indicate movement between levels in the model while the other button could be used to select an item for inclusion in a query. Figure 19 shows the graphical model at a lower level with the user about to select a value for a particular data item.

Fig. 18--Top level of interface using a pointing device

Fig. 19--Three levels down in interface using a pointing device.

The interface presents a flexible research tool since it transcends the bonds between database schema and query language, while allowing a number of frequently dependent factors to be isolated. Since the interface has been designed as a self-contained, front-end processor, it may be used with any database management system. Further, as the interface itself is of a modular design, it would be comparatively straightforward to use part of it with a component of another system; for example, the graphical display could be used with a different query language. Future research may help to refine this tool and bring about a better understanding of the needs of a casual user who is attempting to retrieve information from a computer database.

BIBLIOGRAPHY

1. Verity, John W., "Graphically Speaking with Dr. Edward R. Tufte," Datamation, 31 (April, 1985), 88-92.

APPENDIX

First study, letter to subjects

Dear

Thank you for agreeing to assist with the testing of the prototype database interface aid that we are developing.

I would confirm that a computer will be available for you in the conference room (UTHSCD, 5th floor, Psychiatry), at xx:xx p.m.

As you will know the affective disorders unit is considering using micro computers to hold the data it collects. I am currently working on a program that is designed to make retrieving information as easy as possible for the medical staff. The program you will be using on Friday is a prototype, it will accept queries from you and do some processing of them, it will also save your queries so that they can be analysed and used in the further development of the query processor.

I would be grateful if you could have ideas of maybe six or seven queries that you would be likely to make in the course of your work.

The program will also give you three different 'pictures' of the database to give you information on the types of data held by the database. Your feedback on these will enable us to find out which is the most helpful for you.

Second study, sheet of models used with subjects

Press a Function key to view the items stored under a particular heading.
Press the F12 key to back up.

Query 1 :

Press a Function key to view the items stored under a particular heading.
Press the F12 key to back up.

Query 1 :

Press a Function key to view the items stored under a particular heading.
Press the F12 key to back up.

Query 1 :

Press a Function key to view the items stored under a particular heading.
Press the F12 key to back up.

Query 1 :

Second study, instruction sheet

DATABASE INTERFACE

You will be entering database queries into a computer. The queries are in ordinary English but you will need to check with the display that the data item names you are using are those recognised by the computer.

Example query

What was the alcohol level for Mr Brown?

Use the function keys (labelled 'F1' through 'F12') to view the data item names that come under a particular heading.

e.g.

4 Tests

press F4 to give:

CBC	blood alcohol	sed rate	CAT scan
X-ray	NMR	cell count	arteriogram

You may use the function keys at any point in a query.

You can use upper or lower case characters (or both).

If you make a typing mistake, use the 'BACK SPACE' key to move back and erase the characters.

The text will automatically wrap around at the end of a line but if you prefer, you may use the 'RETURN' key, as on a typewriter.

To end a query, type:

. or ? or !

Second study, queries

QUERIES.

1. Print the results of the sedimentation rate test on Mrs Mary Jones.
2. List all clients with their social security number & the name of their wife or husband.
3. Where is Dr Browns office?
4. Find all patients over 6 feet tall who have undergone chemical therapy.
5. Print the name of the person in charge of the radiography lab, with their phone number.
6. List all patients who had an operation after receiving radiation therapy.
7. Get the names & home telephone numbers of all patients currently receiving chemotherapy.
8. List the medications being used by people who have a major diagnosis of depression.

Second study, questionnaire

QUESTIONNAIRE

Please rate each of the displays

A	very helpful						no help
	1	2	3	4	5	6	7
B	very helpful						no help
	1	2	3	4	5	6	7
C	very helpful						no help
	1	2	3	4	5	6	7
D	very helpful						no help
	1	2	3	4	5	6	7

Can you give any reason(s) for the model(s) you found most helpful?

Do you use a computer:

never	once a month or less	once a week or less	once a day or less	more than once a day
-------	-------------------------	------------------------	-----------------------	-------------------------

Have you ever written a computer program? yes no

Have you ever worked with a database system? yes no

If 'yes', how long ago was the last time, _____

how often at that time, _____

which database system? _____

Thank you for your help.

