UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

AD-A196 839

DTIC FILE COPY

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION	Unclassified
1b. RESTRICTIVE MARKINGS	--
2a. SECURITY CLASSIFICATION AUTHORITY	--
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	--
3. DISTRIBUTION / AVAILABILITY OF REPORT	Approved for public release; distribution unlimited.
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	Technical Report DSI-SBIR-II-2; Technical Report DSI-SBIR-II-3
5. MONITORING ORGANIZATION REPORT NUMBER(S)	ARI Research Note 88-50
6a. NAME OF PERFORMING ORGANIZATION	Defense Systems, Inc.
6b. OFFICE SYMBOL (If applicable)	
7a. NAME OF MONITORING ORGANIZATION	U.S. Army Research Institute for the Behavioral and Social Sciences
6c. ADDRESS (City, State, and ZIP Code)	7903 Westpark Drive, McLean, Virginia 22102
7b. ADDRESS (City, State, and ZIP Code)	Fort Leavenworth Field Unit, P.O. Box 3407 (PERI-SL), Ft. Leavenworth, Kansas 66027-0347
8a. NAME OF FUNDING / SPONSORING ORGANIZATION	U.S. Army Research Institute
8b. OFFICE SYMBOL (If applicable)	PERI-SZ
9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	MDA903-86-C-0204
8c. ADDRESS (City, State, and ZIP Code)	5001 Eisenhower Avenue, Alexandria, Virginia 22333-5600

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
62717	2Q1627 17A790	1.4.4.	1.4.4.S6

11. TITLE (Include Security Classification)

User-Computer Interface Technology: An Assessment of the Current State of the Art

12. PERSONAL AUTHOR(S)

Sam Landee-Thompson (International Information Systems, Inc.)

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Interim	FROM Jul 86 TO Mar 87	June 1988	78

16. SUPPLEMENTARY NOTATION

Stanley M. Halpin, contracting officer's representative

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	User-Computer Interface; Intelligent Interface; Artificial Intelligence; Mental Models (OVER)

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

There is an obvious challenge for those doing research and development work in user-computer interaction to identify and track research developments from a wide range of related fields. This assessment of relevant literature is intended to extend the user-computer R&D community's appreciation for some of the current trends, directions, accomplishments and challenges in developing intelligent interactive system interfaces. The basic document consists of an assessment of the state-of-the-art based upon the analysis of 39 representative recent publications. These are analyzed in terms of defining the problem, recent developments toward intelligent interfacing and difficulties in developing intelligent interfaces. Conclusions and suggestions are offered concerning promising areas for research and development. The document also contains an annotated bibliography in which the contents of 57 recent publications relevant to user-computer interface technology are summarized. This document will prove useful to anyone seeking a definition and synopsis of the state-of-the-art in intelligent user-computer interface technology.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code)	22c. OFFICE SYMBOL
Mr. Rex R. Michel	(913) 684-4933	PERI-SL

DD Form 1473, JUN 86 Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

ARI RESEARCH NOTE 88-50

18. Subject Terms (continued)

Human-Computer Interaction
Human-Computer Interface
Human-Computer Dialogue
Adaptive Interface
Natural Language
User Models

U. S. ARMY RESEARCH INSTITUTE

FOR THE BEHAVIORAL AND SOCIAL SCIENCES

A Field Operating Agency under the Jurisdiction of the

Deputy Chief of Staff for Personnel

L. NEALE COSBY
Colonel, IN
Commander

EDGAR M. JOHNSON
Technical Director

Research accomplished under contract
for the Department of the Army

Defense Systems, Inc.

Technical review by

Laurel E. Allender

ARI Research Note 88-50

User-Computer Interface Technology: An Assessment of the Current State of the Art

Sam Landee-Thompson
Defense Systems, Inc.

for

Ari Field Unit at Fort Leavenworth, Kansas
Stanley M. Halpin, Chief

Systems Research Laboratory
Robin L. Keesee, Director

U. S. Army
Research Institute for the Behavioral and Social Sciences

June 1988

USER-COMPUTER INTERFACE TECHNOLOGY: AN ASSESSMENT OF THE CURRENT STATE-OF-THE-ART

CONTENTS

LIST OF FIGURES

USER-COMPUTER INTERFACE TECHNOLOGY: AN ASSESSMENT OF THE CURRENT STATE-OF-THE ART

INTRODUCTION

A sizable body of literature exists that relates to the interaction between humans and computers. This paper attempts to provide a brief review of current literature with special attention to developments related to intelligent interfaces.

This paper contains five sections. The first section provides an overview of the paper contents. The second section offers a brief view of how the notion of the interface has evolved. While early interface issues deal with perceptual aspects of humans, newer thinking characterizes the human as a decision maker working with, rather than on, a computer.

The third section of this paper discusses intelligent interfaces. Both researcher "wish lists" of what an intelligent interface could be and recent developments toward intelligent interfaces are presented. A discussion of the issue of user models is contained in this section which draws on developments in the computer-aided instruction literature.

The fourth section of this paper addresses difficulties in the development of intelligent interfaces. Two major areas of difficulties are addressed; namely, language understanding and mental models.

The fifth section of this paper offers conclusions based on the literature review. This final section includes a brief discussion of potential future directions for research in human-computer interaction.

Appendix A contains an annotated bibliography of literature relevant to user-computer interface technology. Although not intended to be exhaustive, it does represent the trends, advancements and difficulties extant in this research area.

HUMANS AND COMPUTERS: INTERFACING AND INTERACTING

The study of the human and computer and the interaction between them is the focus of a multidisciplinary field of researchers. Within the literature one finds the terms human-machine interaction, user-computer interface, and user-computer interaction used interchangably. However, the interface between the human and computer refers to the common boundary between them. Interaction, on the other hand, pertains to the mutual action or influence the human and computer have on each other.

Halpin and McCullough (1986) note that the terms human-machine interaction, user-computer interface, and user-computer interaction actually reflect subtle differences in the research perspective. Halpin and McCullough offer the following distinctions in terminology:

> Typically, research in the human-machine interaction area concerns an operator monitoring and/or controlling a system, focusing on the perceptual and mechanical aspects of such tasks.
>
> Much of the human factors contribution in this field lies in the area of the user-computer interface, where the research focuses on the study of input/output devices. Halpin and McCullough characterize this line of research as being by and large atheoretical (some attention has been given to research efforts of cognitive psychology).
>
> The focus of user-computer interaction differs in that the user is characterized as a decision maker, rather than an operator, and is part of the "system" and the research concern is in the area of communication (the authors identify this area as a current gap in research).

Gaines and Shaw (1986a) maintain that empirical human factors research provides insufficient design guidance because the studies have focused on problems and issues of the past. They note that not only are the results of little value to the present but they are of no value to the future. However, the study of the human factors of both the interface and interaction tend to be a necessary prerequisite for the study of total knowledge systems which bring together artificial intelligence and cognitive science (Gaines and Shaw, 1986a).

While criticisms such as this one may be justified, the study of human-computer interfaces and interactions is a relatively new field. Twenty years ago, most of us handed batch cards to an operator who dealt with the machine. In a fairly short time, we now have microcomputers on our desks - the middle man has been removed. Now, rather than deal remotely with each other. the user and computer must interact in a face-to-face fashion; both humans and machines must communicate.

What's in an Interface?

While the interface is the common boundary between the user and the computer, what does that mean? In a general manner, the interface is easy to find, "follow a data path outward from the computer's central processor until you stumble across a human being" (Card, Moran, and Newell, 1983). Actual boundaries are subtle. Input devices, such as a keyboard, mouse, and touch screen, are part of the interface. Dialog techniques, which is the term associated with how humans and computers interact, are part of the interface. Input devices are the human side of the the interaction picture, while computer outputs are the computer side of interaction.

According to some, (for example, Halpin and McCullough, 1986; Gaines and Shaw, 1986; Card, Moran, and Newell, 1983), the effective partnership of humans and computers requires a dialog perspective on the interaction. This perspective goes beyond many of the human factors studies which focus on the perceptual characteristics of interface design. The dialog perspective connotes a cooperative exchange between the user and computer. Schvaneveldt, McDonald, and Cooke (1986) maintain that effective user-computer interactions depend on the communication of structure and organization. In particular, they note the need for user awareness of:

- Data structures in a system or application program.
- Dependencies among the data elements.
- The data needed by the program.
- The data produced by the program.

In a similar manner, Norman (1986) advances the notion of the system image. The system image is what is portrayed to the user through interaction with the system; it is the result of the physical structure, including documentation and instructions. The system image can be contrasted with the programmer's conceptual model of the system, termed a design model. The mental model the user develops of a system results exclusively from interactions with the system image. Norman notes the need to present the appropriate system image to the user. From his perspective, the appropriate system image should assist the user to understand what is going on, in order to give the user a sense of control. Norman points out:

> When systems take too much control of the environment, they can cause serious social problems. Many observers have commented on the dehumanizing results of automation in the workplace. In part, this automatically results from the systems that take control away from the users...The problem arises over and over again as various workplaces become automated, whether it is the factory, the office, or the aviation cockpit. I believe the difficulties arise from the tension between the natural desire to want intelligent systems that can compensate for our inadequacies and the desire to feel in control of the outcome. (p. 52).

Norman and Schvaneveldt, et al., are noting that the interface provides a window into the internal operation of the computer. Schvaneveldt, et al., discuss a window that would convey, at a general level, a view for the user of data base internals. From Norman's perspective, care must be taken to provide users with a transparent window particularly in the area of intelligent tools; users need to be able to understand what the tools are doing and how they operate. These researchers approach the subject of the development of the interface from theoretical perspectives in mental modeling; more will be said about mental models later in this paper.

While the degree of control given to automation is a concern at the individual user level, it is also a concern at the societal level. As Coombs (1986) notes:

> A major concern in the application of computers to the control of physical and mechanical processes is the degree of autonomy to give the system. Although this has long been an important research area, it has received additional, and urgent,

attention since the year 1980, which has been named the "Year of Mishaps" for American technology. As pointed out by Kopec & Michie (1983), the year saw:

- numerous near-misses at Kennedy and other busy national and international airports;

- technical problems with the operation of the Hercules Sea Stallion helicopters which resulted in aborting the rescue mission to free the American hostages in Iran;

- the NORAD military computer "falsely signalling a Soviet nuclear attack on three separate occasions".

Moreover, these followed the incident at Three Mile Island nuclear power station in March, 1979 when inappropriate operator action almost led to a melt-down of the core...We have, however, lost our taste for full automation , and wish to find means of engaging our machines as partners. (p. 1)

The objective of establishing a partnership between humans and machines is an ideal that is likely to become pervasive within the diverse community concerned with human-computer interaction. Some researchers refer to the objective as a symbiotic user-computer system (Halpin and McCullough, 1986; Gaines and Shaw 1986a). Such views will move the research community away from notions of the user as an operator or a passive perceiver at the interface. The user in partnership with the computer will be viewed as a decision maker/problem solver teamed with a computer serving in the role of advisor or assistant.

Guiding Interface Design

Much of the literature concerning the human-computer interface focuses on providing guidance to system designers. The available guidance to designers of the human-computer interface comes in all shapes and sizes, ranging from issues of broad concern to very specific details. A designer turning to the literature to obtain practical aid in solving a specific design problem may be disappointed by the nature of guidance frequently encountered. Some of the guidance may be useful concepts to think about but fall short in directions for actual implementation. Other guidance may appear to be so context specific that the designer may well question the applicability of the guidance to his or her particular problem.

Bannon (1986) identifies a number of issues that system designers should consider in their design of interfaces. These issues include:

- A theory of the user. This issue identifies the need for detailed models of how people think and communicate as fundamental to system development. An additional requirement is for a "more enlightened view of people that recognizes their need for variety and challenge in the tasks that they perform" (in Norman and Draper, p. 26).

- A theory of the task domain. This issue identifies task domain knowledge as key to design of usable systems. An adequate model of the user coupled with an incomplete domain understanding is likely to produce a toy rather than a tool.

- The organization of work around technology. This issue stresses the point "that the design of interfaces should include a concern of how features of the interface might accomodate different kinds of work organization, allowing for flexible tailoring of the system to accomodate particular user groups" (in Norman and Draper p. 27).

- The role of design in society. This issue suggests that the designer consider the purpose of the object they are designing and the impact it will have on society.

Norman (1986) notes the need for interface design guidance to be specific (e.g., how many items in a menu) as well as more general. In an effort to develop high-level design guidance, Norman offers the following prescriptions for structuring the design process:

- Create a science of user-centered design. The need here is for design principles and simulation tools to develop designs before constructing them.

- Take interface design seriously as an independent and important problem. This principle identifies the need for collaborative efforts to accomplish user-centered design. Norman identifies the special knowledge requirements for user-centered design as: knowledge of design, programming, and technology; knowledge of people, principles of mental computation, communication, and interaction; expert knowledge of [illegible] to be accomplished.

- Separate the design of the interface from the design of the system. This principle calls for modularity in design. Included in the modularity principle is the notion that the interface is a translator from the system to the user and that the system does not communicate directly with the user (particularly in system terms such as mystical error messages). According to Norman, modularity will permit system changes to be made without impacting the interface (as well as changes in the interface that do not impact the system).

- Do user-centered system design: Start with the needs of the user. From the user's perspective, the interface **is** the system. Norman suggests that the interface design be driven from the interaction requirements.

The guidelines of Bannon and Norman are obviously general, identifying things to think about as an interface designer. Card, Moran, and Newell (1983) offer principles for the human-computer interface design. These ten principles include (from Card, Moran, and Newell, 1983, p. 418):

1. Early in the system design process, consider the psychology of the user and the design of the user interface.

2. Specify the performance requirements. This is a principle where trade-off analysis is required since optimizing performance on a specific variable often impacts performance on others.

3. Specify the user population. If a highly varied population is the target for the system, different types of users should be characterized since performance may be considerably different.

4. Specify the tasks. While it may not be possible to delineate all tasks, a reasonable benchmark sample is preferable to specifying gross task characteristics. It is important to specify high frequency tasks as well as a sample of qualitatively different types of tasks.

5. Specify the methods to do the tasks. The authors note that a skilled user has a highly integrated knowledge of tasks, methods, and the interconnection between them. Lengthy and awkward methods need to be identified and eliminated during the design process.

6. Match the method analysis to the level of resource commitment in the design process. In an iterative design process, different performance models have applicability at various developmental stages. Card, Moran, and Newell present various models for the various stages.

7. To reduce the performance time of a task by an expert, eliminate operators from the method for doing the task. This can be done at any level of analysis. In their investigation of performance models, Card, Moran, and Newell found that expert performance is composed of a sequence of operators. Expert performance time equals the sum of the time required for each operator. This guidelines suggests that expert performance time may be decreased by reducing the operators required for task performance.

8. Design the set of alternative methods for a task so that the rule for selecting each alternative is clear to the user and easy to apply. This principle is concerned with permitting flexibility between users of varying skill levels.

9. Design a set of error-recovery methods. Experimental data presented by the authors indicate that expert users allow up to 30% of their time for correcting errors. The notion of designing for efficient error recovery may need to be adopted as an interface goal (Brown, 1986, contrasts the need for management of trouble to the notions of "idiot-proofing" automation).

10. Analyze the sensitivity of performance predictions to assumptions. In the design process assumptions are made about user characteristics, the computer system, as well as the task environment. Card, Moran, and Newell suggest that any prediction of human-computer performance should be checked for the sensitivity to these assumptions. This type of analysis helps to identify design decisions that may require re-evaluation as the iterative design evolves.

The guidelines presented by Card, Moran, and Newell focus on the notion of performance models. These guidelines were developed from extensive experimental research and are considered by the authors to be fundamental principles rather than an exhaustive set of guidelines.

At the other end of the guideline continuum, one finds highly detailed specifications. For example, Smith and Mosier (1984) published 679 guidelines in their report Design Guidelines for User-System Interface Software. This report is designed specifically to provide guidance to designers. The 679 guidelines are grouped into six functional areas (e.g., data entry, user guidance). In some cases, the guidance provided in the document is clear (e.g., use an active rather than passive voice in user guidance messages). In other cases, the guidance may be less than straightforward in implementation for the designer (e.g., "when techniques adopted for user guidance may slow an experienced user, provide alternative paths/modes to by-pass standard guidance procedures"). Such guidelines may have more utility in the realm of evaluating than designing an interface.

The typical problem encountered by designers concerns trade-offs that must be made to satisfy competing objectives and imposed limitations; guidelines such as those of Smith and Mosier provide little support to trade-off analysis. Guidelines to aid in trade-off analysis would focus more on identifying options and alternatives to the designer. The designer looking for trade-off guidance may turn to individual research efforts. Miller (1981), for example, offers empirical evidence on the depth/breadth trade-off in hierarchical menus in the context of goal acquisition performance. Miller's research demonstrates that in the performance of goal acquisition tasks optimal performance, fewest errors, and ease of learning are found with hierarchical menus of two levels and eight choices per level. He further suggests that when a trade-off must be made an expansion in breadth is recommended over depth.

Currently, the Human Factors Society's Computer Systems Group has undertaken the drafting of human-computer interaction standards. The guidelines which will be issued will be based on principles that have a foundation in scientific evidence and empirical data and have been recognized and accepted by those knowledgeable in the area. Importantly, the guidelines are intended for direct use by interface designers; the information will therefore be directly applicable in the trade-off decisions faced in the design process. The fact that this society has chosen to undertake this effort is indicative of the existing gap in the currently available standards and design guidelines.

There is certainly no shortage of guidelines available to the designer of the human-computer interface. However, finding guidelines for a specific problem is not straightforward. Given the vast literature available, a useful guideline might be found in a technical report (e.g., Smith and Mosier, 1984), or the guideline may appear in a particular journal article from a large number of possible journals. Finding a needed guideline, or confirming that one does not exist, has the potential to require considerable research effort.

Guidance is available along a continuum of specificity. At the general end of the continuum the designer is likely to ask the question "what does it mean" to consider the societal context in which this system will operate? The general guidelines do not lend themselves (nor is the attempt made) to address specific ways to handle design problems and trade-offs. At the other extreme, very specific guidance may answer some of the designer questions (e.g., how some guidance should be phrased). Hopefully the standardization attempts by the Human Factors Society will bridge the gap between general and specific guidelines, focusing on the development of guidelines to aid in the trade-off decisions that are central to the design process.

Characterizing the Dialog Between Human and Computer.

The key notion of user-computer interaction is that the human and computer participate in a dialog whose purpose is the accomplishment of a task. The aim of the various disciplines studying the field focuses on understanding, evaluating, and improving the way in which humans and computers communicate.

The term most often associated with the interaction between the human and computer is dialog. The term dialog is aptly chosen since it reflects the inherent nature of the stream of symbols which both human and computer must access to accomplish communication (Card, Moran, and Newell, 1983). All mechanisms involved in the dialog are part of the human-computer interaction, including physical devices (e.g., keyboards) and actions taken with physical devices (e.g., keystrokes) as well as computer programs and outputs from the computer (e.g., visual or auditory information).

Given the centrality of the notion of communication, much research has focused on ways to provide orderly exchanges between the human and computer. Research in this area concerns dialog styles (also called "dialog techniques" or "interaction styles"). Dialog styles include command language, menus, question and answer dialogs, natural language, and direct manipulation.

Dialog Styles. Gaines and Shaw (1986b) distinguish three categories of dialog styles. These categories include:

- Formal, such as the prompt-response dialogs of interactive job control languages (e.g., VMS, Unix or CP/M).

- Natural Language, where human use of language is simulated within the narrow context of the data structures and activities of the computer system (e.g., INTELLECT and ASK).

- Graphic, such as the use of windows, icons and desk-top simulations (e.g., Xerox Star and Apple Macintosh interfaces).

Formal dialog styles are direct representations of computer activities and data structures. The structure of formal dialog stems directly from the underlying system. By contrast, natural language is remote from actual system operation. The graphic dialog style is an object simulation where the structure of the dialog stems from the physical world.

Both natural language and graphic dialog styles are inherently attractive since users are likely to have existing skills to bring to the interaction. However, there are numerous problems associated with the use of natural language; these will be discussed later. The attractiveness of formal dialog is that the "user sees what is there" though the demands placed on the user tend to be much greater in terms of system learning requirements. As noted by Gaines and Shaw (1986b), fifth generation systems will focus on the integration of dialog styles (the Macintosh, for example, combines the graphic desk top metaphor with the formal dialog of pull down menus and question and answer dialog boxes).

The literature does not offer an optimum dialog choice (Schvaneveldt, McDonald, and Cooke, 1986). In addition to the potential overhead demands of natural language, Simmons (1986) notes a number of problems associated with the use of natural language, such as:

- Users expect the computer to think as they think.

- Users do not know the degree of understanding possessed by the computer.

- Successful use of natural language requires that users be provided with feedback about misunderstandings.

- Human errors may be reduced by combining natural language with menu selections and prompts; however, this becomes extremely cumbersome if the natural language subset is very large.

- For a system to respond appropriately, the system needs to estimate the user's intention in asking the question.

Many of the problems inherent in the use of natural language revolve around the ambiguous nature of language. The concept of restricted natural language has been proposed in answer to the criticism of ambiguity (Schvaneveldt, McDonald, and Cooke, 1986). The experimental work of Bailey (1985) has demonstrated that restricting natural language, specifically restricting syntax to command-orientation, can improve human performance.

Graceful interaction. Hayes and Reddy (1983) introduce the notion of "graceful interaction" as a requirement of human-computer communication. Graceful interaction refers to the manner in which the computer responds to less than straightforward requests or questions from the user. Hayes and Reddy propose a decomposition of the notion of graceful interaction into a number of independent skills, including the following:

- Robust communication. Those strategies needed to be certain the speaker's utterance is heard and correctly interpreted.

- Flexible parsing. The ability to deal with natural language as used in a natural manner (including ellipses, idioms, grammatical errors, and partial utterances).

- Domain knowledge. A prerequisite for the other components of graceful interaction.

- Explanation facility. System ability to explain system capabilities and limitations, past actions, current actions and why (these would need to be answered in response to user questions and volunteered when communication breaks down).

- Focus mechanisms. System ability to track conversation topics and notice topic shifts.

- Identification from descriptions. System ability to recognize an object from a description (including initiating dialog to clarify as needed).

- Generation of descriptions. System ability to generate descriptions appropriate for context and that satisfy requirements imposed by other components (e.g., robust communication).

Hayes and Szekely (1983) report an undertaking to demonstrate the design concept of graceful interaction. The project was termed COUSIN for COoperative USer INterface. The initial efforts focused on the use of COUSIN at the top command level of an operating system (Unix). COUSIN uses a form-based model of communication with error correction and on-line help. Initial reactions of COUSIN users were extremely favorable particularly concerning error correction and on-line help capabilities. Serious problems arose because insufficient context was provided for users; COUSIN-Unix featured a line-oriented form editor dealing with single fields at a time. Modifications to COUSIN to correct the deficiency were identified; a future version would feature a screen-orientation to preserve context.

INTELLIGENT HUMAN-COMPUTER INTERFACES

Advances in the realm of artificial intelligence have appeared to stimulate many researchers to explore the potential applications of AI to the human-computer interface. While there are many unanswered questions in the area of conventional human-computer interaction, there are even more unresolved questions, as well as more basic issues to confront, concerning intelligent human-computer interfaces.

The Wish List for Intelligent Interfaces.

There is a significant range of characteristics that an intelligent interface might possess. Halpin and Moses (1986) identify the following features of an intelligent interface:

- The system must understand the user.
- The system must know the user's goals and objectives.
- The system must know what tasks are to be performed.
- The system must know the criteria that signify to the user that a task has been accomplished.
- The system must understand the world as the user understands the world.

Rissland (1984) assumes that an intelligent interface would require access to a considerable range of knowledge sources. Rissland identified seven knowledge areas, including:

- Knowledge of the user (including expertise, style, preferences, and history).
- Knowledge of the user's tasks (including context, purpose, and ultimate use of the result).
- Knowledge of the tools (including protocols for invoking and using, default parameters, side-effects, required resources, and usage costs).
- Knowledge of the domain and user's task (epistemological knowledge such as domain and task structure and representation, measures of importance, conceptual hierarchies, and domain-specific concepts).
- Knowledge of interaction modalities (including graphics input and output displays, video, audio, language, and pointing devices).
- Knowledge of how to interact (including when to intervene, monitor, adapt, enter and leave tutoring or coaching modes, length of time to wait for user responses).
- Evaluation knowledge (including how to judge interaction effectiveness and how to act on user complaints).

Rissland notes the impracticality involved in requiring an interface to encompass the full range of knowledge in each area. However, she stresses the need to address each area to some degree.

Schevaneveldt, McDonald, and Cooke (1985) are currently engaged in research to develop a modular user-system interface. These researchers assume that intelligence is "obviously" necessary for the interface. From their perspective, a modular user-system interface is an expert system. The requirements for such an interface include:

- General user knowledge. This knowledge includes facts and data from the fields of cognitive psychology, human factors, and human-computer interaction research. Such knowledge might include memory limitations, perceptual abilities, information processing capabilities, biases in reasoning and decision making, and effective means of communicating` structure and organization to users.

- Specific user knowledge. This knowledge includes information about particular users and populations of users.

- Task and goal knowledge. This knowledge includes goal structures associated with tasks at varying levels of abstraction. This knowledge would be useful for maintaining a sense of context in user interactions with systems.

- Device knowledge. This knowledge includes information about various physical devices that the user actually experiences, knowledge of those devices available to the user, and particular characteristics of the device.

- Dialog manager. The dialog manager is a controller program that directly communicates with both the applications and the user.

Identifying an Intelligent Interface

The literature does not offer a consensus on what constitutes an intelligent interface. Taken literally, some wish lists view the role of an intelligent interface as that of taking on almost "user clone" qualities (e.g., "understanding the world as the user understands it"). Such objectives for an intelligent interface are not likely to be obtainable in the near future. One might even question the desirability of a "user clone;" from the user's perspective an intelligent advisor or assistant may be a more beneficial or desirable role for the computer to serve.

The various wish lists for the intelligent interface do feature commonalities, specifically user models and knowledge bases. In one sense, the need for knowledge bases is prerequisite to user modeling. Further, adaptivity, either within a user or between users, may be viewed as a subset of user modeling (Halpin, personal communication).

An alternative view of the intelligent interface is one that uses artificial intelligence techniques. Techniques include use of high level languages (such as LISP, PROLOG, and INTERLISP) and tools (such as LOOPS and the Knowledge Engineering Environment). This view is quite pragmatic and should not be overlooked when looking at developments in intelligent interfaces.

Developments Toward Intelligent Interfaces

As noted earlier, there is much interest in intelligent interfaces. Interest and wish lists typically precede development and that is certainly the case for intelligent interfaces. The successes reported in the literature have tackled small problems; this may be a somewhat traditional approach for many researchers since the risk is lower and the potential for success higher.

The following sections will highlight the fact that an intelligent interface which aids in the establishment of a partnership between users and computers awaits development. There are efforts underway, which have yet to be completed and hence fully reported, that are attempting to develop intelligent interfaces.

Albert Badre (1984) is in the process of developing the intelligent interface architecture proposed in Hopple and Halpin (1985). The architecture to support an intelligent interface is likely to be completed in the February/March 1987 timeframe.

The architecture proposed by Hopple and Halpin (1985), shown in Figure 1, provides a framework with the necessary components to support dialog between the user and computer. The system supported by the proposed architecture features three subsystems: the user; the interface module; and the hardware/software subsystems. The real world of data upon which the user operates is contained in the hardware/software subsystems. Within this architecture, the interface module serves as a translator between the user and the hardware/software subsystems.

The framework proposed by Hopple and Halpin permits a dialog, rather than a monolog, allowing a mixed-initiative method of computer operation. This architecture requires extensive use of knowledge bases and models by the computer to create an interface which serves as a translator between the user and computer. The interpretation of user commands and requests requires knowledge of user characteristics, user goals, and a model of the current state of the world.

A similar effort is underway at the Computing Research Laboratory at New Mexico State University. One thrust of the research is the development of a modular interface. Expert system technology will be applied to develop an interface-design consultant as well as to implement an intelligent interface.

User models. A user model may be utilized in a number of ways. Greenstein, Arnaut, and Revesman (1986) have demonstrated the utility of employing a model of human decision-making behavior in the dynamic task allocation domain. In this effort, a model was used by the computer to predict what the human would do and the computer then responded accordingly. This model-based approach is in contrast to explicit communication where the human tells the computer what he or she will do. In a comparison of the model-based approach to explicit communication the model-based approach was superior for overall system performance, even though the predictive power of the model was less than perfect (predicted more than 70% of human actions in a process monitoring task).

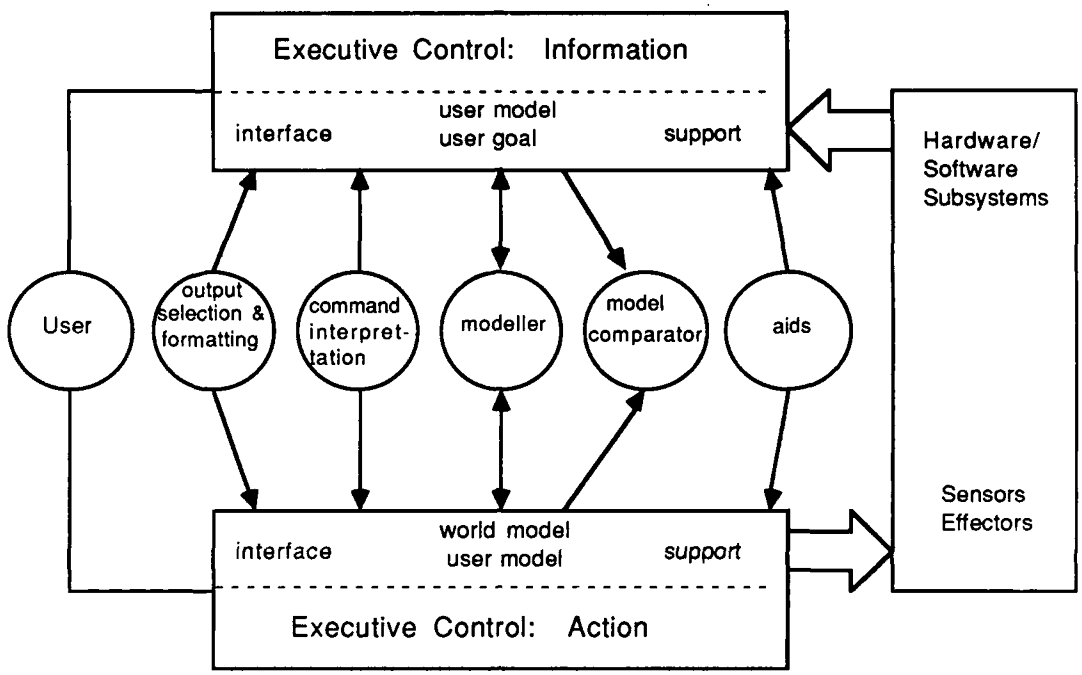

Figure 1. Intelligent interface architecture (from Hopple Halpin, 1985, p. 164).

The notion of a user model may seem straightforward when it is presented in a wish list. However, there are a number of issues in the notion, such as how to obtain the information as well as how to represent the information. The following discussion of user models draws heavily from researchers in the area of tutoring and instructional systems. The rationale for drawing on this literature area is that user models form a basis for teaching and remediation and hence receive considerable treatment by researchers in this area. Researchers in the CAI domain address user models in great depth from the perspective of methods for obtaining the model as well as alternative representation methods. Researchers concerned with user modeling in the user-computer interaction domain might find utility in the classifications and research conducted in the CAI area.

Sleeman (1985) offers the following dimensions for classifying user models:

- Single model of stereotypical user versus collection of models of individual user. This dimension has also been termed canonical models versus individual models (Zissos and Witten, 1985).

- Prescriptive models (specified by the user or designer) versus inferential models (inferred by the system based on user performance). This dimension has also been termed explicit models versus implicit models (Zissos and Witten, 1985).

- Models of long-term user characteristics versus models of the current task.

- The form and nature of information contained in the user model and the inference engine needed to interpret the information. (This dimension appears in the work of Sleeman; the other authors reviewed classify user models according to the previous three dimensions.)

Zissos and Witten (1985) suggest a number of issues associated with the dimensions of users models. The stereotypical user model tends to be less difficult to build than the collection of models. However, the stereotypical user model is of no benefit if the objective is to assist or to coach an individual user. Implicit models may require less specialized knowledge and may save the user time; however, Zissos and Witten (1985) indicate that some researchers contend that such approaches preclude the development of detailed models due to the limited bandwidth between users and computers. Explicit modeling may increase the bandwidth and improve the quality and type of information available. Others point out that some amount of implicit modeling is both possible and frequently desirable. Long-term modeling may appear to be more accurate since more information is collected. One difficulty with long-term modeling is the potential confounding due to the fact that the user being modeled is continually changing; it is also difficult to account for the effects of learning and forgetting.

Sleeman (1985) offers a further distinction of user models in which he focuses on expanding the inferential models. Models developed for a user through inference based on the user's behavior include:

- Scalar models. These models are simple quantitative models.

- Ad Hoc models. These models may, for example, infer a plan based on observation. These models require specialized inference engines.

- Profile models. These are descriptive models, such as matching book attributes to user characteristics.

- Overlay models. These models compare a user model to an expert model on topics that the user intends to acquire.

- Process models. These user models suggest how the user would perform a task. Process models are executable and differ from ad hoc models in that they do not require specialized inference engines.

Elkerton and Williges (1985) discuss representation methods that have been used to represent the user. The choices that developers must make are summarized below.

Symbolic and quantitative models. Symbolic model development stems from work in artificial intelligence and takes on a variety of forms, including rules, frames, scripts, semantic nets, and plans. Symbolic models tend to have a descriptive capacity while quantitative models tend to have predictive capabilities. Elkerton and Williges (1985) maintain that the distinction between symbolic and quantitative models stems from the type of expertise in the model. They characterize symbolic models as "glass-box" experts while quantitative models tend to be "black-box" experts. "Glass-box" experts attempt to perform in a manner similar to humans (such as in problem solving) while the "black-box" experts are not constrained in such a manner.

Performance and cognitive models. Performance models are based on actual human performance (typically quantitative models). Performance models are typically obtained by monitoring user performance and statistically summarizing the data obtained. As Elkerton and Williges note, performance models are difficult to capture for complex tasks, since many complexities may not be apparent from performance measures. The approach taken in the development of cognitive models is the elicitation of expertise from experts in the area of interest. Such a collaborative effort is typically the approach with expert systems, symbolic, and cognitive models (Elkerton and Williges, 1985).

Skilled and unskilled models. Elkerton and Williges (1985) note that models may be constructed to include positive or negative information. While positive information represents skills and behaviors expected of the user, negative information represents inappropriate planning or unskilled performance. Skilled models may serve as a standard against which user performance may be evaluated. The use of skilled models is frequently found in work on computer-aided instruction (CAI) and intelligent CAI. The use of unskilled models has been demonstrated by some researchers to be useful in assisting and training novice users. Elkerton and Williges refer to one research effort that successfully merged skilled and unskilled models to account for the evolving behavior of users.

Static and dynamic models. Frequently, user models are static in nature and allow for little if any change over time. One type of static model is termed a differential model which defines user performance in terms of expert performance. As Elkerton and Williges note, the use of differential modeling may reduce the modeling burden but may be ineffectual in instances of a very large difference between users and experts.

Limited work has been accomplished in the area of dynamic modeling. Elkerton and Williges identify the work of Genesereth in plan recognition as one example of dynamic modeling. Genesereth developed an advisor for a symbolic integration program (MACSYMA) which identified faulty plans and the likely incorrect beliefs that led to the user's inappropriate planning behavior.

Multiple representations. While the use of multiple representations may not be computationally efficient or feasible, there might be an advantage of multiple models for the capability of switching problem spaces if one model does not fit (Elkerton and Williges, 1985). However, there are various problems associated with multiple models. such as the circumstances under which models would switch and the potential need for communication between models.

Zissos and Witten (1985) characterize the representation of user models in three forms:

- Parametric form.
- Discrete-event form.
- Frame-like form.

Parametric representations may be numeric or non-numeric. Zissos and Witten followed the lead of Rowe in modeling item interest in a database by developing a numeric parametric user representation of database use. An example of a non-numeric parametric representation is found in the "alias" facility of interactive operating system (Zissos and Witten, 1985). The "alias" facility uses explicit modeling to permit users to select names to be associated with system utilities. In the case of well defined tasks, parametric models have utility. However, parametric models are not flexible and do not have application to coaching users.

"Discrete-event models arise out of a system-theoretic approach to behavior-to-structure transformation" (Zissos and Witten, 1985). Discrete-event models support analysis of behavior and prediction of inputs by treating user invoked commands as an abstract sequence to be modelled. Zissos and Witten (1985) note that discrete-event models may present difficulties in communication with users, due to the fact that they operate in low-level language sequences of commands.

As the term implies, frame-like models are derived from expert system knowledge representation techniques. With frame-like models, ad hoc structures are used to store information about selected aspects of user performance which are updatable. Zissos and Witten indicate that frame-like models represent a user's understanding of the domain. However, such models are generally imprecise in complex domains since user knowledge is being inferred from behavior. The power or weakness of the use of frame-like models lies in the manner in which information is determined from user actions.

Zissos and Witten identify four techniques for inferring user's knowledge for frame-like models:

- Experience difference.
- Bad plan detection.
- Exhaustive searching.
- Use frequency.

The experience difference technique compares the user's performance against a pre-determined standard, especially expert difference modeling. They note that the value of this technique is high in "domains in which the user's goals are clear and unambiguous, individual actions are easy to isolate from each other, and the sequence of optimal moves, before too long, will exercise all the domain's concepts" (Zissos and Witten, 1985, p. 733).

In less structured domains, the technique termed bad plan detection may be useful. This technique for inferring user's knowledge attempts to identify the user's unfamiliarity with concepts by recognizing pre-stored behavioral sequences. Zissos and Witten note that numerous researchers have used this technique (Anderson and Reiser, 1985; Coombs and Alty, 1984; Rich, 1984; Sleeman, 1982; Shrager and Finn, 1982). Essentially, this technique proceeds as follows: "bad plans" are cataloged by observing various users and analysis of domain semantics; the user's behavior is observed for instances of bad plans. There are numerous positive aspects to this technique, such as computational efficiency and updatability of the catalog of bad plans. However, Zissos and Witten note the serious limitations, namely, they view the technique as being inexact (oversimplification of the task domain), incomplete (cataloging all possible bad plans is impractical), and inflexible (bad plans cannot be detected if they are not contained in the catalog).

A third technique for constructing frame-like models is exhaustive searching of multiple models. This technique has application in restricted domains. Zissos and Wittens note the use of this technique by Sleeman (1982). In this application, a frame-like CAI system built numerous user models (including relevant combinations and permutations of understood and misunderstood concepts). The set of models was explored to identify the model that best fit observed behavior. While this technique may have utility in a limited CAI application, it is likely to be unsuitable for any complex domains.

General evidence of user understanding can be obtained in a fairly simple manner using the fourth technique identified by Zissos and Witten. This technique is based on an analysis of the frequency of use of concepts. The authors note that significant information about the user may be captured by identifying concepts that he or she does not use. This subset of information may provide an opportunity for coaching the user. The authors note that further subdivisions of the "not used" concepts, particularly into those concepts not needed or known but not likely, is rather difficult without the direct engagement of the user in a dialog (thus introducing explicit modeling, which the authors were trying to avoid).

Zissos and Witten (1985) developed an active user assistance or coaching prototype for the EMACS text editor (Stallman, 1984), termed Anchises. They modeled users in two manners: not used concepts and bad plan detection. The not used concepts criterion was demonstrated to be highly accurate. The bad plan detection was found to be of high utility for identifying "user inefficiency in particular circumstances, spotting specially-defined bugs, compensating for noise, and encouraging use of newly-learned concepts" (p. 748). The authors conclude that individual bad plans are not totally productive given the difficult and tedious tasks of cataloging.

Adaptivity. Adaptivity of a computer to a user may be viewed from two perspectives. First, adaptivity may focus on an individual user. Second, adaptivity may focus on adapting to various users, adaptation among multiple users. When one discusses adaptivity for an individual user, one is frequently concerned with the fact that the user and his skill level with the computer are not static; with experience the user transitions from novice to experienced skill levels (and possibly to expert). Adaptivity is concerned with creating a system that is not too difficult for the novice user to learn and use ("friendly" or "natural") while neither too boring nor cumbersome for the experienced or expert user. Accommodating the extremes on the continuum is typically the reason for the desire for adaptivity.

Badre (1984) is particularly concerned with designing the user-computer interface to accomodate user transitions. Badre points out that the novice user tends to rely on recognition memory strategies to accomplish his tasks with the computer. An expert, on the other hand, may need to chunk, summarize, or abstract the actions which might be taken by the novice in various contexts. Badre recommends that the expert interface mode accomodate a free recall of the interaction language. Assuming that a novice user may transition into an expert through practice, Badre (1984) recommends the following conditions in the design and implementation of the interface:

- Two different modes are needed to allow for the performance of a single task (expert and novice modes).

- All novice user actions should be capable of being accomplished using recognition memory strategies.

- Expert interaction should depend upon the free recall of interaction.

- As a byproduct of practice, the novice should have the opportunity to learn the expert's vocabulary and syntax.

- The expert level language of interaction should permit the expert to bypass novice strategies and introduce new vocabulary and generate new sentences to support summarizing and chunking activity.

Transitionality has been implemented into the Infoscope information management system and meets the conditions Badre established. Six modes of interaction are possible in Infoscope: menu prompting; command name prompting; command statement; command statement abbreviation; definitionals; and command files.

Further research is being undertaken to establish potential transitionality patterns that may be appropriate to user levels and task complexity; such patterns could permit "transitionality prompts" to be built into the system. This could permit prompts to be automatically developed to the user at appropriate times. In this manner, profiles would be developed for specific user/task combinations.

A very different view of adaptivity may be found in the work of McCracken and Akscyn (1984) on the ZOG interface system. The interface is not adaptive in the sense that many researchers tend to view adaptivity. However, experimental results with ZOG indicate that the system accommodates both extremes of the novice-expert continuum. How ZOG manages to support the continuum of users is of interest. The Carnagie-Mellon University researchers offer the following explanations about why experienced users do not "outgrow" ZOG:

- ZOG is a fast responding system. The menu selection interface is competitive with a command language interface.

- ZOG permits experienced users to use a broader range of functions than novices.

- ZOG is efficient in developing additional structures. (a chunk of ZOG data is a frame or tree of frames with variable parts - this scheme may be instantiated by copying and providing values for the parts.)

From the novice perspective, basic ZOG use may be learned in less than half an hour. The ability to add new material is typically achieved within an additional two hours.

Though ZOG was originally intended as a research vehicle to study user-computer interaction, the ZOG interface has been applied on-board the U.S.S. Carl Vinson.

Applications in this domain include: document preparation; interaction with other medias such as video disk material; and interface to an expert system (AirPlan).

Slator, Conley, and Anderson (1986) have developed an adaptive natural language interface to a graphics package ("Grafit" by the Graphicus Corp. running on the HP-1000). User models are formed from observation of user action. In current use, only a single model for a collection of users has been implemented; future implementations will focus on the development of models for individual users.

One of the design objectives for the Slator, Conley, and Anderson (1986) research was to implement an interface which would teach users to use a graphic command language. The natural language interface does not support a "conversational" dialog between the user and the computer. Rather, the user issues commands and observes whether or not the command is carried out. While this approach attempts to minimize misconceptions on the user's part about what the computer understands, ambiguities still arise. To resolve ambiguities the script-like decision matrix is used to hypothesize what the user intends on the basis of the actions the user has taken in the past.

Use of knowledge bases. Developments in the use of knowledge bases are quite extensive. The artificial intelligence literature is filled with examples of the use of knowledge bases; no attempt will be made to review all of the developments reported; instead, a few examples will be presented.

Waters (1984) describes a programmer's apprentice which uses knowledge bases in the domain of program editing. Winograd (1984) describes the future direction of the programming environment as heavily knowledge-based. Wilensky (1983) describes numerous knowledge base examples in the area of text understanding. Shen and Signarowski (1985) discuss the knowledge base needs for intelligent flexible control in the area of programming and control of robots.

Rissland (1984) describes a knowledge-based on-line HELP and tutoring system and points out that the knowledge requirements for on-line HELP may be quite varied. Determining what help the user requires (after invoking HELP) requires a minimum amount of knowledge about the tool to be explained and the interaction modalities (some contextual knowledge of the user and domain may be needed). Responding in an intelligent manner to the user, custom-tailoring for example, may require more extensive knowledge of the user, domain, and task. Tailoring could be accomplished using pre-stored examples derived from expert knowledge of the topic to be explained. A more elaborate approach would customize the pre-stored examples.

AI Tool Use in Interfaces. This section highlights some studies which have already been discussed in earlier sections. However, here the orientation is toward AI tools that have been applied to user-computer interaction.

The ZOG interface (McCracken and Akscyn, 1984), discussed earlier, uses a frame as the basic unit of representation. Typically, the ZOG database contains tens of thousands of frames. The frame viewed by users contains a set of items of different types and positioning information about each. For example, local pads are tangential points (cross references to related material in other documents, invoking programs); global pads are general functions (back, help, etc.). A selection on a ZOG frame leads to an associated action. One interesting feature of ZOG is that frame size is restricted to the size of the available display; scrolling is not permitted. If information cannot be contained in a single frame display, the frames are linked in a linear sequence or extra levels of hierarchy are introduced.

The work of Slator, Conley, and Anderson (1986) has been briefly discussed in an earlier section. The development of an adaptive interface to a graphics package uses a script-like decision matrix for modeling user behavior. The natural language understanding aspect of the interface uses a semantic parser which determines the grammar rules that apply to a user utterance (three possible cases - none apply, one applies, or more than one apply). Production rules are used by the interface as an intermediate step to develop a list of candidate commands that the user may mean by the utterance he enters. The script used by the interface is dynamic since there is no ordering of graphic events in this domain. The research is on-going in an effort to fine tune the interface. For example, given the non-determininistic nature of the decision mechanism, mistakes occur. Enhancements to the interface will include eliciting feedback from the user in regard to uncertain translations, allowing the user to approve or disapprove the results of the translations.

The Information Lens (Malone, Grant, and Turbak, 1986) is an intelligent system for information sharing in a multi-user environment. The Information Lens was developed in the Interlisp-D environment using Loops and runs on Xerox 1108 processors connected by an Ethernet. The objective of the Information Lens was to develop a prototype system to aid users in sharing information communicated via electronic messaging systems. The system uses frames, production rules, and inheritance networks to accomplish the desired information sharing.

The Information Lens system is based on frames (a set of semi-structured message types). Each message type has an associated template with slots for information. Each slot has numerous properties, including default values, potential values, and an explanation of the reason the slot is part of the template. Users may directly access slot information in the process of constructing a message.

The templates used in the Information Lens are linked in a network. This enables subtypes of a template to inherit slot information from the parent template. This network structure is visible to the user in a message type browser.

User rule building is permitted in the Information Lens for finding, filtering, and sorting messages. Rule templates based on the same message types used for rule construction are utilized, primarily as an aid to the inexperienced user. In addition, users may specify rules to access messages not directly addressed to them (messages sent to a common file "Lens").

THE DIFFICULTIES OF DEVELOPING INTELLIGENT INTERFACES

While there is a lack of consensus about the definition of an intelligent interface, researchers point to a variety of needs that will influence the development of the intelligent interface. Halpin and McCullough (1986) and Gaines and Shaw (1986b) identify the need to develop a symbiotic relationship between humans and computers; Norman (1986) identifies a need for making systems more pleasurable; Brown identifies the need for increasing the intelligibility of systems. The common theme is the characterization of the user-computer interaction as a dialog, a conversation between the user and the computer. Viewing interaction as a dialog highlights the need for drawing communication research into the interaction design picture.

The difficulties that are likely to confront the development of intelligent interfaces will undoubtedly emerge as development efforts focus on complex domains. However, two particular areas are apt to be especially difficult, understanding language and mental modeling. The difficulty of understanding language need not be viewed as a need for natural language understanding; rather, the view is that a dialog requires understanding of the language by both the speaker and the listener regardless of language type.

The notion of mental models is pervasive in much of the current literature concerned with developing a partnership between humans and computers. At the same time, the notion of mental models presents difficulties in practical application. Capturing and representing mental models is a special case of knowledge acquisition and representation.

Understanding Language

Drawing from the work of Kraut and Higgins (1984), Halpin and McCullough (1986) present six proposed maxims to govern the behavior of speakers in a conversation:

1. Maxim of quantity: speakers should make their contributions as informative as required, providing neither too much nor too little information.

2. Maxim of quality: speakers should convey the truth as they see it.

3. Maxim of relation: speakers should produce a message that is appropriate to their communicative intent or purpose.

4. Maxim of relevance: speakers should produce a message that is relevant to the context and circumstances.

5. Maxim of manner: speakers should try to be clear, coherent, comprehensible, and concise.

6. Maxim of sensitivity: speakers should adapt their messages to their listeners' characteristics, including what they believe their listeners know. (Kraut and Higgins, 1984, p. 90) (taken from Halpin and McCullough, 1986, p. 273)

As Halpin and McCullough (1986) point out, these maxims could serve as rules to tailor computer output; however, conversation features both the speaker and the listener. The listener is not a passive partner in the communication process. Thus, the role of feedback is needed to ensure the understanding and interpretation of the speaker's information.

Terry Winograd (1985) has been involved in research in the area of designing computer systems to "understand language" for some time. In his research efforts Winograd has had to attempt to understand what understanding language is about, be it a human or computer who is doing the understanding.

As noted earlier in this paper, implementing natural language understanding in a machine has met with significant difficulties. However, Winograd's work with SHRDLU successfully demonstrated the possibility of a seemingly natural exchange between a user and computer. As Winograd (1985) readily points out, SHRDLU focused on a limited domain with carefully constrained dialog; in addition, there were numerous shortcomings, such as the ad hoc representation of the internal structure of the speaker/hearer.

Winograd (1985) describes two different approaches to understanding language, objectivity and subjectivity. From an objectivity starting point, each utterance has meaning due to its correspondence to a state of affairs. This approach studies language by analyzing how the structures of utterances correspond to what they describe. The subjectivity starting point, on the other hand, views each utterance as a trigger of processes within the hearer through which meaning is determined. This approach studies language through the study of cognitive structures and processes. Winograd sees difficulties with each extreme; there is a subjective aspect to understanding but taken to the extreme one could not account for the fact that people can share meaning in a way without having identical histories.

Winograd (1985) identifies four domains of understanding:

- Domain of linguistic structure. The traditional domain of linguistics looks for regularities in the patterns of structural elements in utterances.

- Domain of correspondence between linguistic structures and the world. This tends to be the domain of AI research in natural language where the concern is with the regularities in the correspondence between linguistic object structure and the world those objects describe.

- Domain of cognitive processes. Both cognitive psychologists and AI researchers are concerned with this domain. The concern of this domain is with the cognitive structures and processes of the generators and interpreters.

- Domain of human action and interaction. This is the current direction of Winograd's work. "In this domain the relevant regularities are in the network of actions and interactions within a human society. An utterance is a linguistic <u>act</u> that has consequences for the participants, leading to other immediate actions and to commitments for further actions." (Winograd, 1985, p. 185-186)

While Winograd acknowledges that his new direction appears to turn away from cognitive psychology, he maintains that is not the intent. Rather, he is adopting a new direction to focus on articulating the unstated background and interactive environment in which utterances occur. The eventual contribution of Winograd's research direction may have application to the development of a dialog between humans and computers. In particular, the Winograd "action" orientation may serve as an approach to limiting the context requirement for understanding.

Mental models

The following discussion concerning mental models raises serious questions in the area of knowledge elicitation and representation. A number of concerns about the search for elusive mental models exist. The work of some researchers could lead one to the conclusion that eliciting mental models from individuals is not possible. The problems raised by these researchers are not merely linguistic; rather they are fundamental to research questions and methodologies. Maybe the search for mental models has been overly ambitious, or maybe the questions have not focused on the obtainable.

The notion of mental models and their role in human-computer interaction has spread throughout numerous research domains (Rouse and Morris, 1986). As Donald Norman (1983) notes:

> In interacting with the environment, with others, and with the artifacts of technology, people form internal, mental models of themselves and of the things with which they are interacting. These models provide predictive and explanatory power for understanding the interaction. (p 7)

Recently, Rouse and Morris (1986) conducted a review of the current state-of-the-search for mental models. The authors found that while the term "mental models" is used extensively, there are few explicit definitions. Some of the defintions they found included:

- Hypothetical construct (in the domain of sampling, scanning, and planning).
- Special cases of schema (in the domain of manual control).
- Understanding that is characterized as messy, sloppy, incomplete, and indistinct (in the domain of understanding systems).
- Understanding of knowledge representation in rules, rule organization as a network of relations, and explanatory tracing involving network chaining (in the domain of expert systems).
- Internal structure and processes of the device (in the domain of device operation).

One problem, according to Rouse and Morris (1986), involves distinguishing mental models from general knowledge. They assert that mental models are special types of knowledge. They note that without a more precise definition of mental models "there is a great risk that the result of research in this area will simply be that humans have to know something in order to perform their tasks. Clearly, this result will not be a great stride for science." (p. 350)

Rouse and Morris (1986) reviewed the various purposes for which mental models are used. They concluded that there are three common themes regardless of the actual task. These themes are describing, explaining, and predicting. From this the authors offer a functional definition of a mental model:

> Mental models are the mechanisms whereby humans are able to generate descriptions of system purpose and form, explanations of system functioning and observing system states, and predictions of future system states. (p. 351)

This functional definition avoids the general knowledge issue by stating the types of knowledge and their purpose. Further, a distinction is not made between knowledge that is retrieved from knowledge acquired through computation (the authors' stated implication being that mental models need not be computational models).

Capturing a Mental Model. Methods for capturing an individual's mental model have been discussed from the perspective of user models in an earlier section. Focusing specifically on mental models Rouse and Morris (1986) categorize the approaches to their identification in the following manner:

- Inferring the model via empirical study.
- Empirical modeling (inferring the structure and parameters of a mental model by the algorithmic identification of the relation between an individual's observations and subsequent actions).
- Analytical modeling (deriving the mental model for a particular task using available theory and data).
- Verbal and written reports (introspection and verbal protocols).

As Rouse and Morris (1986) note, there are weaknesses inherent in each technique and limitations on the domains they are used to explore. They maintain that inferential methods are useful modeling techniques in tasks where the choice behavior of the subject is limited (e.g., manual control) as opposed to tasks such as problem solving and decision making. Their rationale is that the mental model may be based on external factors when the level of behavioral discretion is low.

Rouse and Morris (1986) hypothesize that the mental model is more likely to be captured in tasks where the model is explicitly manipulated. If the use of the model is part of performing the task, it may be captured with verbal or written reports (if it can be verbalized). It should be noted, however, that the use of introspection or verbal protocols in the search for mental models has its critics (such as, Nisbett and Wilson,1977, who maintain that humans cannot report their higher level processing, a finding not surprising to many knowledge engineers who attempt to elicit expertise from experts.).

Numerous researchers in both cognitive psychology and artificial intelligence are concerned with capturing the mental models of experts. Eliciting knowledge from experts, toward the goal of developing mental models or expert systems, is difficult at best. Verbalizing expertise may be an impossible task for an expert. Worse, as noted by Rouse and Morris (1986), is that "their verbalizations may reflect what they expect the inquirer wants rather than how they actually perform" (p. 353). It is necessary to look at expert performance as well as their verbalization about expertise to identify discrepancies that must be resolved.

An additional problem is beginning to surface from experimental evidence that raises questions about the nature of expertise. Rouse and Morris (1986) identify numerous studies which conclude that expertise may be conceptually different rather than a refinement of novice knowledge. Various researchers are describing the shift from novice to expert in the following manners:

- Models shift from representational to abstract (Larkin).
- Expert models as pattern-oriented representations (Chase and Simon; Dreyfus and Dreyfus).
- Expertise differentiated from novice by superior knowledge organization (Glaser; Pennington).

Accepting these conclusions leads one to question the accessibility of expertise via verbalizations. An effort by Klein, Calderwood, and Clinton-Cirocco (1985) studied the rapid decision making of experts in fire ground decision making (i.e., decisions made by fire chiefs during a fire). Their study suggests the important role of perceptual cuing rather than option generation, a pattern that appears to characterize the type of decision making used by experts in highly time-pressured situations. Recognitional and perceptual matching processes could account for the experts' difficulty in articulating decision bases.

Representing a Mental Model. The representation of a captured mental model has associated with it two concerns: what does the model look like and what is the context in which the model applies. The first issue deals with the spatial-verbal aspect of the model (Rouse and Morris, 1986). As Rouse and Morris (1986) state:

> It is likely that our information processing system is particularly adept at processing spatially oriented information, and, hence, may tend to store information in that manner. Therefore, it seems reasonable to suggest that mental models are frequently pictorial or imagelike rather than symbolic in a list-processing sense. (p. 355)

The representation of verbal models is not even straightforward. Greeno (1983) discusses the notion of an ontology of a domain for representing problem situations. He maintains that the ontology aspect of representation is an important and influencing factor in problem solving. He characterizes the influence of ontology on problem solving in the following manner:

- Facilitates the formation of analogies between domains.
- Enables the use of general reasoning procedures.
- Provides efficiency.
- Facilitates planning.

Rumelhart and Norman (1985) address the general issue of knowledge representation as a problem in notation. Their perspective is that knowledge representation theories attempt to establish a notation which can represent all relevant structures and processes and one in which those processes assumed to be easily carried out can be easily carried out. As Rumelhart and Norman (1985) note, there are three major areas of controversy in the area of knowledge representation relating to representation format:

- Propositional - Analogical. Propositional representations attempt to represent the world through formal statements (networks, schema-based structures, logical formulae). Analogical representations attempt to directly map the important characteristics of the real world and the representing world.

- Continuous - Discrete. This controversy relates to the granularity of the representation and reflects those aspects of the real world considered important for the representation.

- Declarative - Procedural. This controversy concerns the accessibility of information to the interpretive structures. Declarative representations allow examination and manipulation directly by the interpretative processes. By contrast, procedural representations are not directly accessible. Thus, the controversy concerns the explicit-implicit representation notion which the authors contend is a context-dependent issue.

Johnson-Laird (1985) relates the notion of mental models to representation forms. He proposes the existence of at least three types of mental representations. Propositional representations are symbol strings corresponding to natural language. Mental models are structural analogs of the world. Images are perceptual correlates of models from a particular viewpoint.

Rouse and Morris (1986) raise the issue of the context of representation in terms of its general or specific nature. They note that the literature appears to be converging on the conclusion that mental models tend to be specific. They point out that "if specific representations predominate, it is difficult to account for the richness of human problem-solving behavior" (p. 355).

Limitations on the Search. From their review of work in the area of mental models, Rouse and Morris (1986) suggest some fundamental limitations on the search for mental models. One is that the truth will not be found with complete certainty. When attempting to identify a mental model, scientists are making conceptualizations about another person's conceptualizations about the real world. The subjectivity of the scientist is an influence which cannot be totally removed, though the influence may of course be reduced. An additional limitation stems from the nature of measurement; by intruding on a model to "measure" it, scientists are impacting the evolutionary nature of the model. These limitations tend to be fundamental to science in general.

Rouse and Morris (1986) urge that we abandon our search for ultimate truth and adopt a pragmatic approach to the search for mental models. The authors suggest that research focus on areas such as accessibility, form and content of representation, nature of expertise, cue utilization, and instructional issues.

SUMMARY AND CONCLUSIONS

The topic of user-computer interaction has been and will continue to be addressed by a multi-disciplinary field of researchers. The varied nature of the disciplines presents a difficulty in developing a fully integrated view of the state-of-the-art. Individual researchers tend to focus on issues related to their discipline, sometimes to the exclusion of relevant issues from other fields. Human factors researchers, for example, may address perception but fail to consider the implementation costs; and conversely, interface design engineers may show little appreciation for cognitive aspects of the end user. Such obvious problems demonstrate why multi-disciplinary teams are useful in system design; also, this problem demonstrates the difficulty in finding answers to distinct design questions within the literature.

The literature dealing with issues related to user-computer interaction is vast. Even though much has been written on the topic, there are numerous unresolved issues to be explored if the user and computer are to operate in a cooperative partnership. While traditional human factors research has provided a large body of findings relevant to the design of the interface between humans and computers, we need to move beyond the notion of the human as a passive perceiver in front of a terminal. Though designers must be concerned with when and how to use color, for example, the new research questions are concerned with a problem space more akin to cooperative problem solving and decision making involving both the human and the computer. This directional change in research questions adds new complexities to an already complex design problem.

The extensive experimental evidence compiled by human factors researchers should not be discarded or forgotten. However, this work does suffer from a lack of standard terminology and lack of an organizational framework that would improve the utility of research results to designers. As noted earlier, design guidelines are contained in many locations within the literature, including technical reports and a wide range of journal articles. It may be too much to expect a designer to be familiar with all the existing guidelines. While it may seem too easy to point to automation as the answer, access to on-line interface design guidelines would probably be a boon for designers. Such an undertaking would not be a minor task, but the potential payoff would be high by inserting empirically derived design data into the overall system design. One approach to this might be to combine guidelines with a system building tool, such as windows or an expert system shell.

An often stated goal of many researchers is to bring the user into the system design process early. This is particularly relevant when introducing automation into areas previously manual in operation. In such cases there is likely to be an inadequate experience base on which to base automation requirements; the introduction of automation may alter the manner in which the job is accomplished, possibly in ways not foreseen during a requirements analysis study. Rapid prototyping techniques allow for the development of interactive systems which permit end users to participate in the design iterative process by validating or modifying requirements. Rapid prototyping techniques such as storyboarding offer relatively low cost approaches to bringing the user into the design process. One important direction for future development is to bring extensive automation into the design process itself, using knowledge based tools to both speed and improve the design process.

Much interaction research focuses on the "user", meaning the end-user. There are typically a range of classes of users to consider; these classes may reflect a single individual, wearing various user "hats", or multiple individuals. Obviously, one must be concerned about the design of the interaction between the end-user and the system. There is a "developer-user" who is the user of the operating system development tools, involving a variety of tasks including debugging and system evaluation. There is a "system-operator-user" who is responsible for database maintenance, and system reliability. There is a "security officer-user" who is concerned with access, database protection, survivability under failures, and audit trailing. Additionally, there is a "trainer-user" who may be concerned with the system turnover to the target population of end-users. The 'trainer-user" would be concerned with training system use. Basically, this user would serve as a translator or interpreter of system documentation and the interface operation, including database use, system processes and procedures, and the like. Each user type is a real user during various phases of system design and development. From this survey of the literature, little attention appears to be given to users other than the end-user.

Considering a range of users in the development of the interface could lead to the need for robust tools to serve the many users, including the end-users. Robust navigational aids might have more utility to the range of users than specific controlled dialogs using hierarchical menus or scripts. Such aids could permit users to select the "next step" while tracking what the user has done and what remains to be done. Maintaining an audit trail could permit the system to hypothesize the user's next step based on patterns of task performance; such an analysis could be primed by the "developer-user" or "trainer-user" but modified by any user. Development of sophisticated tools for system design will require care in their design to maximize their utility to a very different class of users.

The development of intelligent interaction between humans and computers poses both practical and theoretical problems to researchers. The understanding of meaning confronting the use of natural language is significant. How or if the problems will be resolved is within the realm of educated guesses, or possibly mystics. Trying to bound the problem into a manageable space is likely to be a useful step, as Winograd appears to be suggesting with his "words as actions or promises of actions" approach. While communication research may offer suggestions on how to effectively communicate, this guidance serves more as an evaluative tool than as a bound on the form and content of communication requirements.

One lesson that may be drawn from the literature is that the key to success is to address definable problem spaces. This is particularly true in the elicitation and representation of mental models. As the previous discussions have highlighted confusions exist between mental models and knowledge in general. The mental model component required for intelligent interaction must be definable in a manner that allows for both elicitation and representation. Some researchers are trying to approach the problem in a more practical and bounded manner.

The use of causal schemas is likely to be a useful and successful direction for future intelligent interaction research. The intelligent interface work of Hopple and Halpin (1985), for example, is derived from schema theory. A schema is a knowledge structure expressed in non-propositional terms (Geiselman and Samet, 1980). A schema represents a definable mental model with utility in both the knowledge representation and elicitation areas. Approaching the mental model problem via schema theory, in particular as a causal schema, may address some of the concerns raised in the literature. For one, causal schemas may be elicited; as Nisbett and Wilson point out, it is causal connections which humans are able to report (in contrast to the inability to report on the more general cognitive processes). Second, the notion of causal schema may involve pattern matching and possibly multi-dimensionality, and these are more consistent with emerging notions in the characterization of expertise.

A diverse body of research is likely to remain applicable to the design of user-computer interaction. In attempting to create a symbiotic relationship between users and computers, both basic and applied research in the area of communications is likely to be relevant to aid in identifying the elements needed for shared understanding. Group dynamics research as well as distributed decision making work may provide relevant clues to developing a working partnership between users and computers. There is an obvious challenge to the user-computer interaction researcher to identify and track the research developments from a wide range of fields that have relevance to this line of study.

REFERENCES

Badre, Albert N. (1984). Designing transitionality into the user-computer interface. In G. Salvendy (Ed.), Human-Computer Interaction, Amsterdam: Elsevier Science Publishers.

Bailey, Gregg D. (1985). The effect of restricted syntax on human-computer interaction. Memoranda in Computer and Cognitive Science, Computing Research Laboratory, New Mexico State University.

Bannon, Liam J. (1986). Issues in design: some notes. In D. A. Norman and S. W. Draper (Eds.), User Centered System Design, Hillsdale, NJ: Lawrence Erlbaum Associates.

Brown, John Seeley (1986). From cognitive to social ergonomics and beyond. In D. A. Norman and S. W. Draper (Eds.), User Centered System Design, Hillsdale, NJ: Lawrence Erlbaum Associates.

Card, Stuart K., Thomas P. Moran, and Allen Newell (1983). The Psychology of Human-Computer Interaction. Hillsdale, NJ: Lawrence Erlbaum Associates.

Coombs, M. J. (1986). Artificial intelligence foundations for a cognitive technology: towards the co-operative control of machines. Memoranda in Computer and Cognitive Science, Computing Research Laboratory, New Mexico State University.

Elkerton, Jay and Robert C. Williges (1985). A performance profile methodology for implementing assistance and instruction in computer-based tasks. International Journal of Man-Machine Studies, 23, 135-151.

Gaines, Brian R. and Mildred L. G. Shaw (1986a). From timesharing to the sixth generation: the development of human-computer interaction. Part I. International Journal of Man-Machine Studies, 24, 1-27.

Gaines, Brian R. and Mildred L. G. Shaw (1986b). Foundations of dialog engineering: the development of human-computer interaction. Part II. International Journal of Man-Machine Studies, 24, 101-123.

Geiselman, Ralph E. and Michael G. Samet (1980). Summarizing Military Information: an application of schema theory. Human Factors, 22, 6, 693-705.

Greeno, James G. (1983). Conceptual entities. In Gentner and Stevens (Eds.) Mental Models. Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Halpin, Stanley M. and Gary W. McCullough (1986). User-computer interaction. In Joseph Zeidner (Ed), Human Productivity Enhancement Training and Human Factors in Systems Design Volume 1. New York: Praeger.

Halpin, Stanley M. and Franklin L. Moses (1986). Improving human performance through the application of intelligent systems. Signal, 40, 10, 96-99.

Hayes, Philip J. and D. Raj Reddy (1983). Steps toward graceful interaction in spoken and written man-machine communication. International Journal of Man-Machine Studies, 19, 231-284.

Hayes, Philip J. and Predro A. Szekely (1983). Graceful interaction through the COUSIN command interface. International Journal of Man-Machine Studies, 19, 285-306.

Hopple, Gerry W. and Stanley M. Halpin (1985). Artificial intelligence for C3I: the design and development of a prototype causal schema knowledge-based system. Expert Systems in Government Symposium, IEEE Computer Society, 160-168.

Johnson-Laird, Philip N. (1983). Mental Models. Cambridge, Massachusetts: Harvard University Press.

Klein, Gary, Roberta Calderwood, and Anne Clinton-Cirocco (1985). Rapid decision making on the fire ground. Klein Associates,Inc. Contract MDA903-85-G-0099.

Kraut, R. E. & Higgins, E. T. (1984). Communication and social cognition. In R. S. Wyer and T. K. Srull (Eds), Handbook of Social Cognition. Hillsdale, NJ: Lawrence Erlbaum Associates.

McCracken, Donald L. and Robert M. Akscyn (1984). Experience with the ZOG human-computer interface system. International Journal of Man-Machine Studies, 21, 293-310.

Malone, Thomas W., Kenneth R. Grant, and Franklyn A. Turbak (1986). The information lens: an intelligent system for information sharing in organization. CHI'86 Proceedings, 1-8.

Miller, Dwight P. (1981). The depth/breadth tradeoff in hierarchical computer menus. Proceedings of the Human Factors Society - 25th Annual Meeting, 296-300.

Nisbett, Richard E. and Timothy DeCamp Wilson (1977). Telling more than we can know: verbal reports on mental processes. Psychological Review, 84, 3, 231-259.

Norman, Donald A. (1986). Cognitive Engineering. In D. A. Norman and S. W. Draper (Eds.), User Centered System Design. Hillsdale, NJ: Lawrence Erlbaum Associates.

Peachey, Darwyn R. and Gordon I. McCalla (1986). Using planning techniques in intelligent tutoring systems. International Journal of Man-Machine Studies, 24, 77-98.

Revesman, Mark E. and Joel S. Greenstein (1986). Application of a mathematical model of human decisionmaking for human-computer communication. IEEE Transactions on Systems, Man, and Cybernetics, 16, 1, 142-147.

Rissland, Edwina L. (1984). Ingredients of intelligent user interfaces. International Journal of Man-Machine Studies, 21, 377-388.

Rouse, William and Nancy Morris (1986). On looking into the black box: prospects and limits in the search for mental models. Psychological Bulletin, 100, 3, 349-363.

Rumelhart, David E. and Donald A. Norman (1985). Representation of Knowledge. In Aitkenhead and Slack (Eds.) Issues in Cognitive Modeling. Hillsdale, New Jersey: Lawrence Erlbaum Associate.

Schvaneveldt, Roger, James McDonald, and Nancy Cooke (1986). The user interface in computer systems: a research program. Memoranda in Computer and Cognitive Science, Computing Research Laboratory, New Mexico State University.

Shen, Helen C. and Gary F. P. Signarowski (1985). A knowledge representation for roving robots. IEEE.

Simmons, Robert F. (1986). Man-machine interfaces: can they guess what you want? IEEE Expert.

Slator, Brian M., Walt Conley, and Matthew P. Anderson (1986). Towards an adaptive front-end. Memoranda in Computer and Cognitive Science, Computing Research Laboratory, New Mexico State University.

Sleeman, D. (1985). UMFE: a user modelling front-end subsystem. International Journal of Man-Machine Studies, 23, 71-88.

Smith, Sidney L. and Jane N. Mosier (1984). Design guidelines for user-system interface software. Project No. 522A (The Mitre Corporation). Prepared for Deputy for Acquisition Logistics and Technical Operations, Electronic Systems Division, Air Force Systems Command, United States Air Force, Hanscom Air Force Base, MA; contract no. F19628-84-C-0001.

Stallman, Richard M. (1984). EMACS: the extensible, customizable, self-documenting display editor. In Barstow, Shrobe, and Sandewall (Eds.) Interactive Programming Environments. New York: McGraw-Hill Book Company.

Waters, Richard C. (1984). The programmer's apprentice: knowledge based program editing. In Barstow, Shrobe, and Sandewall (Eds.) Interactive Programming Environments. New York: McGraw-Hill Book Company.

Wilensky, Robert (1983). Planning and Understanding A Computational Approach to Human Reasoning. Reading, Massachusetts: Addison-Wesley Publishing Company.

Winograd, Terry (1985). What does it mean to understand language? In Aitknehead and Slack (Eds.) Issues in Cognitive Modeling. Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Winograd, Terry (1984). Beyond programming languages. In Barstow, Shrobe, and Sandewall (Eds.) Interactive Programming Environments. New York: McGraw-Hill Book Company.

Zissos, Adrian Y. and Ian H. Witten (1985). User modelling for a computer coach: a case study. International Journal of Man-Machine Studies, 23, 729-750.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A

An Annotated Bibliography of Recent Literature Relating to User-Computer Interface Technology

---

This appendix contains summaries of fifty-seven publications relevant to user-computer interface technology. Most of these have been published since 1985. Although it is by no means an exhaustive bibliography, the reader will find it a useful synopsis of the state-of-the-art. The publication summaries are arranged alphabetically by author.

Ahituv, Niv (December 1980). A systematic approach toward assessing the value of an information system. <u>MIS Quarterly</u>, 61 - 75.

Ahituv presents a multiattribute utility assessment approach for assessing the value of an information system. There are a number of problems inherent in this domain. First, there are conceptual problems about value (whose value? individual, team, organization, groups of individuals. what type of value? real or perceived. who is performing the evaluation and when? decision maker, user evaluating continuously or ex-post.). Second, there are problems associated with utility since it is not a undimensional function and that multiple dimensions may be related.

The use of multiattribute utility assessment is seen as a viable approach to the problems, though it is not without problems of its own. MAU requires the identification of relevant attributes, however, attributes may not always be relevant across cases. The definition and measurement of attributes may be nontrivial with regard to some attributes. Also, there is the problem associated with determining the effect of each attribute on the utility function (identifying linear versus threshold functions). A particularly complex problem may be found in identifying the joint utility function which is derived from all the relevant attributes. Ahituv does not profess to have developed the ultimate method for using MAU, rather his approach is limited to the evaluation of information systems.

The information system assessment discussed in the article is confined to the reporting system aspect. The starting point is an established collection of data which is validated and updated. From this point, feasible operations are limited to selective retrieval, data manipulation, and display. Three categories of attributes are identified and defined: timeliness, contents, and format. Each category is decomposed into defined attributes and their utility functions are described.

Athappilly, Kuriakose and Ron S. Galbreath (February, 1986). Practical methodology simplifies DSS software evaluation process. <u>Data Management</u>, 10 - 28.

This articles offers an overview of decision support systems and their evaluation. Flexibility, discovery orientation and ease of learning are identified as general characteristics of DSS. Three general capabilities of DSS are identified: data management, model base management, and dialog generation. Further, the authors identify two types of decision support programs: decision aid and decision modeling. The evaluation of DSS occurs in three phases: (1) needs assessment (the who, what, how and when for DSS; (2) matching software to the organizational needs; (3) variables associated with ongoing use of DSS.

The authors present a five step DSS software evaluation process:

1. Identify decision environment.
2. Define users needs and set objectives.
3. Define DSS requirements and constraints.
4. Develop DSS evaluation framework and evaluate.
5. Implementing and reevaluating.

Each step in the process is further defined with methods for quantifying the process.

Badre, Albert (1984). Designing transitionality into the user-computer interface. In Human-Computer Interaction (G. Salvendy editor). Elsevier Science Publishers.

Badre is concerned with the fact that the user of a computer is not a static entity. The novice user becomes experienced with time and practice and makes a transition to the level of an experienced user and possibly to an expert. Menu-prompting interface tends to be very good for the novice user, allowing the user to rely on recognition memory strategies and permitting the use of trial and error for learning the system. The same menu-prompting interface typically presents a problem for the experienced user, as well as for the expert user.

Badre suggests the following requirements for designing for transitionality:

> The system must allow a task to be performed in at least two different modes.
>
> The novice mode of operation should support task accomplishment through recognition memory strategies.
>
> The expert mode of operation should support task accomplishment through a free recall interaction.
>
> The expert's vocabulary and syntax may be learned as a byproduct of practice as a novice (this requires continuity of vocabulary and syntax across modes).
>
> Expert mode should allow: (a) bypassing of novice interactive strategies and (b) introduce new vocabulary and generate new sentences for chunking and summarizing actions.

Badre presents a discussion of the Infoscope interactive environment which was designed to support the transitionality notion. Infoscope supports user interaction in six modes: menu prompting, command name prompting, command statement, command statement abbreviation, definitionals, and command files. Research is being conducted to look at the patterns developed by subjects of various skill levels as they gain experience with the system.

Bandura, Albert (1984). Representing personal determinants in causal structures. Psychological Review, 91, 4.

This commentary is in response to the criticisms raised by Staddon to Bandura's formalization of models of causality. A major criticism raised by Staddon concerned the lack of formalism in Bandura's model.

In his earlier work, Bandura proposed three conceptions of interaction (behavior, cognitive and other personal factors, and environmental factors). In two of these conceptions, behavior is shown as a by-product of the interaction between a person and situation where behavior itself is not an interacting determinant (the influences on behavior flow in one direction only). Bandura terms this idea triadic reciprocal causation.

Staddon, in his model, views behavior as under the control of current external stimuli and past stimulus inputs. Bandura suggests that evidence does not support the notion that cognitive determinants of behavior can be reduced to past stimuli inputs in causal structures. Bandura maintains that when interacting with the environment people bring knowledge and cognitive skills to option generation, rather than simply reacting to external stimuli or being guided simply by past stimuli.

Bandura acknowledges that thought processes are not directly observable. He maintains, however, that they may be known indirectly. Further, he states that the indicants of cognitive processes are separate from the behavior to be explained.

Past experiences do affect people. Bandura maintains that the issue relates to the unidirectionality or bidirectionality of those experiences on human behavior. He suggests that past experience is represented in the manner of constructive processes (filtering, transformed into propositional knowledge, for example).

Card, Stuart K., Thomas P. Moran, and Allen Newell (1983). The Psychology of Human-Computer Interaction. . Lawrence Erlbaum Associates, Inc.

This book addresses the application of knowledge of human cognitive behavior to practical domains in computer science. The authors' premise is that advances in cognitive science and related sciences are sufficient to allow practical applications of the work.

The focus of HCI is on the mechanisms used in the communication dialog between the human and the computer. Further, the dialog is engaged for the purpose of accomplishing a task. One reason for the frequent problems encountered in the design of HCI is that interactions with computers are relatively new. Older models of HCI tended to focus on operators models where a very limited tasks were accomplished with a limited range of operations. HCI involves working with a machine through communication - rather than operating a machine.

The authors present simple models of the human processor. Identifying three subsystems: perceptual, motor, and cognitive. Each subsystem is described by the following parameters: storage capacity of items; decay time of an item; main code type (physical, acoustic, visual, semantic); and cycle time. Human performance is decomposed into five categories: perception; motor skill; learning and retrieval; and complex information-processing. Experimental data is presented for the parameters for each subsystem. However, caveats are offered that some parameter values may be called into question based on new experimental evidence (e.g., the work of Chase and Ericsson demonstrated that practice can significantly expand the size of working memory). The authors note that the simplicity of their model does not reflect all the complexities arising from cognitive psychology (the organization of semantic memory. for example), however, simplicity was the aim of the model.

The authors introduce a model of cognitive skills, termed the GOMS model. This model portrays the user's cognitive structure as containing the following elements:

A set of Goals.
A set of Operators.
A set of Methods for achieving goals.
A set of Selection rules for choosing among competing methods for goals.

GOMS is used to create a user model. Observation of users is the method by which the model is developed for a particular application. The GOMS model can be formulated at different levels of analysis (including the Keystroke-Level Model for evaluating performance of an interface). Various applications were chosen and demonstrated in the book.

Crow, Frank and Charles Csuri (editors) (January 1986). Displays on display. <u>IEEE Computer Graphics and Applications</u>, 6, 1, 6 - 31.

This article is a collection of five papers related to various research and development efforts on displays, each of which is briefly discussed below.

Shannon Moffat and Angela Reilly describe Stanford University's Video Graphics Research Facility. The center piece of the facility is a Bosch FGS4000 video animation computer which the authors note is similar to systems used in the production of animated 3D visuals for television. The facility is permitting animation to be feasible for academic use.

Marie English and Angela Reilly offer a view of the combination of telescience and computer graphics. (telescience is a term coined to describe how scientists would use the space station, in general the term describes the concept of work done terrestrially.) The authors discuss the SUNSTAR (Stanford University Network for Space Telescience Applications Research) project, a joint effort of Stanford and NASA. Number industry affilitates are involved in SUNSTAR as well (including DEC, IBM, Hewlett-Packard, Apple, Sun, Evans and Sutherland, Silicon Graphics, Bosch, ITT, Motorola, Lockheed, TRW, and McDonnell Douglas). The SUNSTAR laboratory serves as a testbed for a large range of research questions inherent in the design and implementation of a space station.

John Staudhammer offers a brief overview of the hardware revolution which proved to be beneficial to computer graphics. The decline in hardware costs has been an obvious benefit. He also notes that while costs have been reduced, cycle times have not.

Avner Ben-Dor and Brian Jones offer a discussion of the technology and architecture to support recent developments in electrostatic plotting technology. The RPM (Raster Processing Machine) controller is described along with a discussion of its performance evaluation.

John Staudhammer and Anoop Khurana describe a scheme for producing display images of atomic models. The process described by the authors features an interrelated set of procedures and hardware techniques which result in the display of a 1000-atom molecular model in under one-tenth of a second. The authors note that using dedicated VLSI processors would allow the time to be reduced. The article offers detail related to the hardware description and computational models used to develop such displays.

Elkerton, Jay and Robert C. Williges (1985). A performance profile methodology for implementing assistance and instruction in computer-based tasks. <u>International Journal of Man-Machine Studies</u>, 23, 135 - 151.

The authors present the notion of developing assistance and instructional systems as an alternative approach for designing human-computer interfaces. This paper identifies a number of issues relevant to the HCI design, specifically representational issues relating to the use, development, and nature of models.

The authors present a performance profile methodology for assistance and instruction in the task domain of file searching. The profile developed is limited in its applicable to tasks. The profile is useful for tasks that can be described by a limited set of functions or commands and tasks that are feature some amount of repetition during skilled performance. These constraint are sufficient to enable a designer to sample performance for the contruction of an expert model.

The authors note the following limitations to their profile methodology. For one, there is no explanatory power in the representation, the expert developed is a "black-box" expert. Second, the model lacks strategic and planning information. An additional limitation is the lack of sequence information which would be useful for diagnosis and remediation. However, the profile of experts developed can be used to diagnosis when novices have gone astray.

England, Nick (January 1986). A graphics system architecture for interactive application-specific display functions. <u>IEEE Computer Graphcis and Applications</u>, 6, 1.

Typically, there have been two approaches to meeting the requirements of interactive computer graphics display requirements. One approach is the development of highly specialized systems designed for a specific application. A second approach is the development of more general systems with a limited set of display functions capable for being used by a wide range of applications. England offers a third alternative, a high performance, general purpose display architecture capable of supporting both specific application and common graphic functions.

England's graphic subsystem is available in commercial form (the Ikonas Raster Display System - available as the Adage 3000). The key features of this system include:

single common bus structure
modular architecture
general purpose 32-bit graphics processor
large image memory
flexible display controller.

England discusses each feature in some detail. In addition, he offers a description of the various applications that utility this system, which include molecular modeling, medical imaging, computer-aided design, seismic data display, as well as basic graphic functions.

Gaines, Brian R. and Mildred L. G. Shaw (1986). From timesharing to the sixth generation: the development of human-computer interaction. Part 1. International Journal of Man-Machine Studies, 24.

This article is the first of a two-part series concerning the development of human-computer interaction. The specific concern of this paper was the state of the art in HCI R&D, providing an analysis of the past trends in HCI and using these trends to propose future directions.

The authors raise an important issue concerning empirical studies of HCI, namely that empirical human factor studies provide insufficient guidance for HCI designers. The reason for the mismatch is that the empirical studies tend to focus on the problems and issues of the past, the results of which may not have application in the present let alone the future. The authors indicate that the only solution to the dilemma lies in the development of foundational models of HCI that may be projected into novel situations.

The remainder of the article traces the development of HCI, demonstrating how the current inadequacies came about and why they are expected to change.

Typically, HCI is viewed as a distinct discipline, the authors, however, approached the development of HCI within the overall context of system development. From this perspective, HCI may be viewed as developing in accordance with an infrastructure featuring: breakthrough advances; replication period (mimicing the breakthrough); empirical period (formulating design rules); theoretical period (formulating and testing theories); automation period (theories predict experience and generate rules); maturity (assimilation and routine use). From this structure, the current fifth generation is characterized as follows:

> Electronic device technology and virtual machine architecture have achieved maturity.
>
> Problem-orientated languages are in the automation period.
>
> Human-computer interaction is in the theoretical period.
>
> Knowledge-based systems are in the empirical period.
>
> Inductive inference systems are in the replication period.
>
> Autonomous activity systems are in the breakthrough stage.

In the sixth generation, HCI should evolve into the automation period, theoretical foundations should be established for knowledge engineering, and machine learning should be sufficiently evolved to the point that pragmatic rules are established.

The authors point out the pivotal position of HCI within their proposed infrastructure. HCI is between two forms of computing, one based on algorithms and another based on the encoding of knowledge, learning, and expressing goals. This pivotal position is further demonstrated by the objectives of HCI studies. On one hand, the studies focus on interfacing the human effectively to the system and on the other hand, the focus may be on achieving a symbiotic relationship between the human and a system which has the knowledge-processing capability of a human. The first type of studies tend to look at the ergonomics of the interface and interactions. The second type of study looks at a total knowledge system, the marriage of artificial intelligence and cognitive science. The relationship of these two types of studies is a strong one, the first type of studies are a prerequisite to achieving the second types. Development of effective human-computer communication must preceed the accomplishment of symbiosis between the human and computer.

Gaines, Brian R. and Mildred L. G. Shaw (1986). Foundations of dialog engineering: the development of human-computer interaction. Part II. International Journal of Man-Machine Studies, 24.

The second of the two part series of articles is concerned with the development of styles of dialog through to the existing state of the art and projecting into the future.

Major dialog styles may be distinguished into three categories:

Formal - the structure of formal dialog stems directly from the underlying machine.
Natural language - the structure of natural language dialog stems from human linguistic basis of knowledge representation, communication, and inference.
Graphic - the structure of graphic dialog stems from a physical world.

Formal dialog is a direct representation of computer activities and data structures, such as prompt-response dialogs. Natural language dialog is, by contrast, a simulation and is remote from the actual system operation. Graphic dialog is similar to natural language in that it is a simulation, however, it is an object simulation. The attractiveness of both natural language and graphic dialogs are the existing skills of users that are brought into play in such interactions. Within the context of fifth generation systems, the classification of dialogs breaks down as the styles become integrated (for example, the MAC with its graphic desk top metaphor combined with the formal dialog of the pull down menus).

From the authors' perspective, the fifth generation HCI must move from the current reactive paradigm to a prescriptive one capable of moulding technology. To achieve fifth generation objectives, a set of principles that systematically generate rules for dialog engineering is needed. Such principles would be grounded in areas of system theory, computer science, and cognitive psychology. Application areas of such principles would encompass all dialog styles and technology with both current and future utility. Further, the principles need to become embedded in standard procedures. The authors identify numerous developments that satisfy some objectives, some of the principles include:

Principles for dialog design (Thimbleby, 1980).
Virtuality principle (Nelson, 1980).
Direct manipulation (Shneiderman, 1983).
Usability (Shackel, 1983).
Levels of HCI (Norman, 1984).

The authors present a systematic analysis of HCI in an attempt to further identify theoretic principles applicable to HCI. Pask's (1975) conversation theory is identified as directly relevant to the development of a theory of system-system interaction. This theory provides systematic criteria for communication to occur. Conversation theory has been applied by various researchers in a range of domains, including: Coombs and Alty (1984) - expert system design; Shaw (1984) - knowledge engineering; and Begg (1984) - HCI in CAD systems. Between two systems there are three levels of complexity:

Level 1 - the message itself.
Level 2 - the system affected by that message.
Level 3 - the identification of the system.

When placed in a context, this simple list takes on considerable complexity. A simple message, for example, may cause massive effects on the receiving system (e.g., President issuing a nuclear strike message). The message sender may or may not be concerned about the system affected by the message. In this way, the relationship between levels 1 and 2 need not exist. On the other hand, if the message sender is concerned about the message effect on the receiver, a model of the receiver is needed. This in turn may pose varying degrees of difficulty based on the dissimilarity between the sender and receiver, if both systems are alike, complexity in level 3 would not be invoked. This discussion serves to stress one point, complexity in the HCI is relative.

In conclusion the paper identifies strategies for providing HCI design principles (the first four from Norman, 1983):

1. Make system designers more aware of the significance of HCI.

2. Provide methods and guidelines for HCI design.

3. Provide software tools for HCI design.

4. Separate HCI design from other design tasks.

5. Provide an overall framework for HCI in systems involving multiple people in various roles and multiple computer subsystems at different levels. From the overall framework derive the other four strategies.

Gomez, Louis M. and Susan T. Dumais (1986). Putting cognitive psychology to work: examples from computer system design. In Terry J. Knapp and Lynn C. Robertson (eds.) Approaches to Cognition: Contrasts and Controversies, Lawrence Erlbaum Associates, Hillsdale: NJ.

Gomez and Dumais offer a review of how cognitive psychology has met the challenge of aiding the design of computer systems. While the authors note the rather tight coupling between the interests of cognitive psychologists and system designers, they note the difficulties of identifying clear examples of direct application from the psychologists to the designers.

The authors note possible reasons for the lack of examples of cognitive research findings to computer design. For one, the apparent relationship between cognitive research and computer design is fairly recent and technology or knowledge transfer may be characterized as neither rapid nor effective. Secondly, the authors note the likely disconnect between the answers provided by cognitive psychology and the questions posed by designers. This is a generalized problem of the effective communication of research across numerous disciplines.

The authors offer a strategy for bringing cognitive psychology into the design domain. They suggest that "basic" researchers focus more on complex domains and real-world problems. The common ground on which practice and theory tend to meet is in the choice of a relevant domain.

The authors stress the need for developing methods and tools for numerous areas in human-computer interaction, such as theory-driven instruction and training, task analysis, knowledge elicitation, and procedures for interacting with systems. They suggest that the knowledge of task and structure or process be used to predict the success of various designs.

Hahn, Major Daniel A. (November 1985). Leadership: the heart of C2. Military Review, LXV, 11.

Major Hahn points to the need for an expanded definition of command to bring in the notion of leadership. He maintains that the relationship between command and control is predicated on leadership philosophy.

The author characterizes control as a continuum. At one extreme control is characterized by regimentation, regulation, and norms. At the other extreme, control is totally self-regulating.

Major Hahn maintains that manpower is at the heart of effective command and control. He believes that individuals working to their maximum capabilities, properly educated, and strongly committed to their organization will lead to effective command and control.

Hall, Roger I. (1984). The natural logic management policy making: its implications for the survival of an organization. Management Science, 30, 8, 905 - 927.

Hall uses a process-theoretic approach to propose a process model of the natural logic of organizational policy making. He uses this model to explain organizational policy adoption. Further, he suggests the use of systematic procedures in management science.

A five part framework is presented to describe the evolution of an organization:

1. Enactment processes. Initially an organization may select actions based on an elementary map of causality (e.g., budget planning).

2. Selection processes. From an organization's action on the environment, information is registered by the selection processes.

3. Retention processes. This is a memory function where information becomes part of the organization's knowledge base.

4. Driving forces. Hall postulates various driving forces including: increase in status and power; reduction or control of uncertainty.

5. Retained set. Hall proposes that the retained set is derived from past interactions in response to environmental changes and this set includes: standard procedures, current strategy, the present architecture which provides the social setting through which future resource demands are made, and the current structure of causality maps.

Hall suggests the notion of an organization's cause map. A cause map describes the causal paths from policy to goal achievement. Three types of relationships are contained in cause maps: logical or accounting relations, observable relations from feedback relations based on belief and environmental conditioning (but where insufficient evidence is available to confirm relations).

Hall applies the proposed model to explain the policies of a publishing firm. As part of his analysis, Hall identified numerous pathologies in organizational policy making. These pathologies included: lack of proper management supervision to adapt and contain interorganizational power-politicking; lack of reliable procedures for constructing and verifying causality maps; lack of guiding principles for decision making where the natural policy making process is less than reliable; excessive standard procedures leading to a reduction in the variety of controls and a reduction in the ability to react to circumstances.

Halpin, Stanley M. and Gary W. McCullough (1986). User-computer interaction. In Joseph Zeidner (ed.) Human Productivity Enhancement Training and Human Factors in Systems Design Volume 1. Praeger: New York.

This chapter provides an overview of the historical development of research in the area of user-computer interaction. The authors note the inadequacy of past and present theoretical foundations for the future objective of establishing a symbiotic relationship between users and computers.

The major recommendation in this work is to view the interaction between users and computers as a dialog. The dialog characterization has implications for the knowledge requirements of the computer. The authors identify the knowledge requirements as:

Knowledge of the domain
Knowledge of the computer version of the domain
Knowledge of workbase version of the domain
Knowledge of the problem
Knowledge of system operations
Knowledge of physical interface
Knowledge of interface dialog
Knowledge of natural language
Knowledge of other machines and procedures.

The authors stress that this knowledge characterization goes beyond the perceptual characteristics of the interface as well as beyond the notion of the user as information processing mechanisms. To achieve interaction as a dialog, an additional step is needed; namely, bringing into the picture research related to effective communication to achieve a dialog rather than a monolog. The authors propose a software architecture to support the interactive dialog between users and computers.

Halpin, Stanley M. and Franklin L. Moses (June 1986). Improving human performance through the application of intelligent systems. Signal.

This article provides an overview of the research efforts proposed and in-place for the ARI Field Unit, Leavenworth, KS, and the Battlefield Information Systems Technical Area. One brief section of the paper addressed the notion of the intelligent interface which is summarized below.

The intelligent interface takes the form of an intelligent shell serving as interpreter and assistant for both the user and the software system. The authors identify a number of features critical to achieving this dual function:

The system must understand the user.
The system must know the user's goals and objectives.
The system must know the tasks to be performed (user tasks).
The system must know the criteria that signifies to the user that the task has been accomplished.

The system must understand the world as the user understands the world.
The system must understand how to access and use the system's tools to accomplish the tasks required to support the user.
The system must have ways of verifying and modifying as needed it's model of the world in comparison to the user's model.
The system must have mechanisms for eliciting feedback for such comparisons (as well as knowing when to make comparisons).

The remainder of the article identifies numerous ARI projects (on-going and no longer going) that will be working on different aspects of the intelligent interface problem.

Hayes, Philip J. and D. Raj Reddy (1983). Steps toward graceful interaction in spoken and written man-machine communication. International Journal of Man-Machine Studies, 19, 231 - 284.

Graceful interaction refers to a manner of system to man communication that deals appropriately with anything a user says, rather than a restricted vocabulary. Components of graceful interaction are identfied as independent skills, including:

Robust communication.
Flexible parsing.
Domain knowledge.
Explanation facility.
Focus mechanisms.
Identification from descriptions.
Generation of descriptions.

The authors maintain that graceful interaction is fundamental to the establishment of a dialog between users and computers.

Hayes, Philip J. and Pedro A. Szekely (1983). Graceful interaction through the COUSIN command interface. International Journal of Man-Machine Studies, 19, 285 - 306.

The COUSIN (COoperative USer INterface) project is a research effort to provide a friendly and supportive interface to users. COUSIN features a form-based model of communication, on-line help, and error correction. COUSIN implementations include COUSIN-Unix (operating system command interface) and COUSIN-SPICE (command interface for the Perq, a PC with bit-map and pointing device).

The basic mode of communication between the user and an application program is indirect via the reading and updating of fields in forms specific to the application. COUSIN controls access to the forms. Fields within each form have both type restrictions and default values. On-line help focuses is informing user of available fields, types, and defaults.

The major user objection to COUSIN concerns the line-oriented form editor. Since this provided the user with only a single field at a time, users experienced difficulties in retaining context. A screen-oriented approach was offered as a solution for future implementation.

Jones, Jack William and Raymond McLeod, Jr. (1986). The structure of executive information systems: an exploratory analysis. Decision Sciences, 17, 220 - 249.

Jones and McLeod conducted research into the question: "where and how do senior executives get their decision-making information?" The answers they found may have a significant impact on MIS and DSS. The authors findings are summarized below.

Executives manage information volume and value by controlling selection of sources and media (sources - internal to organization, external to organization; media - written, verbal).

A high volume of low value information is tolerated to avoid overlooking important information.

The sources of media used in decision making depends upon the decision makers' role (identified in this context as improving projects, allocating resources, handling disturbances, and negotiating).

Computer-generated information plays a minor role in a senior executive's information-gathering network. (this does not mean the information is not valued, it means that senior executives do not obtain it directly).

Research suggests the possibility of an "information chain" in the decision-making process which requires further study.

Informal systems may play a more important role than formal systems in providing senior executives with information.

Internal information is highly valued by senior executives with scheduled meetings being the preferred medium for all decision roles, though preferred sources and media vary based on role.

The authors suggest based on their research that MIS and DSS concepts should be broadened to match the actual information networks used in practice. Further, they suggest that automation aid the flow of informal information.

Katz, Joseph L. (Fall 1985). Artificial Intelligence at MITRE. The AI Magazine, 228-232.

This article discusses the research efforts in progress at MITRE in the area of artificial intelligence. MITRE's research in AI originally operated through internally funded research and development. In 1978, a major effort was undertaken by MITRE to investigate and demonstrate the applicability of promising AI technology. This project is known as KNOBS (for knowledge-based system) and is currently supported by Rome Air Development Center.

Katz highlights a number of MITRE AI programs by the center conducting the work. Principle investigators of the various programs are identified. The programs discussed below focus on the topic of planning and do not include the other MITRE programs discussed in the article.

MITRE, Bedford (Stuart Goldkind, PI). The current focus of planning research centers on an implementation called KNOBS Replanning System (KRS). The system creates plans for resource allocation, revises plans when expectation failures occur, and minimizes the impact of revisions on the remainder of the plan. The replanning aspect deals with earlier plans as objects to reason about and change. Research focuses on mechanisms for meta-level reasoning and planning. Additionally, research is exploring ways in which the system can use global strategies in the planning process, as well as strategy understanding and explanation. Dynamic planning is also a research issue.

MITRE, Bedford (Joseph Katz, PI). A prototype expert system is being developed for NASA, termed EMPRESS (Expert Mission Planning and REplanning Scheduling System). EMPRESS will be used for scheduling activities for cargo payloads for space shuttle missions. The support EMPRESS will provide is of two types: schedule checking to identify conflicts and track resource use; and automatic replanning - proposing partial solutions to scheduling problems.

MITRE, Bedford (Judah Mogilensky, PI). KNEECAP is a demonstration system under development to apply expert system technology to Space Shuttle crew activity planning. KNEECAP will perform constraint checking on user inputs, identify failures and offer explanations, and an autoplan capability will be available to the user.

MITRE, Washington (John Benoit, PI). OPLANNER demonstrates aiding for the joint logistics planning domain. Using estimates of resource requirements OPLANNER develops partial plans. OPLANNER is characterized as a hierarchical and rule-based system. Domain specific information is represented in a large semantic network. OPLANNER has been implemented on a VAX 780 in Franz Lisp (currently being converted to the Symbolics).

Lebowitz, Michael (1986). Integrated learning: controlling explanation. Cognitive Science, 10, 219-40.

Inductive techniques of machine learning tend to take one of two approaches: similarity-based learning (SBL) or explanation-based learning (EBL). SBL programs learn by the comparison of instances, using little high level domain knowledge. By contrast, EBL uses extensive domain knowledge applied to a single instance, building an explanation of various components and their relationship as well as applying generaliziations from the instance for understanding further instances.

EBL is advantageous to cases where substantial domain knowledge is available. The major disadvantage noted by Lebowitz is that EBL is less likely to generalize coincidental type information that arises in the use of SBL. The further disadvantage of EBL is that it may be computationally expensive where extensive amount of domain information are required.

Lebowitz suggests the need to bridge the gap between SBL and EBL, primarily through the integration of the two. The model proposed learns by noticing similarities, either when specific rules are unavailable or when the payoff for EBL use is not likely to be high.

Lebowitz developed a program called UNIMEM which takes facts about objects from a domain and organizes them into a generalized memory with specific instances stored in terms of generalized concepts. UNIMEM learns by observation. UNIMEM is not provided with a set of concepts to learn nor with prepared sequences to learn. Rather, UNIMEM creates a hierarchy of concepts by noting similar instances and assuming that the similiarities represent domain regularities.

In an attempt to explain causality, a determination must be made of why a set of conditions led to an observed behavior. Lebowitz notes the difficulty of this in a domain with limited knowledge. His proposed solution is to introduce predictability. Predictive features are those features of a generalization that are most nearly unique to a particular generalization. Non-predictive features occur across various generalizations and may be associated with different feature combinations. The use of predictability is to provide a focus for the use of explanatory rules.

In an example UNIMEM was used in an attempt to explain congressional voting records. A number of things were demonstrated in the example. For one, the initial rules developed ("by hand") could be evaluated. While this was not a focal point of the example, the potential for the evaluation of the reliability of the rules has value for domains in which tentative heuristics represent the best estimate that may be made. Second, a brute force approach to explanation was not needed. This can represent a considerable computation savings for developing explanations in domains in which general and contradictory rules are the only ones available.

Determining when to generalize is addressed by Lebowitz in terms of interest. He defines interest as a heuristic measure of what is likely to help in learning. The use of interest in UNIMEM occurs in two ways. For one, focusing on generalizations rather than individual instances is an application of interest. Deciding how to focus the EBL process is a second application of interest. In this paper, components of the interest measure were not discussed. Rather, the focus concerned a rationale for the plausibility of computing an interest measure. Interest heuristics developed by Lebowitz in earlier work focus on relevance and novelty. In general, interest focuses on generalizations that describe a number of instances (rather than a few), possibly generalizations that involve an unusual set of instances, and generalizations that organize other generalizations. Interest rules would be used to focus the explanation process on features of a generalization. In this way, novel, but not too novel, features would be a focus for explanation.

In summary, Lebowitz proposes a three step process:

Apply EBL to generalizations derived from similarities noted (rather than to individual instances).

Use interest to determine when to learn.

Use predictability to help control the explanation process.

Lewell, John (1985). Computer graphics a survey of current techniques and applications. Van Nostrand Reinhold, NY: NY.

Lewell's book offers a survey of computer graphic technology. This book offers an overview of the history of computer graphics, from plotter drawn pictures, through computer enhanced pictures from Voyager 2, and flight simulation imagery, as well as computer art.

Discussions include how computer graphics are created and used, looking at hardware and techniques involved. A range of graphic systems are discussed including vector displays, raster displays, and variations of vector and raster (direct-view storage tube, plasma panel displays). Various application areas of computer graphics are contained in this book, including graphic design, illustration, television and film, the sciences, and business and industry.

Leyton, Michael (1986). Principles of information structure common to six levels of the human cognitive system. Information Sciences, 38, 1 - 120.

Leyton proposes a theory of information structure in human cognitive systems and offers evidence of the theory in six levels of cognition. The levels of the cognitive system identified include:

1. Simple cell function.
2. Psychological scales.
3. Perceptual organization.
4. Categorization.
5. Grammatical structure.
6. Planning.

Leyton maintains that a general theory of information structure may be constructed into which the individual theories (e.g., theories of grammatical structure, categorization, etc.) are demonstrated to pertain to the more general phenomena of information itself.

Leyton presents a number of general informational principles and their validity to the six cognitive levels (details related to the theoretical structure of the principles is not presented in this article). Leyton argues a counter position to current notions of the processing computer analog theory of cognition. Leyton maintains that "any cognitive representation is structured as a computer." Further, he proposes a content machine analogy which states "the cognition of, or the extraction of information from, a stimulus set, is the description of the latter as a machine."

Extensive empirical evidence is presented for each of the six cognitive levels and their interpretation according to the postulates and corollaries of Leyton's informational principles.

Malone, Thomas W., Kenneth R. Grant, and Franklyn A. Turbak (April 1986). The information lens: an intelligent system for information sharing in organizations. _CHI'86 Proceedings_.

The Information Lens is a system designed to help people share information communicated via electronic messaging systems. The use of this intelligent system is to attempt to address two problems: the overwhelming of users with too many irrelevant messages; and ignorance of users to information that is relevant. The Information Lens aids users to filter, sort, and prioritize messages addressed to them as well as aids users in finding useful messages that are not addressed to them.

The Information Lens was developed for the electronic mail environment. The system was built on top of the existing electronic mail system. Four capabilities were provided by the Information Lens:

> Structured message templates are used to help users compose and read messages.
>
> Users can specify rules to filter and classify the messages they receive.
>
> Messages may be tagged by the sender to indicate the message is readable by those other than the specified addressees, such messages are sent to a specific mailbox called LENS.
>
> Message receivers can specify rules to locate and display messages contained in the LENS mailbox.

The Information Lens system has a set of semi-structured messages. For each message type, the template contains a number of fields or slots which have associated with them several properties including default values and likely alternatives. Users construct messages using these templates. The user selects a message field and may query for default and alternative values for direct insertion. In this manner, the construction of messages may be quite rapid and require few keystrokes. Alternatively, a user may chose a generalized message type and use the text editor to write the message.

Message templates are arranged in a network. In this way, templates inherit field names and values from the parent template. Subtype messages may in turn add new fields or override the values inherited from the parent. The inheritance network may be browsed by users when creating new message types.

As noted earlier, Lens allows users to develop rules for locating, filtering, and sorting messages. The rules used in Lens are IF THEN statements consisting of a test and an action. Rules are constructed by users in much the same way as messages are constructed (termed by the authors a template-based graphical rule construction). Users construct rules related to the field specifications in the message (complexity is permitted within a single specification by stringing boolean combinations).

A major advantage of Lens as noted by the authors is the easy of adding additional capabilities. Further extensions may include a topic network and a knowledge-based server.The information filtering approach used in this effort was based on user elicited protocols for filtering collected from the users of this electronic mail system. The authors note that this was one approach (cognitive filtering as they termed it). Two other approaches are possible that like Lens do not require the representation of the content of the messages. These approaches include: social filtering (supports personal and organizational relationships among community members) and economic filtering (quanity and quality of information flow is controlled by various positive and negative incentives).

This paper did not provide parameters concerning either the number of users involved in the electronic mail environment nor the quantity of message traffic handled. However, the Information Lens system did exploit various AI concepts such as the use of frames, inheritance, and production rules.

McCracken, Donald L. and Robert M. Akscyn (1984). Experience with the ZOG human-computer interface system. International Journal of Man-Machine Studies, 21.

ZOG is a general-purpose interface system designed by CMU. Originally designed for internal CMU use, ZOG has been applied to the U.S.S. Carl Vinson as a computer-assisted management system.

ZOG is based on the concept of menu-selection with a vast database of menus. The basic unit of representation in the ZOG system is a frame. Three types of interaction are used in ZOG: navigation, invoking programs, and editing. Navigation is the default interaction. Selections are made via keyboard entry or mouse, with the system responding by displaying the next frame.

The design of the ZOG system was based on a set of principles, including the following:

> A single environment with a single human-computer interface would help to avoid the idiosyncracies of multiple interfaces when switching between programs.
>
> An interface should be a flexible and efficient tool with the workings of the interface under user control. Active intelligence should be at the subsystem level, not at the interface level.
>
> Users should be able to manipulate the data stored in the computer directly.
>
> Semi-automatic operation should allow the user general-purpose facilities for performing operations for which the system has not been specifically adapted.

The system should be easy to learn to use.

The environment should permit safe exploration and learning by doing.

The database is a major component of the ZOG system. The design philosophy of the database calls for:

An architecture which accomodates hundreds of thousands of frames. However, the size should not adversely affect system responsiveness. The system should support simultaneous users.

The database should represent both textual and graphic information.

The database should have a network structure to support the linking of data items.

In addition to network structures, tree structures should be represented. In this way data items may be both cross-referenced to frames as well as data items linked to tree structure levels.

The database should contain nothing but menus. Menus contain pure information as well as links to other menus.

Four levels of organization should be contained in the database. At the lowest level there is a substructure on single data items (e.g., value of a data item); next are the single data items (e.g., phrase, paragraph, picture); next are collections of items (e.g., menu - what is displayed to the user); at the highest level, there are functional groupings of menus (such groupings may have a common structure derived from a schema menu copied when the frame is created).

The default view of the database should be a breadth-first organization of the tree structures.

The user-system interactions of ZOG are highly prescribed, for example:

With few exceptions, user interaction with the system is via menu selection from the currently displayed menus.

System response is extremely rapid (well under one second when the selection results in the display of another frame).

Browsing through a network of linked displays is the default system mode.

User selections should have actions associated with them.

The only functions available to the user are those currently visible functions (the user relies on recognition memory, only).

Common commands of general utility are available on every frame. A general editor is available on every frame.

Frames should fit the size of the available display space, scrolling within a single frame should not be permitted. The rationale for this stems from the design concerns for simplicity and the need for visibility of available functions. In the event that additional space is required for a frame, linearly sequenced frames or extra levels of hierarchy are suggested.

Additional principles were identified to address the way in which the system is extended to add new functions. Adding new applications to the system follows the principles:

The data structures of the new application are mapped into frame formats and interconnected with structures in the database.

New programs are written in a manner such to imbed them within the system (in this way they are invoked without leaving the system - termed agents).

Environment frames contain slots for input and output parameters and values for agents (this frame has a slot editor associated with it for checking and allowing value selection from menus).

Input data is obtained by direct access of frames by agents.

CMU has applied ZOG to numerous application areas, including database systems, management information systems,instruction/training, document management, software management, and electronic communication. For the U.S.S. Carl Vinson application, ZOG interfaced with an expert AI system (AirPlan). ZOG serves as an input and output interface to AirPlan. A slot editor provides the means for updating AirPlan, while the output of AirPlan is sent to ZOG for frame display and access may users in the network. An added feature of ZOG highlights changed information in the frame to reduce the cognitive load on users.

While ZOG is not an intelligent interface (by CMU description), it is interactive while not adaptative. The intelligence of ZOG comes from its network structure. Experience with ZOG has demonstrated to CMU a number of important results of their design philosophy. While ZOG is learned very quickly (basic navigation skills are acquired in under half an hour), ZOG is not outgrown with experience. CMU researchers account for this findings by several aspects of ZOG: the high response rate of ZOG makes it competitive with a command language interface; expert ZOG user use a broader range of functions than novices and appreciate the reduction of memory burden inherent with ZOG; experts appreciate the efficiency of ZOG in developing additional structures. While the authors discuss a number of strengths of ZOG, one of the more relevant is the fact that ZOG exploits schemas for building databases. A chunk of ZOG data (e.g., a frame or a tree of frames) is a schema containing variable parts. The schema may be instantiated by simply copying it and providing values for the variables parts. In a similar manner, whole trees of frames may be instantiated.

ZOG presents an interesting (and different) view of adaptivity (which is frequently viewed as synonymous with intelligence). The adaptivity of ZOG is within the system not the interface. In this way, the amount of adaptivity is prescribed. This appears to have been a deliberate choice in order to provide a fast responsing system. ZOG does not support a large range of interaction modes (fast database query language is not supported). Users are required to operate exclusively in a "breadth-first" view of information. ZOG cannot handle certain types of task domains (such as complex, unordered task environments.

In spite of the limitations of ZOG, the perspectives of the researchers in the design of the interface are not without merit. It is obvious that ZOG development was guided by specific objectives, such as system response speed, which may have accounted for a 180 degree turn from the more traditional approaches to interface design.

McMahon, Major (P) Timothy L. (November 1985). The key to success: developing a C2 philosophy. Military Review, LXV, 11.

McMahon maintains that the current notion of C2 will not support the winning of future battles. The authors suggests that current definitions are mechanistic in nature and fail to account for the humanistic elements.

He suggests that command is a dynamic process of identifying intent, putting intent into operation, and infusing that intent into subordinates. Effective leadership leads to effective command.

Command is supported by control which is characterized by communication. The achievement of control is accomplished through management procedures. Control is a necessary means of command.

From these distinctions, the author maintains the proper perspective of command and control is achieved. The commander must balance the two. He notes that the balance between command and control is not a fixed one, the primary conditions which impacts the balance is uncertainty. The author suggests that as uncertainty increases, control increases, in this manner constraining action. Similarly, as certainty increases, command increases, and greater freedom of action is acceptable.

McNamara, Timothy P. (1986). Mental representations of spatial relations. Cognitive Psychology, 18, 87-121.

McNamara reports the results of an investigation testing three classes of theory about the mental representation of spatial relations. The three theorical perspectives tested include:

1. Nonhierarchical theories which maintain that spatial relations among objects are represented in a network or imagelike, analog manner.

2. Strongly hierarchical theories which maintain that environmental regions are stored in different branches of a graph-theoretic tree where only those spatial relations needed for accurate representation are encoded.

3. Partially hierarchical theories which also maintain that environmental regions are stored in different branches of a graph-theoretic tree but predicts redundancy in the representation (i.e., spatial relations may be encoded between locations in different regions of an environment).

The subtlies of specific theories within the general classes identified above are discussed. In addition, McNamara offers a review of various methodologies which have been used to study spatial knowledge.

The experimental results of this study favor the partially hierarchical theories of the mental representation of spatial relations. McNamara offers further validation of these findings by developing a parametric description of a partially hierarchical theory and predicting performance in a recognition task. A computer simulation was used to verify the experimental findings of the study. Potential criticisms of the study are addressed and supporting evidence provided.

Miller, Dwight P. (1981). The depth/breadth tradeoff in hierarchical computer menus. Proceedings of the Human Factor Society - 25th Annual Meeting, 296 - 300.

This article reports the efforts of an experimental study of menu configurations. In this study, goal acquisition performance was studied to assess the impact of various menu configurations. Subject performance was measured in terms of variability, learning, error rate, and acquisition time.

The study concluded that a menu hierarchy of two levels with eight choices per level produced the least variability among subjects, was easiest to learn, produced the fewest errors, and showed the fastest acquisition time. The author recommends that when system requirements do not permit two levels with eight choices per level that the tradeoff should expand breadth rather than depth.

Monarch, Ira and Jamie Carbonnel (Spring 1987). CoalSORT: a knowledge-based interface. IEEE, 39 - 53.

This article discusses a prototype which has been developed to interact in an intelligent manner with bibliographic databases. The intelligence contained in the prototype is a frame-based semantic representation of an expert's domain knowledge. The interface to users features communication through menus, windows for displaying and tracking information, and the selection of search terms.

The authors maintain that the development of an intelligent interface for information retrieval systems is feasible. Key to the success of an interface is the effective communication with each user to insure consistent use among various users. The authors believe that a frame-based semantic and knowledge network provides the basis for effective communication. The question is one of presenting the conceptual system to the users.

The authors suggest the need to integrate natural language with graphically oriented knowledge-based systems. In this way, browsing may proceed by graphically presenting knowledge bases, enabling users to become familiar with contents. Once familiarity is established, natural language interaction may be used to allow detailed and open-ended system querying.

A number of interfaces, such as ZOG and Rabbit, are discussed as information retrieval interfaces. The CoalSORT and its operation are discussed in detail along with the proposed future directions for the research.

Nisbett, Richard E. and Timothy DeCamp Wilson (1977). Telling more than we can know: verbal reports on mental processes. <u>Psychological Review</u>, 84, 3, 231 - 259.

This is the often cited article which is highly critical of researchers ability to tap into higher order cognitive processes. The authors present evidence on the general inaccuracy of verbal reports on higher level processes and possible reasons to account for the inaccuracies. Three major conclusions are offered:

> The general inaccuracy of verbal reports on higher order processes indicates that correct and reliable reports cannot be produced.
>
> Verbal reports are not tapping into higher order processes, rather the reports are based on causal connections between stimulus and response.
>
> Periodically correct reports about higher order processes are due to the incidental correct use of a priori causal theories, not direct access the higher order processes.

Norman, Donald A. (1983). Some observations on mental models. In <u>Mental Models</u> (Eds. Dedre Gentner and Albert L. Stevens. Lawrence Erlbaum Associates, Hillsdale: NJ.

Norman asserts that numerous mental models are of concern to system development. He suggests the following be considered: target system (this refers to the system the user is learning or using), conceptual model of that target system, the user's mental model of the target system, and the scientist's conceptualization of that mental model.

Norman offers the following observations on mental models:

> Mental models are incomplete.
>
> Human abilities to "run" their mental models are severely limited.
>
> Mental models are unstable (details are forgotten especially through lack of use).

Mental models do not have firm boundaries (similarities in operation or with other devices can be a source of confusion).

Mental models are not scientific (superstitious behavior tends to be maintained).

Mental models are parsiminious (trade-offs are willing made to expend extra physical effort to reduce mental complexity).

These observations are expanded upon through experimental evidence obtained in studies of human error and human-machine interaction.

Norman, D. A. and S. W. Draper (Editors). User centered system design new perspectives on human-computer interaction. Lawrence Erlbaum Associates, Inc. 1986.

This book contains a rather high level treatment of issues and concerns in the human-computer interaction field. The book contains a large number of design principles. These principles, however, are discussed at a very high level, rather than a more practical level. Selected chapters are reviewed below.

Cognitive Engineering by Donald A. Norman

Norman characterizes the interface as the meeting place of the programmer and the user. The programmer's conceptual model of the system is characterized as the design model. This conceptual model is typically based on the user's task, requirements, and capabilities, as well as an understanding of user background, automation skills, and possibly concerns stemming from human limitations (e.g., short term memory). In the process of using a system, the user develops a conceptual model reflecting his understanding of the system operation, Norman terms this the user's model. This model is formed and refined through contact with what Norman defines as the system image, which is the interface. From this perspective, the design concern is the development of an appropriate system image.

Knowledge-Based Interface Design by William Mark

In his chapter, Mark expands upon Norman's notion of system image. Mark's focus is upon the translation of programmer's conceptual models of system operation into a model of the actions at the interface. Mark proposes the system image model which is a conceptual model of the actions available at the interface. He presents a methodology based on his research for the Consul system (used for interface design for office automation).

From Cognitive to Social Ergonomics and Beyond by John Seely Brown

Brown notes the importance of informal communication networks which may form a "community knowledge base" about system use. Such channels may serve to propagate misunderstandings (as illustrated by his example related to the use of Interlisp-D), however, the channels could be a viable source for facilitating understanding. He notes that the basis of both the understanding and misunderstanding is the human need to explain and make sense out of experiences with systems.

Brown is at odds with proponents of designing systems whose goal is to be idiot proof. He contends that it is not possible to anticipate and "design away" all potential problems and misunderstandings between people and the machines they use. From this perspective, the emphasis should be on designing for the management of trouble (of both a communicative and operative nature).Thus, the interface design focus is to aid the human in the recognition and repair of misunderstandings.

While reducing the opacity of system operation is a necessary design concern, simply showing how the system functions is not enough. Brown maintains that the system representation must facilitate "the user's ability to "grow" a productive mental model of relevant aspects of the system." Animating internal processes and procedural structures may be useful in helping the user to "see" the system operation.

Learning through experimentation and discovery is typically useful. However, Brown notes that for such an approach to be viable for system users, experimentation cannot lead to disasters. Users must be able to recover, such as through a complete undo capability.

Brown raised two additional points related to aiding users in understanding systems. For one, he sees a need for encouraging an awareness of the importance of the embedding social infrastructure for the effective use of systems. Secondly, Brown notes the need for developing new training strategies to teach relevant new skills. In particular he notes the need for training strategies for continued learning rather than teaching rote procedures.

Brown presents data from studies of an instructional system designed to aid casual users in carrying out complex procedures with a reprographics system. An interaction framework was developed that differentiated actions of both the user and machine and whether or not the actions were accessible to the other. From the analysis it was shown that much of the human behavior was unavailable to the machine. When the communication bandwidth is small, the system could not provide the user with the support necessary to eliminate difficulties. Brown notes the lack of resemblance between a description of an interaction and the way in which an interaction may actually occur. Detecting misunderstandings requires that both the human and system infer the underlying intentions of each other and the assumptions each uses to make the inferences.

Osberg, Timothy M. and J. Sidney Shrauger (April 1986). Retrospective versus prospective causal judgments of self and others' behavior. <u>The Journal of Social Psychology</u>, 126 (2), 169 - 178.

Osberg and Shrauger report an experimental study designed to: examine the influence of time perspective on causal attributions; and determine whether the actor-observer and self-serving bias findings extend to situations in which judgments about factors affecting future behavior are made.

The experimental findings of these researchers suggest differences between the way individuals account for past behavior and the judgments made about potential factors likely to affect future behavior. Stronger attributions to dispositional factors were made by subjects when considering future behavior as opposed to past behavior. The authors suggest that this may be explained by the uncertainty associated with future events and that the uncertainty may be reduced by predicting that occurrences will be dispositionally caused.

Additionally, the researchers conclude from their evidence that actor-observer differences in attribution were demonstrated for future and past behavior. The authors note that the time perspective may be a mediating factor influencing the strength of the effect.

Peachey, Darwyn R. and Gordon McCalla (1986). Using planning techniques in intelligent tutoring systems. International Journal of Man-Machine Studies, 24, 77 -98.

The authors present an architecture for the development of intelligent tutoring systems using AI techniques developed for the planning and controlling of robots. The proposed architecture contains:

Domain knowledge base.
Student model.
Teaching operators.
Planner.
Plan executor.

The planner: develops a teaching plan tailored to a particular student; is goal oriented; simulates the effects of operator actions on a student. The executer uses the teaching plan to guide the student through the course and to call the planner if a plan revision is required. The student model contains both misconceptions and "correct" conceptions of the student. The operator is much like a rule in a production system, using a set of preconditions and expected effects to indicate expected changes in the student model due to operator actions. The knowledge bases in the architecture include knowledge about the student, subject, and teaching.

A major feature of the use of planning techniques in CAI is the use of replanning in which unsuccessful plans are identified and revised. The authors note that dynamic replanning is not a novel concept, it is founded in earlier research with some novel features: "The plan representation is NOAH-like (sacerdoti, 1977) but not hierarchical. The Planner uses a STRIPS-like problem reduction technique (Fikes and Nilsson, 1971) but generates a non-linear plan containing unordered sequences and alternatives. The Executor maintains a list of frontier steps, checks that steps achieve their relevant effects (stored in goal nodes in the plan), and reinvokes the Planner if the plan is unsuccessful. The remainer of the failed plan can be used in devising a new plan which recovers from the difficulties which twarted the failed plan."

A fully implemented CAI course has not been developed to empirically test the proposed architecture.

Rasmussen, Jens (March/April 1985). The role of hierarchical knowledge representation in decisionmaking and system management. IEEE Transactions on Systems, Man, and Cybernetics, SMC-15, 2.

In this article, Rasmussen is focusing on the characterization of decisionmaking within the domain of supervisory control of complex systems. His proposal of a hierarchical representation stems from the difficulty of developing models to allow perdiction of information processes during those events which rarely occur.

Rasmussen suggests that the decisionmakers knowledge may be structured in different levels of abstraction within a functional hierarchy. These levels of abstraction include: functional purpose, abstract function, generalized functions, physical functions, and physical form. Experimental support is provided in the article to support the need for an abstraction hierarchy for problem solving.

Implications of Rasmussen's ideas for the design of decision support systems include: an interface to support "top-down" information processing in the structuring of databases; a systematic representation, such as means-end hierarchy, is needed for modeling and the prediction of decision errors; need for experimentation in more complex, and less context-free, settings.

Reddy, Ramana Y. V., Mark S. Fox, Nizwer Husain and Malcolm McRoberts (March 1986). The knowledge-based simulation system. IEEE Software, 3,2.

This article discusses the Knowledge-Based Simulation (KBS) system developed to aid managers in answering what-if questions. KBS is an objected-oriented modeling system, KBS serves as an interpreter by accessing the model, providing simulation, model checking and data analysis.

The basic unit in the KBS model is the schema which represents objects, processes, ideas, etc. The model is a collection of Schema Representation Language (SRL) schemata. In SRL schemata form networks. Slots in the schemata may tie to schema in other slots as well as inheriting slots and values.

One important function of KBS is to aid the user in determining whether or not a goal has been reached. With KBS this is approached by specifying simulation goals as constraints on the performance of entities being modeled by the system. The status of goal achievement may be displayed by alternative methods: rating on a continuous scale from -1 (far from goal) to +1 (goal completely satisfied); and graphic representation using Kiviat graphs.

KBS also uses causal path analysis to detect causal relations embedded in the simulation model. This knowledge of causal relations is exploited to generate scenarios and achieve organizational goals.

KBS has been applied to a number of problem domains including manufacturing facilities, assembly plant, distribution and inventory. However, KBS is essentially a testbed for experimentation with various knowledge-based techniques for model building, execution, and analysis. This work has led to an interesting conclusion related to interface requirements for complex models; namely an apparent need for combining good graphics with knowledge-based techniques to ease the creation and understanding of a model.

Revesman, Mark E. and Joel S. Greenstein (January/February 1986). Application of a mathematical model of human decisionmaking for human-computer communication. <u>IEEE Transactions on Systems, Man, and Cybernetics</u>, Vol SMC-16, No. 1.

In prior research, Greenstein and Revesman had proposed the use of model-based communication as an alternative method of communication between the human and computer. The development of the model stemmed from the use of dynamic task allocation, in which the allocation of tasks that can be performed by the human or computer be assigned to both. To avoid a conflict between the human and computer each must be aware of what the other is doing. To accomplish this understanding, dialog-based communication was viewed as undesirable because of the increase in workload that would accompany its use. The model-based communication approach was proposed as a potential alternative.

The study reported in this article investigated the effects the model-based communication on the performance of the human and computer working in parallel and the effects of explicit communications from the computer to the human.

The experimental situation required subjects to monitor a display which contained the status of nine hypothetical machines. The subjects' task was to monitor the status of the machines for failures and to "repair" machines when it was cost effective. Cost effectiveness was to be determined across a number of parameters including time to repair, cost of failing to repair, and mean time between failures. Four conditions were investigated: (1) no model/no communication (computer determined repair actions without a model of predicted human actions and no information was given to subjects about the actions taken by the computer); (2) no model and communication (identical to condition 1 except that an asterick appeared on the appropriate display when the computer was repairing a specific machine); (3) model and no communication (computer used a model of human performance to select its actions based on predicted human actions but no information about the computer's actions were given to subjects); (4) model and communication (identical to 3 except that an asterick appeared in the appropriate display to indicate the computer's repair actions). The sample size of subjects was small (8) drawn from a pool of subjects used in previous experiments with the apparatus (to minimize training time).

Performance measurements made included: number of redundant actions; mean score; idle time and number of actions.

The use of the human performance model enhanced system performance. The model was of particular value in the absence of communications to the human (the computer worked around the human). While communications with the human did not significantly effect computer performance, it did improve human performance. The use of the model of human performance improved the performance of the test subjects without degrading the performance of the computer.

One interesting feature of this study is that fact that the computer to human communication was minimal (an asterick), in contrast to text messages. Further, this minimal information was demonstrated to be effective (in some measures, the communication effect masked the improvements caused by implementation of the model).

This article did not discuss the model of human performance used. The model is contained in the Revesman and Greenstein article "Development and validation of a mathematical model of human decision making for human-computer communication" IEEE Transactions on Systems, Man, and Cybernetics, vol. 16, no. 1, January 1986.

Rissland, Edwina L. (1984). Ingredients of intelligent user interfaces. International Journal of Man-Machine Studies, 21.

This article discusses specific general features of intelligent user interfaces. The author raises a wide range of questions asked in an effort to identify the type of interface to develop, such as:

> What makes an interface intelligent for the naive, occasional, or expert user?
> What sources of knowledge will be required for an interface to be what we want? What is the role of domain-, task-, tool-, user-, and context-specific knowledge?
> What is the role of user modelling?

These are a small sample of the questions the author posed. The focus of the article concerns the sources of knowledge to an interface.

The assumption is made that an intelligent interface must have access to numerous knowledge sources. Sources identified include:

1. Knowledge of the user. Knowledge includes expertise, style, preferences, and history.

2. Knowledge of the user's tasks. Knowledge includes context, purpose, and ultimate use of the result.

3. Knowledge of the tools. Knowledge includes protocols for invoking and using, default parameters, side-effects, required resources, and usage costs.

4. Knowledge of the domain (and user's task). Epistemological knowledge such as domain and task structure and representation, measures of importance, conceptual hierarchies, and doamin-specific concepts.

5. Knowledge of interaction modalities. Knowledge concerns graphics input and output displays (visual metaphors, color, menus, and windows), video, audio, language, and pointing devices.

6. Knowledge of how to interact. Knowledge includes when to intervene, monitor, adapt, enter and leave tutoring or coaching modes, length of time to wait for user responses.

7. Evaluation knowledge. Knowledge includes how to judge interaction effectiveness, and how to act on user complaints.

It is likely to be impractical for an interface to encompass the full range of knowledge for each area identified. The need to address each area to some degree, while greatly influenced by specific domains and tasks, is seen by the author as a worthwhile objective.

The author provides examples of the knowledge requirements for on-line HELP and tutoring. Parsing a user's request for HELP requires knowledge of the tool, interaction modalities and possibly knowledge of the domain and the user. Responsing to a user request for HELP requires domain-, task-, and user-knowledge. Taking a step forward and customizing HELP for a user requires a significant base of domain- and user-knowledge as well as recent history of user-system interaction. Delivering the HELP response to the user requires knowledge of modalities, the user, and how to interact.

The author identified three categories of needs for intelligent interfaces, including: services; style; and understandability. Users services provided by an intelligent interface included:

Carry out menial tasks.
Automate routine tasks.
Provide assistance with more complex tasks.
Provide easy access to tools.
Provide status information.
Provide on-line assistance and documentation.
Allow multi-tasking.

The styles in which the services are provided should:

Be helpful and forgiving.
Encourage experimentation.
Minimize errors.
Allow direct manipulation of objects and tasks.
Not get in the user's way.
Be under user control.
Adapt to the user's style (when possible, without requiring the user to provide explicit instructions).
Be unambiguous and consistent.
Have a repertoire of good presentation services.

From the understandability perspective, the intelligent interface should:

Be learnable through conceptual models, not strictly by rote.
Aid the transition from novice to expert.
Allow the user to "macro-ize" and customize tasks.

The author presents four problems that are of concern to discussions of intelligent interfaces: omniscience; control; change; and evaluation. Interfaces may be characterized along a continum of intelligence (from "dumb" to "super-human"), the omniscience problem refers to pushing the interface to the extreme of intelligence. The amount of intelligence appropriate to the interface has not been determined and is likely impacted by numerous factors relating to user, goals, tasks, domains, etc.

In a simple vein, the issue of control is not easily resolved. In some instances, the nature of the user may resolve a control issue. One heuristic noted is if the user does not use it (and it is important) then maybe the interface ought to take responsibility for it." There are numerous factors involved in attempting to resolve the control issue.

An intelligent interface must be capable of change. As important is an identification of the sort of changes needed and to what end the changes are desired. The author notes that there exists a wish list of changes (such as learning user tasks, adapting to user preferences), however, progress has not gone far toward such goals.

The fourth problem area concerns the lack of answers about interface evaluation. A limited range of methodologies are currently available (such as "keystroke-level model" analysis and the use of "benchmark" tasks).

Rolandi, Walter G. (December 1986). Knowledge engineering in practice. AI Expert, 1, 4.

This article is a general discussion of knowledge engineering as practiced by the author. Of particular interest is the author's practical discoveries:

1. Number of Knowledge Engineers to Elicit Expert Information. During the expert interviewing process never have more than two knowledge engineers. A two member team is optimum where one member serves in a quality assurance role of the communication between the expert and the knowledge engineer. In addition, the use of a single knowledge engineer in elicitation sessions is problematic since one may focus on a single reasoning line to the exclusion of others.

2. Use of Case Studies. Knowledge engineering is an inductive process and requires a large number of cases from which to generalize. The emergence of a pattern typically occurs as a function of the number of cases.
3. Identifying Relevant Data-structuring Schemes. The knowledge engineer may identify such schemes through the chunking techniques used by the expert.

4. The Problem of Contamination. The knowledge engineer should consider experimental manipulation of sources of contamination (author defined as the introduction of uncontrolled influences).

5. Avoid Excessive Interpretation. While the knowledge engineer structures the behavior of the expert, excessive interpretation should be avoided. Typically, the smaller the case study size, the more likely it becomes that excessive interpretation will occur. Quantification of behavior is one method for avoiding the problem.

6. Confirmation Bias. A knowledge engineer who seeks to confirm his particular hypothesis is likely to interpret data in accordance with that hypothesis. The potential for such a confirmation bias in the data collection and analysis must be actively guarded against.

7. Interviewing the Expert. The author cautions the knowledge engineer against "announcing" his interpretation of the expert's behavior to the expert. Also, he cautions the knowledge engineer to refrain from debating the expert about his procedures.

8. Taping Interviews. The author recommends the use of audio tape recorders and cautions against the use of video tape. A microcassette recorder tends to be less obtrusive.

Rouse, William and Nancy Morris (March 1986). On looking into the black box: prospects and limits in the search for mental models. Psychological Bulletin, Vol. 100, No. 3.

Rouse and Morris offer a review and critique of the mental models literature and identify some of the problems and limitations of this line of research.

Definitions. The authors note that while the discussion of mental models is rather widespread in the literature there is a lack of consensus, as well as a general lack of explicit definitions. Those definitions offered range from highly general to extremely specific. One confusion that exists in the literature is the lack of differentiation between knowledge and mental models. The authors propose a functional definition of mental models: "mental models are the mechanisms whereby humans are able to generate descriptions of system purpose and form, explanations of system functioning and observed system states, and predictions of future system states."

Nature of Mental Models. The authors suggest that mental models are likely to be dynamic in nature, possibly even specific to a particular situation. This poses considerable difficulty in practical study.

Researcher Bias. The authors' note a fundamental problem in the area of mental model research. Namely, the difficulty imposed by a researcher developing a mental model of the subjects' mental model. The authors suggest that the search for the truth of a mental model cannot be ascertained with complete certainty. They suggest that an uncertainty factor (analogous to Heisenberg's uncertainty principle in physics) may need to be adopted in this line of research.

Eliciting Mental Models. The problem of eliciting a mental model may be far more difficult than typically viewed. For one, some evidence suggests that expertise may be more in the form of a pattern-oriented function than researchers have previously believed. The type of technique that may be successful in eliciting a mental model is dependent upon the type of domain. Important variables in this area concern: where the model might lie on a continum between explicit and implicit subject models; and the amount of behavioral discretion (i.e., choice behavior) the subject has in the task performance. From this framework, the search for mental models is the most difficult in areas characterized by high levels o:' behavioral discretion where the nature of model manipulation is not explicit (e.g., value judgments). This is not a surprising statement.

Searching for Mental Models. While the authors are rather critical of the search for mental models, they are by no means suggesting it is a worthless endeavor. On the contrary, the authors suggest that while the ultimate "truth" in models cannot be obtained with complete certainty, the truth may not be necessary. They suggest that research move toward a more pragmatic approach.

Rumelhart, David E. and Donald A. Norman (1985). Representation of Knowledge. In Issues in Cognitive Modeling (Eds. A. M. Aitkenhead and J. M. Slack). Lawrence Erlbaum Associates, Hillsdale: NJ.

The authors discuss a number of issues concerning knowledge representation. In general, the problem of representation is concerned with the mapping concepts and relations of the represented world into concepts and relations in the representing world. The author advance the idea that representation problems are in fact notation problems.

Rumelhart and Norman identify three major controversies related to knowledge representation:

Propositional - Analogical Controversy. The propositional side of representation focuses on developing a representation through formal "statements" (using networks, schema-based structures, or logical formulae). Analogical representation, on the other hand, attempts a direct mapping between the representation world and the represented world.

Continuous - Discrete Controversy. This controversy is concerned with the acuity in the represented world. The authors note that the choice of continuous or discrete representation is basically reflecting the choice of what features are important.

Declarative - Procedural Controversy. This controversy concerns the accessibility of information to interpretative structures. In a declarative representation information is in a format that permits examination and manipulation by the interpretative processes. In a procedural representation information is not available for manipulation (a procedure is executed and its result analyzed). The authors make a case for the issue that the choice between declarative and procedural representation is context dependent.

Schmidt, LTC Robert L. (November 1985). A doctrine for command. Military Review, LXV, 11.

LTC Schmidt maintains that "control" is not needed in command and control. He notes that control is a relatively new concept and has been developed out of a need for certainty in an uncertain battle. Further, he maintains that the need for control arises from a lack of consensus about operational doctrine and its implementation.

LTC Schmidt maintains there is a need for a doctrine of command. As he notes: an understanding of the commander's intent means an understanding of his thought processes and his notions on what is important and why. Command and leadership are not separable entities.

Control will always be necessary, the degree of control is from his view a variable. The variability in control will be influenced by a number of factors including the skill of leadership, unit's discipline, morale, state of training.

Schvaneveldt, Roger, James McDonald, and Nancy Cooke (1985). The user interface in computer systems: a research program. Memoranda in Computer and Cognitive Science, MCCS-86-10. Computing Research Laboratory, New Mexico State University.

This memorandum outlines the research efforts of CRL in the area of user computer interface. Their research effort is composed of three major focuses: development of a modular user interface; application of expert system technology to developing an interface-design consultant and to implementing an intelligent interface; empirical investigations of effective user interaction techniques.

These researchers are basing their interface design on the notion of coherence models. Coherence models rest on the observation that humans have difficulty remembering incoherent facts about a topic and that understanding is facilitated by organization and memory is facilitated for material which is understood. From this perspective the researcher identify the need for an interface to provide a coherent view of the computer system structure. The researchers hypothesize that interfaces which embody coherence models will facilitate the development of a user's mental models and such models will be advantagous over models based on metaphors.

Schwenk, Charles R. (March 1984). Effects of planning aids and presentation media on performance and affective responses in strategic decision-making. <u>Management Science</u>, 30, 263 - 272.

This article presents a study of two types of decision aids for strategic decision-making. Strategic decision-making is defined as a special type of ill-structured problem-solving. The two types of decision aids studied are termed Dialectical Inquiry System (DIS) and Devil's Advocate (DA).

DIS and DA are techniques to aid in questioning assumptions. DIS focuses on examining assumptions, negating these assumptions, and developing counter proposals based on the negated assumptions. By contrast, DA identifies the assumptions and critiques them. The resulting information from both aids are presented to decision-makers in a debate format.

Schwenk reports on a study to examine the effects of DIA and DA on the generation of alternative strategies, the final choice of strategy, and participants' affective responses to the strategy formulation task. An additional factor studied was the medium of presentation of the planning aids. The experimental setting involved a case analysis task, a task which the author notes does not represent the full complexity of strategic decision-making.

Three types of planning aids were given to the participants, DIS, DA, and E (a planning committee report which contained an analysis of issues, data, and recommendations). The results of this study indicated that the DA aid encourages the generation of more alternatives than either DIS or E. Participants given DIS or DA, however, showed greater satisfaction with their performance on the task and the task results than the decision-makers receiving E. Decision-maker satisfaction was not different between DIS and DA (which represents disconfirming evidence obtained in field studies of this issue).

In terms of medium of presentation, DA presented in a live manner (via videotape) showed positive effects (reduced the functional focus, however, may not lead to increased option generation). DA appears to have positive effects when presented in the form of written memos and reports. In this case, DA encouraged the generationof strategic options and reduced the functional focus.

Simmons, Robert. F. (Spring 1986). Man-machine interfaces: can they guess what you want? IEEE Expert.

This article addresses a variety of ways in which natural language may be used in the automated arena. The author identifies four major areas: natural language interface; natural language programming; natural language dialogs; and text knowledge systems. The major problem that exists with the use of natural language is the typical requirement for active interpretation of explicit statements to determine the implicit meaning.

Natural Language Interface

Simmons addresses the issue of whether natural language interfaces can recognize human goals. Computer operations correspond to many of the primitive and concrete goals of the user, in this sense, such operations are tools by which the user attempts to achieve his complex and varied intentions.

The desirability of using natural language to communicate with computers has been recognized since the inception of computers. However, a wide range of problems have surfaced in the use of natural language interfaces, such as:

Users expect the computer to think as they think.

Users didn't know the degree of understanding possessed by the computer.

Providing users with feedback about misunderstandings appears necessary for users to be sucessful.

Combining menu selections and prompts with natural language, while useful for reducing human errors, will be extremely cumbersome if the natural language subset is very large.

In order for a svstem to respond appropriately the system needs to estimate the user intention in asking the question.

Programming in Natural Language

The second topic of discussion by the author is natural language programming. Like the interface issue, the potential use of natural language in programming dates back to the early history of computing. The current notion of the use of natural language is to allow the programmer to specify what the program should do and allow a smart compiler to determine how to do it. English may, for example, be compiled into procedural logic, which has been accomplished experimentally for limited domains (such as block problems in robotics).

The positive aspect of natural language programming is that intentions tend to organized and specified. However, the problem of guesswork may be present if specifications are not directly made by the programmer. Thus, while intentions are more readily specified with natural language programming, the capability to predict user intentions is likely to still be necessary. Due to the lack of attention natural language programming has received, the lack of documentation and HELP systems to aid the programmer in understanding errors is not a surprise but is a necessity if natural language programming will be viable in the future.

The author sees two potentially useful applications for natural language programming. One application area is for expert knowledge systems. The second application area is to aid in documentation by using a grammer to map procedures into English descriptions. This second application area is likely to require complex inference systems since intentions are not likely to be explicitly stated.

Natural Language Dialogs

A third topic addressed by the author is naturai language dialogs. The dialog between man and machine requires both the man and machine to interpret each other's explicit statements to determine the implicit meaning. The work of Appelt, SRI, on a system called Kamp is detailed. Kamp focuses on cooperative planning and problem solving. The system has two agents, a robot and an apprentice, using tools to fix a compressor assembly. The robot possesses an understanding of the object to repair as well as a model of the apprentice's knowledge. The author notes a major attribute of Kamp is in its ability to plan actions and the communication sets needed to accomplish the action.

Text Knowledge Systems

The fourth topic addressed by the author concerns text knowledge systems. Such systems are aimed at automatically translating natural language texts and articles into knowledge bases which may be browses, queried, or summarized. The major difficulty encountered in such systems relate to intentions and the lack of knowledge of the intentions for which the texts and articles were written. The author notes the needs for significant developments in the areas of discourse analysis theories and computational techniques before text knowledge systems will become practical.

Shoham, Yoav (1985). Reasoning about causation in knowledge-based systems. IEEE, 629 - 634.

Shoham offers a definition of causation which relates to the representation of temporal information. Further, Shoham presents a notion he terms causal boxes as a general paradigm for organizing temporal knowledge.

The author maintains that causation can not be defined without what he terms an ontology of time which in turn requires the construction of a theory of causal conditionals. The ontology of time proposed associates a proposition with a time interval. Collections of generating or clipping rules are termed causal conditionals. Causal boxes are organized groups of rules which may overlap or be mutually incompatible (which the author maintains, though does not show, does not suffer from many of the problems associated with adding temporal information to systems).

The author offers a brief application of his ideas to the domain of automated planning.

Shoval, Peretz (1986). Comparison of decision support strategies in expert consultation systems. International Journal of Man-Machine Studies, 24, 125-139.

Shoval offers the findings of experimental work conducted with an expert system developed for information retrieval. The expert system supports three different support strategies: participative strategy (presents intermediate feedback to user during system search and permits the user to evaluate search results and direct search); independent strategy (user is informed of results of search when completed by system); conventional strategy (information is provided to user at the user's request, system performs no evaluation on its own). The first two strategies utilize expert system capabilities, while the third is non-expert or conventional interaction in nature. Three experimental questions are asked:

Of the three strategies, which is more effective in suggesting the appropriate query terms?

Which of the strategies is most preferred by users?

Which expert system is the more accurate and fast in responding?

The results reported indicate that both expert system strategies enhanced performance. Situation plays a large part in user preference. While typically expert systems are preferred over conventional, instances are found where conventional wins out. In terms of efficiency, the "independent" strategy is more efficient than the "participative" strategy. It should be noted that the "independent" strategy contained a powerful evaluation function to identify good terms.

Shoval suggests that given the situational aspect of strategy preference, expert systems may wish to incorporate a "suggest" stage as an option for users (in this problem it would give the user an opportunity to obtain cross-references for a suggested term). He also suggests that the evaluative function be incorporated into the "participative" strategy. It should be noted that this experimentation was conducted in a laboratory setting, implementation to real world settings is needed and will require significant additions to the system utilized in this study.

Slator, Brian M., Walt Conley, and Matthew P. Anderson (1986). Towards an adaptive front-end. Memoranda in Computer and Cognitive Science, MCCS-86-54. Computing Research Laboratory, New Mexico State University.

This report discusses the implementation of a natural language interface to a graphics package. The natural language portion of the interface is based on semantic parsing theory, using a four step process: scanning, semantic marker switching, candidate derivation, and grammar searching.
User models are developed to aid the system in dealing with ambiguous utterances.

The user model developed in this effort is based on observations of user behavior. The modeling mechanism is a script-like decision matrix which provides the basis for rules for resolving ambiguities. In the interface version discussed in this paper, a single user model has been developed on the basis of a collection of users, future work will feature an individualized matrix for each user.

Sleeman, D. (1985). UMFE: a user modelling front-end subsystem. International Journal of Man-Machine Studies, 23, 71 - 88.

The focus of this article is on a data-driven user modeling front-end subsystem (UMFE). The UMFE subsystem develops a user model from which appropriate responses to user queries of a main system (e.g., expert system) are generated. The user models are developed on the basis of as few questions as possible in an effort to determine the user's level of sophistication. The paper discusses the use of UMFE used in conjunction with NEOMYCIN.

UMFE makes a number of assumptions including the following: (1) users know what they know; (2) user knowledge is stable and context independent; (3) the relative difficulty of concepts is consistent across users; (4) causal explanation understanding may be aided by omitting difficult intermediate steps.

UMFE knowledge base:

1. List of domain concepts organized into a hierarchical data base (in this case, NEOMYCIN). Each concept has a difficulty and importance rating (1 - 10 scale) associated with it. The ratings are particular to a target population.

2. Inference rules state that if a particular concept is known/not known, then the associated set of concepts are also known/not known.

In simplistic terms, the user model identifies the highest points in the hierarchical data base where the user knows the concepts (through direct query). The UMFE subsystem assumes knowledge of the concepts below and lack of knowledge above. Inferencing rules are used to identify additional concepts likely to be known to the user. The model may be modified based on additional information and methods of conflict resolution are utilized (which are demonstrated in the paper).

Sohn, Tae-Won and Julius Surkis (1985). Systems approach for constructing dynamic motivation models. Cybernetics and Systems: An International Journal, 16, 145 - 170.

Traditional approaches to constructing models in organizational studies are descriptive and have not dealt with dynamic behavior. The authors demonstrate the applicability of a cybernetics and systems framework to the study of organizational behavior.

Three stages for the System Dynamics modeling process are identified:

1. In terms of feedback linkages, construct a causal loop model.
2. Develop a detailed flow diagram based on the causal loop model.
3. