THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 61, No. 9, November 1982
Printed in U.S.A.

Database Systems:

Human Factors in Data Access

By T. K. LANDAUER, S. T. DUMAIS, L. M. GOMEZ, and G. W. FURNAS

(Manuscript received September 23, 1981)

Database users make choices, form queries, and understand output. Good computer systems must accommodate the ways that humans best accomplish such tasks. Here we review relevant facts and principles from experimental studies of human information processing. We discuss known characteristics of human memory, language use, and problem solving, and suggest ways in which such knowledge can be applied to the design of systems that will render better service.

I. INTRODUCTION

Intended functions of many interesting electronic database systems require humans to ask questions and be satisfied by the answers. This is no easy matter. Even a system that stores and retrieves vast quantities of information with great efficiency can fail utterly to satisfy its human partners. Somehow, information systems must be made to provide what users want, even when the users don't really know, or can't say very clearly. It seems fairly obvious that, to do this, database system design will have to incorporate systematic knowledge about people, as well as knowledge of hardware, algorithms, and data structures.

The knowledge needed about people will be human-factors psychology, but of a somewhat new kind. In the past, machines (even electronic machines, like gun controllers or PBXs) primarily augmented people's perceptual and muscular abilities, and improving person-machine cooperation required psychological optimization of displays and control devices. Electronic databases will augment—serve as prosthetics for—people's memories and problem-solving abilities. The collaboration between people and their machines is going to be at a much deeper level.

We probably should not talk about improving human-computer "interfaces"—instead we should talk about improving human-computer integration. No matter how powerful a system, if the functions it performs are not comprehensible or useful to its users, no amount of tinkering with its screens and command languages will make it acceptable. We cannot just build a complex system, then hope to give it an appropriately fitted surface; we have to go beyond matching faces to something more like a meeting of minds.

Unfortunately, not nearly enough of the needed applied human-factors psychology exists yet. There is some informed opinion and analysis available. References 1 through 4 provide good reviews of psychological issues in database access. But, directly relevant empirical research has only been going on for a few years. There is also a good deal of pertinent general psychological background (methodology, data, and theory) and some preliminary attempts have been made to see how it might be applied.[5–14] We do think that some progress has already been made, by us and by others, in comprehending the nature of the problem and how to attack it. In this paper, we offer a sample of relevant background knowledge, research results, and ideas. First we describe some exemplary problems, some signs and symptoms of the underlying issues in need of exploration. Then we offer a very brief overview of some relevant psychological facts and principles. Next we describe some examples of recently collected data that may improve our understanding of the problems, and some ideas whose pursuit may help to alleviate them. Finally, we offer some technical and hortatory remarks on the methods and strategies that we think will be required for significant progress.

II. A SAMPLE OF SOME PSYCHOLOGICAL PROBLEMS RELATED TO DATABASE SYSTEM USE

Here is a partial list of some kinds of problems that are frequently encountered in the use of current database systems.

1. Novice freakout.
2. Synonymy and polysemy.
3. Unknown or mismatched categories, features, dimensions, and values.
4. Abstractness and complexity of categories, addresses, and specification.
5. Unsatisfactory browsing and traversal.

2.1 Novice freakout

Novice freakout is simply what sometimes happens when an experienced typist or executive first meets the machine. Despite a lot of public speculation, no one really knows whether the source of this problem is plain old-fashioned novelty, or something especially frightening about semi-intelligent computers. We are fairly sure, however, that the diagnosis that "It's because it doesn't speak English" is a symptomatic response to the problem, not a valid analysis of its cause or prescription for its cure.

2.2 Synonymy and polysemy

Humans can, and often do, express the same idea (i.e., describe the same object or category) in many different ways. And, they often use the same word in many different ways. Computer systems do not know the same things as humans and are not as good at using context to disambiguate expressions. Therefore computers need to be much more careful in the use of language. One result of this is that computers and humans are constantly criticizing one another. When humans use relatively unconstrained vocabulary in entering their requests into a system (e.g., in a bibliographic database search), they typically use a greater variety of expressions than the system recognizes.[3] This is the *synonymy* problem. Often, the user and system employ the same words but with different meanings. This is the *polysemy* (or homography) problem.

2.3 Mismatched features, dimensions, and values

If the *features, dimensions,* and *values* by which categories and their members are specified by one partner in the human-computer team are not known to the other, ill results can be expected. Typically, the data in a database are logically arranged for efficient system processing. Unfortunately, there is no guarantee that the partitioning or connections of data objects in the system will correspond to their partitioning or connections in the user's mind. An object that belongs in Category A in the system may be thought of as a member of Category B by the user, or the user may not know to which category to assign it. For example, books on computer science are categorized under "Generalities" in the Dewey Decimal System.

When the Dutch began to set up their version of the British Prestel home information system, they discovered that information could not be found easily using the existing British search tree.[15] In response, the designers convened a group of experts that included information scientists, human factors psychologists, and software experts to improve the first three levels of the menu structure. The improved version was tested by having representative users try to find representative targets. The discouraging result was that, on average, users traversed the first three levels correctly on only about half their attempts. Wrong decisions seemed to be primarily the result of inaccurate category names and overlapping categories.

We call this the Yellow Pages problem. The Yellow Pages provide a systematic and logical partitioning of businesses and a certain amount of useful cross-referencing. Nonetheless, it is not always easy from the user's standpoint to find a vendor appropriate to a desired item. For example, if you look for copper sheets, you will not find them listed as such. While many reasonable guesses as to the category of business that might handle them (e.g., hardware stores, building supply stores, sheet metal shops, heating contractors, roofing contractors) are listed, they may all be wrong.

2.4 Abstract and complex definitions

We properly think of abstraction and careful symbolic notation as vital implements of precise thought. We view declarative descriptions with well-defined terms and operators as elegant and powerful ways of specifying subsets of data. But the ways of thinking (and talking) that are involved in using abstract symbolic notation are utterly foreign to most people, and immensely difficult to learn. Computer scientists enthusiastically employ these tools in their thought and work, because it makes their business possible. Moreover, most of them are highly selected individuals who got where they are by being good at abstract logical thought from an early age. It is very hard for such people to realize how different they are from most other people. As a result, they often design systems that are powerful for them to use but difficult for others.

2.5 Browsing and traversal

People often complain that automated database systems do not give them the ability to comfortably "look around" for something they might be interested in, or to try to find something that they cannot specify exactly but know they would recognize if they found it. Here the problem is probably one of a mismatch in the organization of data structures, of links between one piece of information and another. The system ordinarily has a sufficient, and perhaps even powerful method of ensuring complete traversal. But it may not correspond at all well to the thinking of the human user. Its field identifiers, arc labels, key patterns, relations, etc. may have no counterpart in the user's mind. In addition, the scope and focus of search may be nonoptimal or too rigid. For example, users may be provided only with a small amount of detailed information at any one time, when the option of a more global information view might be more useful.

While these are only some of the problems people have in using automated data-access systems, they should suffice to show the seriousness and depth of the problems. What is most important to notice about all of these problems is that they could never be seen or measured on a system without a user. *They are not properties of a computer system but of a computer-human system taken as a whole.* To understand and work towards the solution of these problems we will have to know some important properties of users: how they remember, name, classify, organize, search, represent, and process information. All of these topics are parts of the field of *cognitive psychology.*

III. OVERVIEW OF COGNITIVE PSYCHOLOGY

Cognitive psychology is a subfield of human experimental psychology that has developed over the last two decades to deal with thought, perception, and memory. It overlaps and draws extensively on linguistics, philosophy, artificial intelligence, and neuropsychology. Although it has developed primarily as an academic pure science, much of its content offers a useful basis for understanding problems of person-computer integration. The central method of cognitive psychology has been to treat the human mind as a general information processing system. The extension of this approach to the study of human-machine partnerships is quite natural. In this section we highlight portions of this field that are especially relevant to the data-access problem.

Human memory is, of course, enormous and marvelous. No good method exists for estimating its true information capacity in a really satisfactory way. The estimates one reads of 10^{13} bits are based on approximate anatomical numbers combined with some stunningly gratuitous assumptions about the relations of anatomy to memory. Nevertheless, observable facts about its performance are sufficient to demonstrate that the capacity and capabilities of human memory are impressive by current computer standards. The average undergraduate at a highly selective college knows something about the meanings of about 100,000 different words. Each word is a complex and variable sound or light pattern with an associated complex set of knowledge about meanings and relations to other words and objects. Most of these words are capable of starting a person through a huge network of associated facts. Consider what you know relevant to the word "dog". Dogs have four legs; some are brown and/or have tails; they are related to wolves; your aunt Jane has one named Spot who bit a postman named Dick; some are movie stars but none are presidents; some pull sleds, have rabies, eat Kennel Ration, etc.

Each of one's 100,000 known words constitutes a category, i.e., the set of all the things to which it might refer. And there are many unnamed categories in memory as well. Human categories come in many varieties. They may be defined quite arbitrarily—essentially by enumeration—or by heterogeneous family resemblances one to another, or by a common resemblance to one or more prototypes. Mental categories may also be rule-defined, and if so by a variety of kinds of rules; consider coins, relatives, fruits, and Chinese food. It is rare that the categories people spontaneously use are precisely defined; it is much more common that they are fuzzy both in the sense that one category overlaps with many others and that many elements in any one category are only probabilistically known to be contained in it. What is known about each category, about the relations among categories, and about each member of a category is always variable both between one person and another and between one time and another for the same person. The reason is simply that human knowledge is acquired continuously from the world by a large amount of highly idiosyncratic and somewhat haphazard experiences.

Another interesting fact about human mental or linguistic categories is that they are often highly context-dependent and flexible: what is "big" depends on whether one is talking about buildings or insects.

The relations among categories are also less orderly, and generally more flexible than the ones we would like to use for artificial information systems. There is some hierarchical, nested organization, but not a lot. If people are presented with an extremely common object like an apple or car, they will easily give it an appropriate name and there will be a fair amount of agreement between any two people on what to call it.[16] But if one asks for the superordinate of a common object, e.g., "What is the superordinate of car?", people are usually able to give an answer, say "vehicle," but there is much less agreement on the appropriate class name. If they are then asked for superordinate categories for the superordinate they may not even be able to give a sensible answer, and the consistency between people in what they nominate becomes very low. The same is true for subordination. A subset of apple may be MacIntosh, but what is a subset of MacIntosh? There are possibilities, of course, but they tend not to be compelling or unique, and will generate little agreement between different people. The organization of the information stored in human memory appears to be fairly flat; there are at most a handful of levels, and the hierarchical structure that does exist is not identical for every person. Rather the mind appears to contain a multitude of fairly specific connections between one concept and others.

Our ability to recognize things and words, that is, to place them in at least one known category and, still more, our ability to associate other information with every fact, is quite remarkable. But we are very poor at listing all of the members of a large class, any one of which we could quite easily recognize as such. For example, the name of every state in the United States is easily identified as such, but most Americans are unable to recall them all in fifteen minutes.

Another thing people are naturally very bad at is formal logic. Incorrect simple syllogisms like, "all women are mortal, Sally is mortal;

therefore, Sally is a woman," are judged correct by a very large portion of the population. If the statements or premises are stated in very abstract form, or if their content is familiar but leads towards a different common sense conclusion than that implied by the statement itself, people are particularly error prone.[17] Most people are also poor at the skills required to represent problems as mathematical or logical abstractions. For example, given the word problem, "At Mindy's restaurant, for every four people who order cheesecake, there are five people who order strudel." Then, of those people who can form an equation at all, the majority will erroneously write down "4c = 5s." [18]

In making inferences from available evidence, people are guilty of a wide range of failures, most notably a tendency to grossly overvalue the most easily remembered information.[19] Even simple arithmetic is a difficult matter for the human mind. While the associative memory feats described earlier compare quite well with modern computing techniques, humans can only add about one single digit number per second for a few seconds. Unfortunately, all of these weaknesses in logical and computational abilities are pretty much true of even the most intelligent, well trained people. "Easy" syllogisms are incorrectly judged by college students after taking logic courses; algebra word problems not only contribute to most of the population dropping out of mathematics at an early age, but evoke errors even in mathematics and computer science graduate students.[18] Essentially, only statisticians actually make decisions by applying Bayes' rule. In short, ordinary human beings act as if they had the memory of the largest computer system ever built, but the logical and computational ability of a LISP program running on a pocket calculator!

For present purposes, perhaps the most important principle of all concerning the human mind is this: New knowledge depends intimately, pervasively, and automatically on old knowledge. In a sense, information does not exist until it is coded or categorized, that is, until it is interpreted in terms of and fit into some existing knowledge. Humans, with their enormous capacity for associative memory, interpret new pieces of information in enormously rich and variable ways. The way humans perceive or understand an event is to code it with respect to what they already know. Since what a human already knows is complex, huge, and unique, each experience is interpreted in an extremely complicated and personal manner. The simple use of one's native tongue provides an example. The phrase "my name is Tom" seems quite effortlessly and directly interpretable to an English speaker. The elaborate reference to previous knowledge involved is hidden from conscious awareness. The phrase "mimi yako nani Mbogua" is roughly equivalent in Swahili, but when spoken strikes the Western ear as totally uninterpretable sound.

The coding involved in translating the sound to meaning is almost entirely involuntary and unconscious. In this respect language is not alone; a similarly complex and unconscious process is involved in the interpretation of any meaningful event. The more we know relevant to that event, the more richly it will be interpreted and the more our experience of it will differ from that of other people.

One important moral from all this is that there is simply no way that an unaided, naked-eyed expert can see things the same way as a beginner. This is especially true with regard to the perceived ease of use of complex systems. Almost anything appears easy to do to the person who has learned how to do it well. One person's introspective feeling of ease does not assure that another person will not have difficulty. Certainly it is better to try to put oneself in the user's place than not to do so. Keeping the end-user in mind throughout system design and development, and trying one's best to imagine what it is like to know nothing, certainly can't hurt. But even the best efforts of intuition and empathy often fall far short. All too often, documents beginning "This system provides a simple, easy-to-use and easy-to-learn ···," really mean that the author, after hundreds of hours of use of whatever it is, finds it quite familiar and handy. It is very much like the child who says "English is so easy, why is French so hard?" In truth, only systematic and controlled observations of actual users can lead to accurate appreciation of the novice's or occasional user's perspective.

A final topic from cognitive psychology that we wish to discuss in a little more depth is the psychological structure of human knowledge. How is knowledge represented and organized in the mind of an individual? What we mean by structure is the relation between different elements of knowledge and the resulting constraints on how one can think about them. There are many ways in which knowledge is structured by humans and there are several quite good methods for revealing important aspects of this structure, at least for relatively simple cases.

An example of a means of revealing knowledge structure is proximity scaling.[20] Given items from a domain of knowledge, we wish to represent how people think about the similarity relations among them. Take, for example, knowledge of the spectral colors. We choose a set of pure colors spanning the spectrum, pair them in all possible ways, and ask people to assign a number corresponding to the perceived similarity between members of each pair. We then assume some kind of underlying structural representation (e.g., Euclidean space, hierarchical tree) and attempt to fit the data to it. Typically, this is accomplished by an iterative numerical estimation procedure. In the case of colors, the similarity judgments can be modeled very well by a three-dimensional Euclidean space in which each color is assigned three

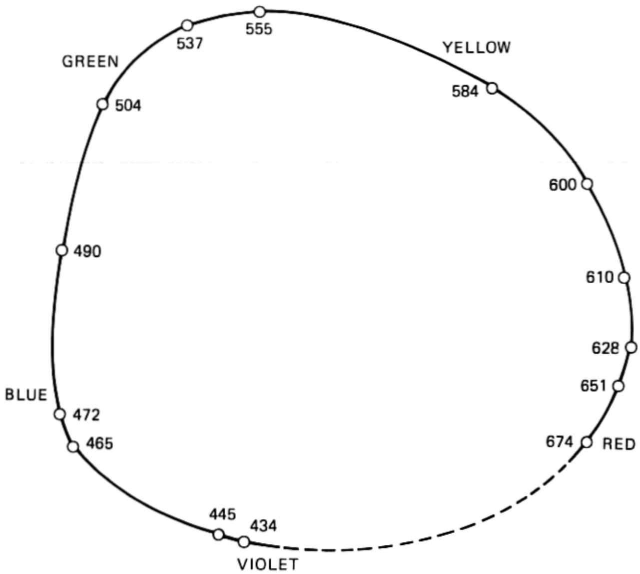

Fig. 1—Two-dimensional spatial representation of psychological similarity of 14 colors.

orthogonal coordinates. Figure 1 shows a commonly used representation of the two dimensions of this space needed to represent the aspect we think of as hue. Distances in this inferred structure quite faithfully predict the judgments of similarity between any two colors differing only in hue. If you are not familiar with this picture, note that the psychological perception of similarities in colors places the physically most different colors (in terms of wavelengths) near each other. One could not have guessed what the structure in the perception would be from a knowledge of the physics of color; one needed to make the right observations on the human receiver of the stimulus.

Actually, the difference between "physical" and "psychological" structure is well worth amplifying. Even in the simple case of color, the bending of the spectrum into a circle is not the only divergence of the two realities. A colored object in the world around us generally reflects a myriad of different wavelengths at various relative intensities. To specify that reflectance would require an infinite list of numbers, one for each wavelength (technically, spectral reflectances form an essentially infinite dimensional vector space). This is by no means what humans perceive. What they do perceive instead of this physical reality has been a topic of psychological study for over a century and is in fact very complicated: e.g., we see "colors" that have no correspondent in the spectrum, we can see blues (or greens or reds) when there is no spectral blue (or green or red) present at all. For our purposes here, the most important psychological aspect of color is that the infinite dimensional space of colored light maps quite well into a perceptual space of only three dimensions—basically the plane in Figure 1, plus a third dimension of brightness. Instead of the infinite string of numbers needed to indicate spectral composition, it takes only three numbers to characterize a perceived color. Some of this research diffused into the popular culture long enough ago that this may seem a trivial, almost physical fact. It is not. It is a psychological reality that is highly nontrivial. Physics could not have discovered this nor the fact that the color world of bees has four dimensions and that of cats only two. The point is that our internal representation of the structure of color, or sound, or objects in the world is not a simple fact about the external world, but is very strongly shaped by our perceptual and psychological mechanisms and experiences. These psychological effects must be studied explicitly, and multidimensional scaling is one tool to do so. The color circle in Figure 1 is just a simple example.

If instead of colors, we have people rate the similarity between pairs of named animals, the results do not fit very well into a Euclidean space. What they do fit in this case is a hierarchical tree structure. If one assumes that the conceptions of animals are arranged in a tree and that the perceived difference between any two is a function of the minimum arc length connecting them, then a structure can be induced that reflects human judgments quite well. Figure 2 shows an example for judgments of sixteen animals. Left-right position is essentially meaningless in this diagram. The up-down distances and connections show the structure and the strength of the relations.

In a recent paper,[21] Pruzansky, Tversky, and Carroll surveyed a large number of perceptual and cognitive domains in which pairwise similarity judgments had been obtained. They found that any particular domain was almost always fit considerably better by one or the other of dimensional or hierarchical models. There are, of course, variants within each of these general types of structure; and there are other kinds of structures as well. Methods are available for fitting some of these others to data. But there is a good deal more work to be done in exploring new kinds of structures, and structures based on relations other than judged similarity. Current methods are also quite cumbersome, if not prohibitive, for very large sets of objects.

IV. APPLICATIONS, IDEAS, AND DATA

In this section, we want to consider how one can use knowledge about people's knowledge to improve data access. Let us start by pursuing our discussion of knowledge structure, which will allow us to develop a simple example of an application.

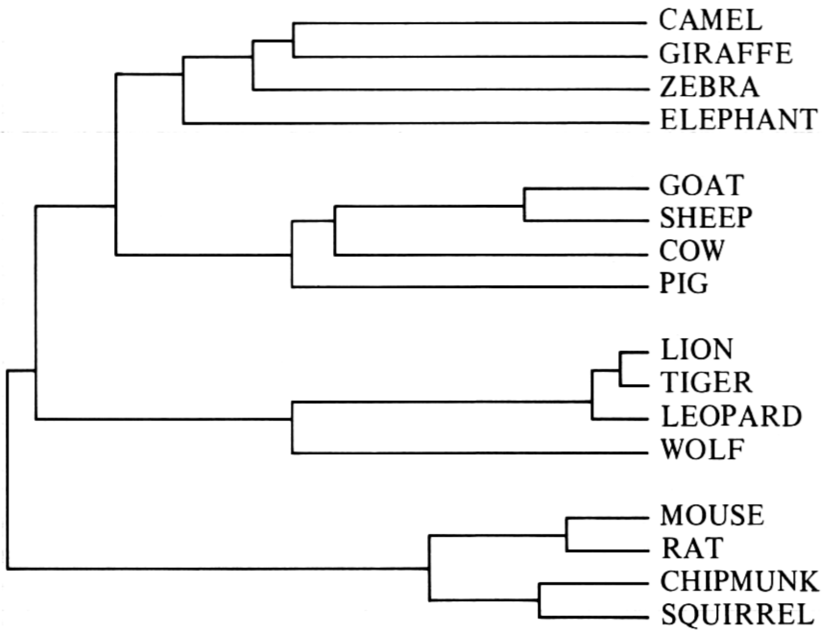

Fig. 2—Hierarchical tree representation of similarity judgments for 16 animals.

4.1 Knowledge structure

In subdividing a set of data for humans to use, it is necessary to respect the psychological structure of that data. Humans will probably prefer to specify and be told about subdivisions and relations that they can already understand. Appropriate and meaningful access paths for the human user will depend on the human user's knowledge structure (which will not necessarily be the most convenient for system design). Imagine a system allowing a textile dyer to retrieve the chemical formula for a dye of any desired color. What we know about the continuous, three-dimensional structure of psychological "color space" argues strongly for an interface device with three analog controls, e.g., three knobs that could be continuously manipulated with the resultant color displayed on a screen. A tree-structured menu, while not impossible, is likely to be inferior, because a tree just does not do justice to the psychology of the domain: it requires the artificial introduction of successive orthogonal partitions into a space that is psychologically continuous and of low dimensionality. On the other hand, suppose one built an information device for zoo visitors, in which they could specify a particular animal and obtain information on a computer terminal. Here the tables would be turned: a menu tree should have a strong advantage. There are more diverse things that must be specified about an animal (and about some sets of animals but not others) than can be captured in two or three continuous parameters. In dimensional representations the aspects of variation are equally well defined throughout the space (bright/dark variation makes sense regardless of hue). This requirement is repeatedly violated with animals, e.g., the differences among fish are typically irrelevant to mammals. Tree structures, on the other hand, are very well suited for diverging classification criteria, so that a hierarchical menu system, with sequential choices, first among broad classes, then among subclasses, and so forth, would be superior because it better matches how people think about the domain about which they are trying to retrieve information.

What about other structures and other differences between structures? The strict hierarchical tree structure is represented, and probably well served, by typical menu selection systems. A strict hierarchical menu, however, assumes a logical partitioning of the domain at each step (i.e., each object belongs to one and only one subset). But what if, as is frequently the case, people's categories overlap? The structures that one can use to represent overlapping categories are necessarily not as neat and search-efficient. The matter at issue is whether there are ways to accommodate the complexity of people's minds and still help them retrieve information in powerful ways.

One approach is to consider alternative knowledge structures that can incorporate important features of human representation but still form the basis of systematic search procedures. Furnas has taken this approach to the problem of category overlap and multiple superordinates.[22] Given an object like a house, a person can think of it as a dwelling, an investment, a tax entity, a subclass of monthly expenses, etc. All are different, somewhat overlapping, superordinate categories. How can one build a retrieval system that captures this richness? A typical, strictly hierarchical menu system is inadequate, since it allows only one superordinate for each node. A keyword access system would be representationally adequate, but has the drawback (especially serious for novice users) that it provides no guidance about what is in the database. Users must spontaneously generate keywords in efforts to second-guess a possibly unfamiliar set of system keywords necessary to reach a target. A long history of psychological research points to the greater difficulty of this spontaneous generation procedure when compared to the mere comprehension involved in menu selection. (Recognition/recall differences in the literature on learning date at least to 1904.[23]) Another possibility, then, is to use a menu-driven retrieval system, but allow connections to form less constrained structures than trees. Consider arbitrary general graphs, for example. These too are representationally adequate, but again at the expense of a virtue which trees possess. Trees allow a user to begin with high-level choices, thereby eliminating large classes of descendants from any further consideration. An unrestricted graph structure provides no such guarantee of successive narrowing of the search. Without any top-down strategy available, retrieval could conceivably require an exhaustive traversal to search the graph, with the concomitant tedium and memory taxing record keeping (i.e., "Have I looked here before?"). Moreover, the number of choices offered at any node might be large, thus frequently overtaxing people's limited short-term memory and decision capabilities. Thus, one might want to represent information in some graph structure with intermediate restrictiveness. Furnas has explored directed acyclic graphs, where links are conceptually "directed", connecting superordinates to their subordinates. This arrangement allows the representation of overlapping subsets and multiple superordinates, and thus offers one plausible compromise between complete representation and manageable organization.

Furnas has implemented these as a menu retrieval system in which at each node the user has a choice of both multiple subsets and multiple supersets. Like more familiar tree-structured systems it is well suited for top-down, general to specific, retrieval (with no risk of getting caught in cycles in the refinement process). Its greater generality allows a richer variety of interaction, however. An example is given in Panel A of Figure 3, for a system that allows access to a recipe file. In a typical menu access scheme, as illustrated in Panel A, the user would arrive at a node for salads and be given several choices of different kinds. If none was satisfactory, the only choice would be to retreat up the one link from salads to the superordinate from which it was originally reached, say "early courses." But the user might not really be interested in early courses any more, but rather in salads as a kind of cold food. In that case a different superordinate would be more useful, as illustrated in Panel B. Still another possible superordinate for salad is shown in Panel C. What the experimental system does is to offer a reasonable set of upward (as well as downward) choices at each node. Such a scheme retains a good deal of the constraint of a menu system, thus providing guidance for the user but is presumably less unnaturally restrictive. It is one example of a novel access method that is motivated by concern for the human user's representation of the knowledge. There are a number of problems to be resolved in learning how to construct and use such a structure effectively. A particularly important issue is how to obtain the information about user knowledge that will allow one to choose optimum superordinates.

4.2 Symbolic codes

For the next example, consider the problem of learning symbolic

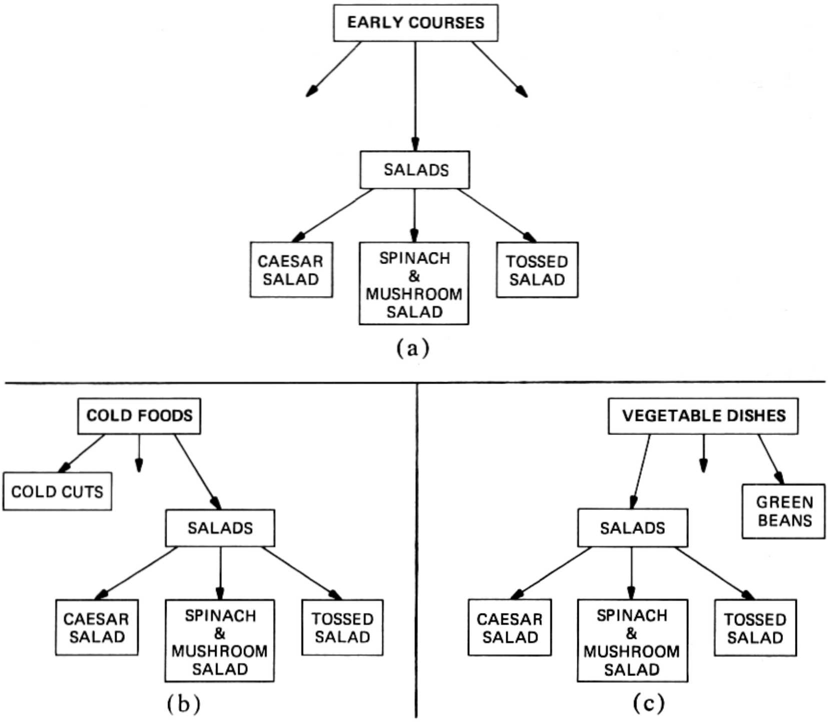

Fig. 3—Illustrations of a menu access scheme with multiple superordinates.

codes to stand for objects in a database. Shorthand abbreviations or codes are not necessarily evil, nor are they new annoyances occasioned by the introduction of computer systems. Almost anyone who keeps large numbers of records very soon falls into the habit of making short codes to stand for otherwise lengthy entries. However, the advent of electronic devices has exacerbated the problem simply because more people are entering more different things. For example, the repair-installation personnel in some telephone companies now have to make entries that require as many as seven different coding schemes, one of which has many thousands of different codes. It seems quite likely that frequent users of database systems will always want to refer to entities by short, symbolic codes. It is one thing to say that these should be mnemonic, but something else to know exactly what makes something easy to remember and use for such a purpose.

A principle governing at least one aspect of the problem comes from a theorem derived by Shepard from a model of how errors are generated when humans learn to associate a set of symbols to a set of objects.[24] The minimum number of errors, and presumably the fastest learning, is shown to result from a mapping in which labels that are perceived as similar (and therefore likely to be confused with each other) are assigned to objects that are also perceived as similar. Briefly, the reason is that an error generated by confusing two codes will sometimes be compensated for by errors in selecting the objects to which they apply; whereas other mappings will preserve all errors. We demonstrated a possible application of this conjecture in the following way. From previous work we knew that animal names correspond well to the structure shown in Figure 2.[25] We also knew from previous work that digits and letters of the alphabet are more similar the closer they are in lexicographical value.[26, 27] We constructed 16 alphanumeric codes and assigned them to 16 animal names, as shown in Fig. 4.[28]

	(A)	(B)
LEOPARD	F5	D4
CAMEL	B9	A1
ELEPHANT	D4	A9
WOLF	A1	D7
GOAT	F1	B1
TIGER	B6	D2
RAT	F7	F2
SQUIRREL	A3	F7
ZEBRA	D2	A6
SHEEP	A9	B3
COW	F2	B6
CHIPMUNK	D1	F5
MOUSE	B3	F1
LION	A6	D1
PIG	B1	B9
GIRAFFE	D7	A3

Fig. 4—Assignments of alphanumeric codes to 16 animal names. Column A shows a random assignment, and column B shows an assignment based on the similarity relations of Fig. 2.

For one group of subjects the assignment was made at random, as shown in column A. For another group of subjects the assignment was such that codes that are perceived as like each other were assigned to animals that are like each other according to the structure of Fig. 2. These are given in column B. It is probably not obvious from Fig. 4 that the mappings in the two columns are systematically different or that one would be easier to learn than the other. However, columns A and B of Fig. 5 show the lack of correspondence and the correspondence, respectively, of the underlying mappings to the structure of Fig. 2. (Note that subjects were not shown anything like Fig. 2 or Fig. 5, column B. They were just shown sequences of name-code pairs in random order, as in Fig. 4.)

The mapping based on Shepard's theorem produced almost 50-percent faster learning than did the random mapping. You may say to yourself something like, "But of course, the systematic mappings make sense and the others do not." This is the point. Also important is the fact that the relations among symbols were arrived at in a systematic, mechanical way that might be applied in places where "what makes sense" is not so obvious.

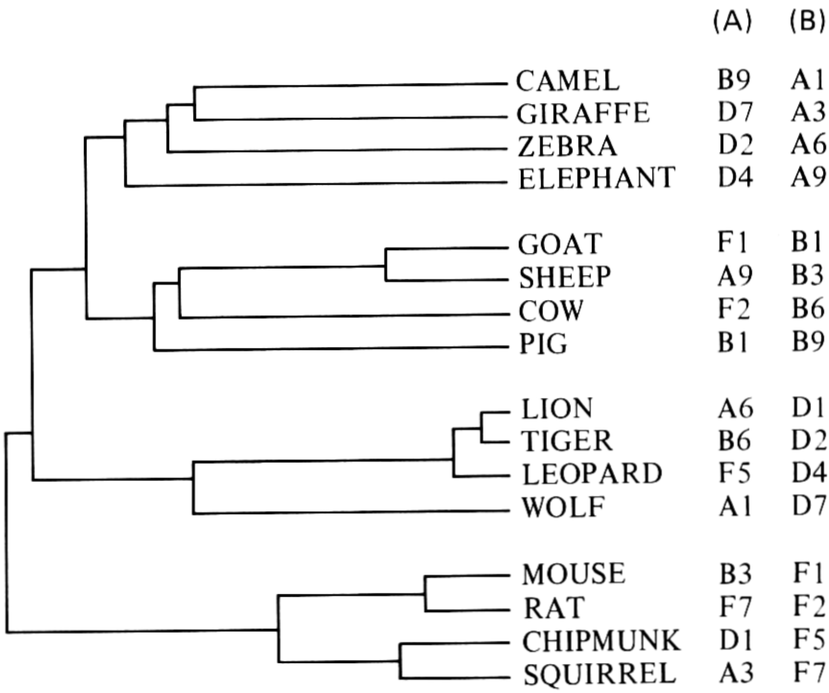

Fig. 5—Assignments of alphanumeric codes to 16 animal names. The codes in column B are assigned so as to correspond to the underlying psychological structure; those in column A are randomly assigned.

For a "real life" example of this problem on a much larger scale, consider the Dewey Decimal System for library classification. The codes are intended to convey information about the content classification of a book. They were developed by the thoughtful application of librarians' knowledge, intuition, and opinions about content relations. They undoubtedly serve the internal filing and retrieval functions of libraries quite well, and most certainly reflect at least the grosser aspects of the academically conventional categorization of fields of knowledge. It is doubtful, however, that they come close to optimizing the ease with which a new user of a library can learn the codes, or use them to go directly to the shelf where information of interest may be stored. To improve the utility of codes for this purpose, it is necessary to study the structure of knowledge domains as conceived of by the user, and incorporate corresponding perceived structure in the codes mapped to them.

4.3 Natural data designators

An alternative to providing people with menu paths, or prespecified codes to be memorized, is to allow them a relatively free choice of self-generated names or descriptions of the information they seek. Key word schemes and indexes allow people to generate candidate patterns which may give access to information that was previously stored in the system with the same designation. Query languages usually combine this approach with methods by which the user can specify combinations or other logical or arithmetic operations on the data referenced by the input terms. The vocabulary and syntax for data specifications may be strongly restricted, or they may be virtually free-form. For any such system to work, it is essential that the vocabulary and logical expressions with which the system deals be similar to those that people can easily use. If the query language is highly restrictive, it will require more learning if it forces the user to communicate in new and unfamiliar ways. If the user's input is relatively free-form, it is even more likely that the success of the system will depend on its ability to understand correctly the user's words and phrases. In any case, then, it is essential to know as much as we can about the kinds of words and expressions people find natural to use in specifying data.

First consider the matter of vocabulary. Anyone who has used a bibliographic search system, or even a common book index, will recognize that the users' description often fails to match any of the entry points provided by system designers or indexers. The problem is, thus, communication between system designers and users. The vocabulary part of the problem is whether these two groups of people use the same words. Because we know that people can say the same thing in more than one way, and because we observe less than perfect performance in information retrieval systems, we suspect that vocabulary frequently does not match. But how bad is the mismatch, especially in circumstances where designers and users are trying their best to agree? We have collected data bearing on this question in a variety of domains.[29] In one study Gomez and Kraut asked cooks to provide three to seven key words, in order of importance, for index descriptors of a large set of recipes. Overall, the chances that the first word used by any one cook (e.g., a user) would match that used by another (e.g., the indexer) was less than one in five. In a study of text editing by Landauer, Galotti, and Hartwell, secretarial students were asked to describe changes indicated by author's marks on a manuscript. The likelihood that any two secretaries used the same main verb to describe the same editing operation was less than one in ten. Similar results have been obtained by Furnas for people giving superordinate categories for sale items and by Dumais for people specifying common objects. In somewhat less formal studies, we have asked programmers and computer scientists to provide a one-word name for a program that gives information about entertainment events. Here, the frequency with which any two people used the same name was much less than one in ten. In all these cases, the name providers were under instruction to give terms they thought that others would use. They simply could not do it.

The implication of these results for retrieval system design is clear; ways must be found for dealing with synonymy. In the long run, this will probably require knowing more about the psychological nature of word meanings. However, some useful prescriptions can be made now. Computer entities, such as data files and objects, commands and programs, should be specifiable by many different names. Rather than a single name, or a few aliases for each entity, the system needs to recognize a large number of alternative names. Our data[29] show that 10 to 20 names may be needed to account for 80 percent of untrained users spontaneous "keywords" for an object. One way to implement this is with an *adaptive indexing* scheme that keeps track of the entry words actually offered by users, adding new pointers as it gains experience. (Of course, this will generate ambiguities, but they can be interactively resolved.)

Furnas has installed such a system as an experimental help facility (called AID, for Adaptive Index to Documentation) for the *UNIX** operating system user's manual, and some other utility databases and documentation on our local machines. Suppose a user tries a word that fails to return the desired object, but eventually finds that object by another route. Subject to user verification, the AID program estab-

* *UNIX* is a trademark of Bell Laboratories.

lishes a new index pointer from the user's earlier word to the final target. Then the next time this user, or anyone else, tries that new word, the computer will have some idea what to return. The program deals with ambiguity by giving the user a frequency ordered list of all objects previously referred to by a given input. We are still studying this scheme and so do not yet have conclusive data on its utility. But we think the general approach of having the system adapt to the users' vocabulary as well as the users to the system's is certain to be advantageous.

Overcoming the vocabulary mismatch problem is, unfortunately, only part of the solution. Even if we assume that we can correctly understand all the words people use, we may still not be able to know what information they are really seeking when they form a query. Database management systems typically require the user to enter well-formed expressions that logically specify a particular subset of the database. We have already noted that people do not do logic very well. They do not accurately form or understand complex Boolean or relational calculus expressions. They make frequent errors in using and interpreting quantifiers or distributed negatives. In natural spoken language, delimiters, like parentheses, or locutions, such as "the quantity," are rarely used. Yet these kinds of constructions are essential to specify data in most query languages.

In everyday speech, people seem to skirt this problem by avoiding expressions such as "not a cow or not brown and not big." What do they do instead? Clearly, people do manage to tell each other what pieces of information they would like. They often query other human beings with apparent success. Dumais and Landauer have studied the way people naturally specify data, using a version of the password game.[30] They gave college students from New York University 50 well-known objects (e.g., "*Newsweek*," "Empire State Building," "motorcycle") and asked them to provide a description that would allow another person, or a fictional computer, to guess each of the objects. The only restriction was that they were not to use the object word itself in the description. Later the descriptions were given to other people who tried to guess what was intended. The communications were reasonably successful; on the average the guesses were correct about 80 percent of the time. The more interesting aspects of the data are qualitative, involving the kind of statements that the students used. The expressions were rarely precise and seldom involved complicated relations. The most common form of specification was a single superordinate category, plus a few features: for example, for the target San Francisco, "a large California city, famous for cable cars and the Golden Gate Bridge." Another common form of specification was a set of several subordinate or associated categories: for example, for the target motorcycle, "Yamaha, Suzuki, and Harley-Davidson." Lengthier Boolean expressions than these were uncommon. They made little use of negation or exclusion. For example, the target *Newsweek*, which was included to lend itself easily to this kind of specification, was almost never described as "a popular weekly news magazine that is **not** *Time*." Perhaps the most striking feature of the descriptions was that they were, strictly speaking, quite vague. That is, they seldom specified the particular object uniquely. The most common specification for the target Empire State Building was something like "a tall building in New York City," whereas a specification like "next tallest building in New York City after the World Trade Centers" was quite rare. (Recall that the subjects were New York City college students.)

It is apparent that people rely heavily on presuppositions about knowledge in the receiver. They appear to assume that the other person will give back a "most memorable" response in the absence of further specification. The specifier unconsciously assumes that there is a strongly ordered priority among the possible items, and that the other person will share that priority sufficiently to be able to guess what is intended.

Apparently, in order to make data access more natural, we are going to have to build into systems more of this kind of knowledge of the world. Note that we are not necessarily claiming that systems will need to know nearly as much about the world as humans do. It may suffice for them to have a certain amount of statistical knowledge of what is usually meant by what.

V. ENSURING AND VALIDATING USABILITY

The common way of discussing the human factors problem in computer systems design is to speak of improving machine-user interfaces. Often this implies a design process in which the designer thinks of and specifies a system to perform a particular function, builds it, and then seeks help in optimizing the input, display, and sometimes command language aspects of its interaction with the user. These are good things to do, but they are unlikely to suffice in making data access systems maximally valuable to human users. What is necessary is to design systems that represent and provide data in ways that people are best able to think and communicate about. For those database systems whose principal users will be humans (rather than other machines), this means one should *start* the design process by obtaining information about the user. One needs to know what kind of information the users will really want, how that information is thought about by them, how it is described by them, what they will want to do with it, and in what ways they can best understand its description.

Only with this kind of information in hand can one really decide what the system should do in the first place. The next step is to invent a flexible trial system, and build a prototype in which alternative design ideas can be tested. Ideally, any conceptually separable aspects of the data representation and access methods that will impinge on the user should be designed and tested separately before being put together into an overall system. The notion of modular design has become well accepted in programming. Among other things, it allows the designer to separate and test parts of the program for their correctness and efficiency. The same idea can be applied to usability. To be more specific, such things as the choice of the best data model, the best mode of query (e.g., whether menu or keyword or something else), the size, kind, and vocabulary of the query language, and its syntactic rules, should be explored separately to be sure they are appropriate for the intended users and purpose before they are incorporated in an overall system.

At each step, and once the total system is assembled (unfortunately easily used parts do not necessarily combine into an easily used whole) the test of whether it "works" needs to be considered carefully. The traditional test applied by an engineer to a newly constructed invention is to "try it out". Without much outside help, the inventor can run the device through its paces and adequately evaluate whether it does what it is supposed to. This kind of test is, unfortunately, not sufficient for human usability purposes. The essence of the difficulty is the enormous influence that previous experience and knowledge has on the way a person perceives and interacts with the world. What may seem to be a very clean and easy system to use for its inventor, or even for colleagues with roughly similar backgrounds and motivations, may be totally unacceptable for other people. It is necessary to test a system on people who are like those who will eventually use it.

This prescribed procedure will seem quite time-consuming, difficult and tedious. It is. Currently, the only known way to get substantial improvement in the usability of systems is to design, test, design, test, through several iterations. Unfortunately, we do not yet have a large enough pool of well-tested and general principles to allow confident usability design from the outset. Nor do we have a large store of reliable pieceparts with proven usability qualities. What will eventually provide such principles and components, we believe, will be two developments. First, more research into the underlying issues in human-computer integration will lead to increased general understanding. Second, modular and iterative testing with representative users will lead to the accumulation of parts and techniques that are known to work at least sometimes, and which will make better starting places for the next system. Convergence is more rapid with a good starting point, so usable systems should get easier and easier to develop.

Another way to make convergence more rapid is to make the iterations easier. One way of doing this is to use simulations that can be easily changed. For example, one can simulate a system or module by having a human do part of what the program is intended to do. One puts a representative user at one terminal, and the designer at another. When the subject interacts with the "system", the designer responds, perhaps with the help of a relatively simple program, in one of several ways that the intended system might. This kind of experiment will almost always provide surprising information about what users actually do, and what aspects of the system's features are easy and hard to use.

All this elaborate testing with real users is a great added burden that will make systems much harder to design and build on time. We believe the effort is worthwhile because we believe the goal is important and there is currently no good alternative. Certainly, in developing large systems, it is essential to have human factors specialists working along with hardware and software specialists from the earliest design stages, and during trials and revisions. They will act as committed advocates of the intended users' interests, and their professional training and experience will make them more sensitive and knowledgeable about usage issues, and less likely to overvalue untested intuitions. But it is important to realize that the formulas, tables, and principles that guide traditional engineering are in short supply for usability aspects of design, and there is no guarantee that psychologists will have superior intuitions. What human factors specialists are best at by virtue of their training is performing efficient evaluation experiments, ones that can make "throw aways" and iterations much more effective in producing humanly usable products.

VI. CONCLUSION

We subscribe to the view that the electronic information revolution will rival the invention of movable type in its impact on people's use of knowledge. The technology of print kept serving civilization better as centuries of trial and error added authors, publishers, bookstores, indices, tables of contents, page numbers, libraries, librarians, card catalogues, and universal schooling. Undoubtedly, many of the comparable improvements for the electronic age will also come from natural evolution. Our hope, though, is that modern knowledge of cognitive psychology, experimental methods, and data analysis, systematically applied, will speed the course of evolution this time around.

REFERENCES

1. L. A. Miller and J. C. Thomas, "Behavioral Issues in the Use of Interactive Systems," Int. J. Man-Machine Studies, *9* (1977), pp. 509–36.

2. P. Reisner, "Human Factors of Data-Base Query Languages," Computing Surveys, *1* (1981), pp. 116–40.
3. B. Shneiderman, "Improving the Human Factors Aspect of Database Interactions," ACM Transactions on Database Systems, *3*, No. 4 (1978), pp. 417–39.
4. J. C. Thomas, "Psychological Issues in Database Management," Proc. Third Int. Conf. Very Large Data Bases, Tokyo, 1977.
5. M. Brosey and B. Shneiderman, "Two Experimental Comparisons of Relational and Hierarchical Database Models," Int. J. Man-Machine Studies, *10* (1978), pp. 625–37.
6. B. M. Durding, C. A. Becker, and J. D. Gould, "Data Organization," Human Factors, *19*, No. 1 (1977), pp. 1–14.
7. J. D. Gould and R. N. Ascher, "Use of an IQF-Like Query Language by Non-Programmers," IBM Res. Rep. RC 5279, IBM T. J. Watson Res. Ctr., Yorktown Heights, NY, February 1975.
8. D. Greenblatt and J. Waxman, "A Study of Three Database Query Languages," in *Databases: Improving Usability and Responsiveness*, B. Shneiderman, Ed., New York: Academic Press, 1978.
9. L. A. Miller, "A Study in Man-Machine Interaction," Proc. AFIPS 1977 NCC, Vol. 46, AFIPS Press, Montvale, NJ, pp. 409–21.
10. P. Reisner, "Use of Psychological Experimentation as an Aid to Development of a Query Language," IEEE Trans. Software Eng. *SE-3*, No. 3 (1977), pp. 218–29.
11. P. Reisner, R. F. Boyce, and D. D. Chamberlin, "Human Factors Evaluation of Two Data Base Query Languages: SQUARE and SEQUEL," Proc. AFIPS 1975 NCC, Vol. 44, AFIPS Press, Montvale, NJ., pp. 447–52.
12. D. W. Small and L. J. Weldon, "The Efficiency of Retrieving Information From Computers Using Natural and Structured Query Languages," Rep. SAI-78-655-WA, Science Applications, Arlington, Va., Sept. 1977.
13. J. C. Thomas, "Quantifiers and Question Asking," IBM Res. Rep. RC 5866, IBM T. J. Watson Res. Ctr., Yorktown Heights, N.Y., 1976.
14. J. C. Thomas and J. D. Gould, "A Psychological Study of Query by Example," Proc. 1975 National Computer Conference, AFIPS Press, Montvale, NJ.
15. J. L. Frankhuizen and T. G. M. Vrins, "Human Factors Studies With Viewdata," Ninth Int. Symp. on Human Factors in Telecommunication, Holmdel, NJ, 1980.
16. E. Rosch and C. B. Mervis, "Family Resemblances: Studies in the Internal Structure of Categories," Cog. Psych., 7 (1975), pp. 573–605.
17. P. C. Wason and P. D. Johnson-Laird, *Psychology of Reasoning: Structure and Content.* Cambridge, Mass.: Harvard University Press, 1972.
18. E. Soloway, J. Lochhead, and J. Clement, "Does Computer Programming Enhance Problem Solving Ability? Some Positive Evidence on Albegra Word Problems," Technical Report, Department of Computer and Information Science, University of Massachusetts, Amherst, 1981.
19. A. Tversky and D. Kahneman, "Judgment Under Uncertainty: Heuristics and Biases," Science, *185* (1974), pp. 1124–30.
20. J. B. Kruskal and M. Wish, *Multidimensional Scaling.* Beverly Hill, Ca.: Sage Publications, 1978.
21. S. Pruzansky, A. Tversky, and J. D. Carroll, unpublished work, 1980.
22. G. Furnas, "Psychological Structure in Information Organization and Retrieval: Arguments for More Considered Approaches and Work in Progress," Workshop/Symposium on Human Computer Interaction, Atlanta, GA, 1981.
23. R. McDougall, "Recognition and Recall," J. Phil. Psych. and Sci. Meth., *1* (1904), pp. 229–33.
24. R. N. Shepard, "Role of Generalization in Stimulus-Response Compatibility," Perceptual and Motor Skills, *13* (1961), pp. 59–62.
25. S. Sattath and A. Tversky, "Additive Similarity Trees," Psychometrika, *42* (1977), pp. 319–45.
26. J. M. Parkman, "Temporal Aspects of Digit and Letter Inequality Judgements," J. Exp. Psych., *91* (1971), pp. 191–205.
27. R. N. Shepard, D. W. Kilpatric, and J. P. Cunningham, "The internal representation of numbers," Cog. Psych., *7* (1975), pp. 82–138.
28. T. K. Landauer, S. T. Dumais, and G. W. Furnas, unpublished data.
29. G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais, "Statistical Semantics: Analyses of the Potential Performance of Keyword Access Systems," In press, B.S.T.J.
30. S. T. Dumais and T. K. Landauer, unpublished work.