BIBLIOGRAPHY

Books

- Davis, C. G. and others, editors, Entity-Relationship Approach to Software Engineering, B. V. (North Holland), Elsevier Science Publishers, 1983.
- Foley, James D. and Andries Van Dam, Fundamentals of Interactive Computer Graphics, Reading, Addison-Wesley Publishing Company, 1983.
- Gentner, Dedre and Albert L. Stevens, editors, Mental Models, Hillsdale, Lawrence Erlbaum Associates, 1983.
- Shneiderman, Ben, Software Psychology: Human Factors in Computer and Information Systems, Boston, Little, Brown and Company, 1980.
- Shortliffe, E., Computer-Based Medical Consultation: MYCIN, B. V. (North Holland), Elsevier Science Publishers, 1976.
- Sime, Max E. and Margaret J. Coombs, editors, Designing for Human-Computer Communication, London, England, Academic Press, 1983.
- Vassiliou, Yannis, editor, Human Factors and Interactive Computer Systems, Norwood, Ablex Publishing Corporation, 1982.
- Winograd, Terry, Language as a Cognitive Process, Volume I Syntax, Reading, Addison-Wesley Publishing Company, 1981.
- Woods, R. M., R. M. Kaplan and B. Nash-Webber, The Lunar Sciences Natural Language Information System, New York, Coward McCann, June, 1972.
- Zar, Jerrold H., Biostatistical Analysis, Englewood Cliffs, Prentice Hall, 1974.

Articles

- Brosey, M. and Ben Shneiderman, "Two Experimental Comparisons of Relational and Hierarchical Database Models," International Journal of Man-Machine Studies, 10 (November, 1978), 625-632.
- Cuff, R. N., "On Casual Users," International Journal of Man-Machine Studies, 12 (February, 1980), 163-187.
- Glenn, Bernice T., "Alphabet and Text in Presentation Graphics," Computer Graphics World, 7 (January, 1984), 10-22.
- Hayes, Phil, Eugene Ball and Raj Reddy, "Breaking the Man-Machine Communications Barrier," Computer, 14 (March, 1981), 19-30.
- Herot, Christopher F., "Spatial Management of Data," ACM Transactions on Database Systems, 5 (December, 1980), 493-514.
- Lien, Edmund Y., "On the Equivalence of Database Models," Journal of the Association for Computing Machinery, 29 (April, 1982), 333-362.
- Mayer, Richard E., "Different Problem-Solving Competencies Established in Learning Computer Programming With and Without Meaningful Models," Journal of Educational Psychology, 67 (December, 1973), 725-734.
- _____, "The Psychology of How Novices Learn Computer Programming," ACM Computing Surveys, 13 (March, 1981), 121-141.
- Michard, A., "Graphical Presentation of Boolean Expressions in a Database Query Language: Design Notes and an Ergonomic Evaluation," Behaviour and Information Technology, 2 (July-September, 1982), 94-97.
- Miller, G. A., "The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information," Psychological Review, 63 (March, 1956), 81-97.
- Murch, Gerald, "Jerry Murch Discusses Perceptual Ergonomics," Computer Graphics World, 7 (April, 1984), 52-56, 119.

- Nickerson, Raymond S., "Why Interactive Computer Systems are Sometimes Not Used by People who Might Benefit from Them," International Journal of Man-Machine Studies, 15 (November, 1981), 469-483.
- Pal, S., "Improving the Human Factors Aspects of Software Products," Computer Bulletin, 39 (March, 1984), 16-18, 20.
- Reilly Smith, Susan and John W. Roach, "Improved Design for Graphics Displays," IEEE Computer Graphics and Applications, 4 (February, 1984), 42-51.
- Reisner, Phyllis, "Use of Psychological Experimentation as an Aid to the Development of a Query Language," IEEE Transactions on Software Engineering, 3 (May, 1977), 218-229.
- _____, "Human Factors Studies of Database Query Languages: A Survey and Assessment," ACM Computing Surveys, 13 (March, 1981), 13-31.
- Verity, John W., "Graphically Speaking with Dr. Edward R. Tufte," Datamation, 31 (April, 1985), 88-92.
- Welty, Charles and David W. Stemple, "Human Factors Comparison of a Procedural and a Non-Procedural Query Language," ACM Transactions on Database Systems, 6 (December, 1981), 626-649.
- Yu, V.L., B. Buchanan and E. Shortliffe, "Evaluating the Performance of a Computer-Based Consultant," Computer Programs in Biomedicine, 9 (September, 1979), 95-102.

Reports

- Du Boulay, B., T. O'Shea and J. Monk, The Black Box Inside the Glass Box: Presenting Computer Concepts to Novices, Paper 133, Department of Artificial Intelligence, University of Edinburgh, Edinburgh, Scotland, University of Edinburgh Press, 1980.
- Gould, J. D. and R. N. Ascher, Use of an IQF-Like Query Language by Non-Programmers, IBM Research Report RC 5279, Yorktown Heights, IBM Thomas J. Watson Research Center, February, 1975.

Small, W. and L. J. Weldon, "The Efficiency of Retrieving Information from Computers Using Natural and Structured Query Languages," Report SAI-78-655-WA, Science Applications, 78 (September, 1977), 146-153.

Publications of Learned Organizations

Burgess, Clifford G., "A Database Interface Aid," Proceedings of the 1984 IEEE Computer Society Workshop on Visual Languages, Hiroshima, Japan, December, 1984, pp. 72-76.

Cohen, Gerald S., Proceedings of the 8th Annual Symposium on Computer Applications in Medical Care, Washington, November, 1984.

Collins, D. C., editor, Proceedings of the Third International Conference on Information Systems, London, England, December, 1982.

Frailey, Dennis, J., American Federation of Information Processing Societies Conference Proceedings: 1984 National Computer Conference, Las Vegas, July, 1984.

Hsaio, David K., editor, Proceedings of ACM SIGMOD, Vol. 2, Munchen, Germany, March, 1974.

Janda, Ann, editor, Human Factors in Computer Systems, ACM CHI'83 Proceedings, Boston, December, 1983.

Jones, Anita K., editor, Proceedings of the 45th American Society for Information Sciences Annual Meeting, Columbus, October, 1982.

Loeckx, Jacques, editor, Proceedings of the ACM Pacific Conference, San Francisco, April, 1975.

Melenbaum, Eli and Vincent Stone, editors, Proceedings of the 2nd International Conference on Databases, Jerusalem, Israel, November, 1982.

Rustin, Randall, editor, Data Models: Data-Structure-Set versus Relational, Baltimore, Association for Computing Machinery Press, May, 1974.

Shneiderman, Ben, editor, Databases: Improving Usability and Responsiveness, IEEE Workshop on Databases, Haifa, Israel, October, 1978.

Van Dam, Andries and John Stankovic, editors, American Federation of Information Processing Societies Conference Proceedings: 1975 National Computer Conference, Anaheim, May, 1975.

Encyclopedia Article

"Model," Webster's Seventh New Collegiate Dictionary, Springfield, G. and C. Merriman Company, 1963.

Unpublished Materials

Burgess, Clifford G. and Kathleen M. Swigger, "A Database Memory Aid," unpublished paper, North Texas State University, Denton, Texas, 1984.

Lochovsky, F. H., "Database Management Systems User Performance," unpublished doctoral dissertation, University of Toronto, Toronto, Ontario, Canada, 1978.

Ray, H., "An Empirical Investigation of the Effects of Individual Differences and Data Models on the Ease-of-Use of Database Query Facilities by Casual Users," unpublished doctoral dissertation, North Texas State University, Denton, Texas, 1984.

Rohr, Gabriele, "Mental Representations," unpublished paper presented at the 1984 IEEE Computer Society Workshop on Visual Languages, Hiroshima, Japan, December, 1984.

Welty, Charles, "A Comparison of a Procedural and a Non-Procedural Query Language: Syntactic Metrics and Human Factors," unpublished doctoral dissertation, University of Massachusetts at Amhurst, Amhurst, Massachusetts, 1979.