NATURAL VS. PRECISE CONCISE LANGUAGES FOR HUMAN OPERATION OF COMPUTERS: RESEARCH ISSUES AND EXPERIMENTAL APPROACHES

Ben Shneiderman, Department of Computer Science
University of Maryland, College Park, MD.

This paper raises concerns that natural language front ends for computer systems can limit a researcher's scope of thinking, yield inappropriately complex systems, and exaggerate public fear of computers. Alternative modes of computer use are suggested and the role of psychologically oriented controlled experimentation is emphasized. Research methods and recent experimental results are briefly reviewed.

1. INTRODUCTION

The capacity of sophisticated modern computers to manipulate and display symbols offers remarkable opportunities for natural language communication among people. Text editing systems are used to generate business or personal letters, scientific research papers, newspaper articles, or other textual data. Newer word processing, electronic mail, and computer teleconferencing systems are used to format, distribute, and share textual data. Traditional record keeping systems for payroll, credit verification, inventory, medical services, insurance, or student grades contain natural language/textual data. In these cases the computer is used as a communication medium between humans, which may involve intermediate stages where the computer is used as a tool for data manipulation. Humans enter the data in natural language form or with codes which represent pieces of text (part number instead of a description, course number instead of a title, etc.). The computer is used to store the data in an internal form incomprehensible to most humans, to make updates or transformations, and to output it in a form which humans can read easily. These systems should act in a comprehensible "tool-like" manner in which system responses satisfy user expectations.

Several researchers have commented on the importance of letting the user be in control [1], avoiding acausality [2], promoting the personal worth of the individual [3], and providing predictable behavior [4]. Practitioners have understood this principle as well: Jerome Ginsburg of the Equitable Life Assurance Society prepared an in-house set of guidelines which contained this powerful claim:

> "Nothing can contribute more to satisfactory system performance than the conviction on the part of the terminal operators that they are in control of the system and not the system in control of them. Equally, nothing can be more damaging to satisfactory system operation, regardless of how well all other aspects of the implementation have been handled, than the operator's conviction that the terminal and thus the system are in control, have 'a mind of their own,' or are tugging against rather than observing the operator's wishes."

I believe that control over system function and predictable behavior promote the personal worth of the user, provide satisfaction, encourage competence, and stimulate confidence. Many successful systems adhere to these principles and offer terminal operators a useful tool or an effective communication media.

An idea which has attracted researchers is to have the computer take coded information (medical lab test values or check marks on medical history forms) and generate a natural language report which is easy to read, and which contains interpretations or suggestions for treatment. When the report is merely a simple textual replacement of the coded data, the system may be accepted by users, although the compact form of the coded data may still be preferable for frequent users. When the suggestions for treatment replace a human decision, the hazy boundary between computer as tool and computer as physician is crossed.

Other researchers are more direct in their attempt to create systems which simulate human behavior. These researchers may construct natural language front ends to their systems allowing terminal operators to use their own language for operating the computer. These researchers argue that most terminal operators prefer natural language because they are already familiar with it, and that it gives the terminal operator the greatest power and flexibility. After all, they argue, computers should be easy to use with no learning and computers should be designed to participate in dialogs using natural language. These sophisticated systems may use the natural language front ends for question-answering from databases, medical diagnosis, computer-assisted instruction, psychotherapy, complex decision making, or automatic programming.

2. DANGERS OF NATURAL LANGUAGE SYSTEMS

When computer systems leave users with the impression that the computer is thinking, making a decision, representing knowledge, maintaining beliefs, or understanding information I begin to worry about the future of computer science. I believe that it is counterproductive to work on systems which present the illusion that they are reproducing human capacities. Such an approach can limit the researcher's scope of thinking, may yield an inappropriately complex system, and potentially exaggerates the already present fear of computers in the general population.

2.1 NATURAL LANGUAGE LIMITS THE RESEARCHER'S SCOPE

In constructing computer systems which mimic rather than serve people, the developer may miss opportunities for applying the unique and powerful features of a computer: extreme speed, capacity to repeat tedious operations accurately, virtually unlimited storage for data, and distinctive input/output devices. Although the slow rate of human speech makes menu selection impractical, high speed computer displays make menu selection an appealing alternative. Joysticks, lightpens or the "mouse" are extremely rapid and accurate ways of selecting and moving graphic symbols or text on a display screen. Taking advantage of these and other computer-specific techniques will enable designers to create powerful tools without natural language commands. Building computer systems which behave like people do, is like building a plane to fly by flapping its wings. Once we get past the primitive imitation stage and understand the scientific basis of this new technology (more on how to do this later), the human imitation strategies will be merely museum pieces for the 21st century, joining the clockwork human imitations of the 18th century. Sooner or later we will have to accept the idea that computers are merely tools with no more intelligence than a wooden pencil. If researchers can free themselves of the human imitation game and begin to think about using computers for problem solving in novel ways, I believe that there will be an outpouring of dramatic innovation.

2.2 NATURAL LANGUAGE YIELDS INAPPROPRIATELY COMPLEX SYSTEMS

Constructing computer systems which present the illusion of human capacities may yield inappropriately complex systems. Natural language interaction with the tedious clarification dialog seems archaic and ponderous when compared with rapid, concise, and precise database manipulation facilities such as Query-by-example or commercial word processing systems. It's hard to understand why natural language systems seem appealing when contrasted with modern interactive mechanisms like high speed menu selection, light pen movement of icons, or special purpose interfaces which allow the user to directly manipulate their reality. Natural language systems must be complex enough to cope with user actions stemming from a poor definition of system capabilities.

Some users may have unrealistic expectations of what the computers can or should do. Rather than asking precise questions from a database system, a user may be tempted to ask how to improve profits, whether a defendant is guilty, or whether a military action should be taken. These questions involve complex ideas, value judgments, and human responsibility for which computers cannot and should not be relied upon in decision making.

Secondly, users may waste time and effort in querying the database about data which is not contained in the system. Codd [5] experienced this problem in his RENDEZVOUS system and labeled it "semantic overshoot." In command systems the user may spend excessive time in trying to determine if the system supports the operations they have in mind.

Thirdly, the ambiguity of natural language does not facilitate the formation of questions or commands. A precise and concise notation may actually help the user in thinking of relevant questions or effective commands. A small number of well defined operators may be more useful than ill-formed natural language statements, especially to novices. The ambiguity of natural language may also interfere with careful thinking about the data stored in the machine. An understanding of onto/into mappings, one-to-one/one-to-many/many-to-many relationships, set theory, boolean algebra, or predicate calculus and the proper notation may be of great assistance in formulating queries. Mathematicians (and musicians, chemists, knitters, etc.) have long relied on precise concise notations because they help in problem solving and human-to-human communication. Indeed, the syntax of precise concise query or command language may provide the cues for the semantics of intended operations. This dependence on syntax is strongest for naive users who can anchor novel semantic concepts to the syntax presented.

2.3 NATURAL LANGUAGE GENERATES MISTRUST, ANGER, FEAR AND ANXIETY

Using computer systems which attempt to behave like humans may be cute the first time they are tried, but the smile is short-lived. The friendly greeting at the start of some computer-assisted instruction systems, computer games, or automated bank tellers, quickly becomes an annoyance and, I believe, eventually leads to mistrust and anger. The user of an automated bank teller machine which starts with "Hello, how can I help you?" recognizes the deception and soon begins to wonder how else the bank is trying to deceive them. Customers want simple tools whose range of functions they understand. A more serious problem arises with systems which carry on a complete dialog in natural language and generate the image of a robot. Movie and television versions of such computers produce anxiety, alienation, and fear of computers taking over.

In the long run the public attitude towards computers will govern the future of acceptable research, development, and applications. Destruction of computer systems in the United States during the turbulent 1960's, and in France just recently (Newsweek April 28, 1980 — An underground group, the Committee for the Liquidation or Deterrence of Computers claimed responsibility for bombing Transportation Ministry computers and declared: "We are computer workers and therefore well placed to know the present and future dangers of computer systems. They are used to classify, control and to repress.") reveal the anger and fear that many people associate with computers. The movie producers take their ideas from research projects and the public reacts to common experiences with computers. Distortions or exaggerations may be made, but there is a legitimate basis to the public's anxiety.

One more note of concern before making some positive and constructive suggestions. It has often disturbed me that researchers in natural language usually build systems for someone else to use. If the idea is so good, why don't researchers build natural language systems for their own use. Why not entrust their taxes, home management, calendar/schedule, medical care, etc. to an expert system? Why not encode their knowledge about their own disipline in a knowledge representation language? If such systems are truly effective then the developers should be rushing to apply them to their own needs and further their professional career, financial status, or personal needs.

3. HUMAN FACTORS EXPERIMENTATION FOR DEVELOPING INTERACTIVE SYSTEMS

My work with psychologically oriented experiments over the past seven years has made a strong believer in the utility of empirical testing [6]. I believe that we can get past the my-language-is-better-than-your-language or my-system-is-more-natural-and-easier-to-use stage of computer science to a more rigorous and disciplined approach. Subjective, introspective judgments based on experience will always be necessary sources for new ideas, but controlled experiments can be extremely valuable in demonstrating the effectiveness of novel interactive mechanisms, programming language control structures, or new text editing features. Experimental testing requires careful statement of a hypothesis, choice of independent and dependent variables, selection and assignment of subjects, administration to minimize bias, statistical analysis, and assesment of the results. This approach can reveal mistaken assumptions, demonstrate generality, show the relative strength of effects, and provide evilence for a theory of human behavior which may suggest new research.

A natural strategy for evaluating the effectiveness of natural language facilities would be to define a task, such as retrieval of ship convoy information or solution of a computational problem, then provide subjects with either a natural language facility or an alternative mode such as a query language, simple programming language, set of commands, menu selection, etc. Training provided with the natural language system or the alternative would be a critical issue, itself the subject of study. Subjects would perform the task and be evaluated on the basis of accuracy or speed. In my own experience, I prefer to provide a fixed time interval and measure performance. Since inter-subject variability in task performance tends to be very large, within subjects (also called repeated measures) designs are effective. Subjects perform the task with each mode and the statistical tests compare scores in one mode against the other. To account for learning effects, the expectation that the second time the task is performed the subject does better, half the subjects begin with natural language, while half the subjects begin with the alternative mode. This experimental design strategy is known as counterbalanced orderings.

If working systems are available, then an on-line experiment provides the most realistic environment, but problems with operating systems, text editors, sign-on procedures, system crashes, and other failures can bias the results. Experimenters may also be concerned about the slowness of some natural language systems on currently available computers as a biasing factor in such experiments. An alternative would be on-line experiments where a human plays the role of a natural language system. This appears to be viable alternative [7] if proper precautions are taken. Paper and pencil studies are a suprisingly useful approach and are valuable since administration is easy. Much can be learned about human thought processes and problem solving methods by contrasting natural language and proposed alternatives in paper and pensil studies. Subjects may be asked to write queries to a database of present a sequence of commands using natural language or some alternative mode [9].

There is a growing body of experiments that is helping to clarify issues and reveal problems about human performance with natural language usage on computers. Codd [5] and Woods [8] describe informal studies in user performance with their natural language systems. Small and Weldon [7] conducted the first rigorous comparison of natural language with a database query language. Twenty subjects worked with a subset of SEQUEL and an on-line simulated natural language system to composed queries. Shneiderman [9] describes a similar paper and pencil experiment comparing performance with natural language and a subset of SEQUEL. The results of both of these experiments suggest that precise concise database query language do aid the user in rapid formulation of more effective queries.

Damerau [10] reports on a field study in which a functioning natural language system, TQA, was installed in a city planning office. His system succeeded on 513 out of 788 queries during a one year period. Hershman, Kelly and Miller [11] describe a carefully controlled experiment in which ten naval officers used the LADDER natural language system after a ninety minute training period. In a simulated rescue attempt the system properly responded to 258 out of 336 queries.

Critics and supporters of natural language usage can all find heartening and disheartening evidence from these experimental reports. The contribution of these studies is in clarification of the research issues, development of the experimental methodology, and production of guidelines for developers of interactive systems. I believe that developers of natural language systems should avoid over-emphasizing their tool and more carefully analyze the problem to be solved as well as human capacities. If the goal is to provide an appealing interface for airline reservations, bank transactions, database retrieval, or mathematical problem solving, then the first step should be a detailed review of the possible data structures, control structures, problem decompositions, cognitive models that the user might apply, representation strategies, and importance of background knowledge. At the same time there should be a careful analysis of how the computer system can provide assistance by representing and displaying data in a useful format, providing guidance in choosing alternative strategies, offering effective messages at each stage (feedback on failures and successes), recording the history and current status of the problem solving process, and giving the user comprehensible and powerful commands.

Experimental research will be helpful in guiding developers of interactive systems and in evaluating the importance of the user's familiarity with:

1) the problem domain
2) the data in the computer
3) the available commands
4) typing skills
5) use of tools such as text editors
6) terminal hardware such as light pens, special purpose keyboards or unusual display mechanisms
7) background knowledge such as boolean algebra, predicate calculus, set theory, etc.
8) the specific system - what kind of experience effect or learning curve is there

Experiments are useful because of their precision, narrow focus, and replicability. Each experiment may be a minor contribution, but, with all its weaknesses, it is more reliable than the anecdotal reports from biased sources. Each experimental result, like a small tile in a mosaic which has a clear shape and color, adds to our image of human performance in the use of computer systems.

4. REFERENCES

1) Cheriton, D.R., Man-Machine interface design for time-sharing systems, Proceedings of the ACM National Conference, (1976), 362-380.

2) Gaines, Brian R. and Peter V. Facey, Some experience in interactive system development and application, Proceedings of the IEEE, 63, 6, (June 1975), 894-911.

3) Pew, R.W. and A.M. Rollins, Dialog Specification Procedure, Bolt Beranek and Newman, Report No. 3129, Revised Edition, Cambridge, Massachusetts, 02138, (1975).

4) Hansen, W.J., User engineering principles for interactive systems, Proceedings of the Fall Joint Computer Conference, 39, AFIPS Press, Montvale, New Jersey, (1971), 523-532.

5) Codd, E.F., HOW ABOUT RECENTLY? (English dialogue with relational databases using RENDEZVOUS Version 1), In B. Shneiderman (Ed.), Databases: Improving Usability and Responsiveness, Academic Press, New York, (1978), 3-28.

6) Shneiderman, B., Software Psychology: Human Factors in Computer and Information Systems, Winthrop Publishers, Cambridge, MA (1980).

7) Small, D.W. and L.J. Weldon, The efficiency of retrieving information from computers using natural and structured query languages, Science Applications Incorporated. Report SAI-78-655-WA, Arlington,Va., (Sept. 1977).

8) Woods, W.A., Progress in natural language understanding - an application to lunar geology, Proceedings of the National Computer Conference, 42, AFIPS Press, Montvale, New Jersey, (1973), 441-450.

9) Shneiderman, B., Improving the human factors aspect of database interactions, ACM Transactions on Database Systems, 3, 4, (December 1978a), 417-439.

10) Damerau, Fred J., The Transformational Query Answering System (TQA) operational statistics - 1978, IBM T.J. Watson Research Center RC 7739, Yorktown Heights, N.Y. (June 1979).

11) Hershman, R.L., R.T. Kelly and H.G. Miller, User performance with a natural language query system for command control, Navy Personnel Research and Development Center Technical Report 79-7, San Diego,CA, (1979).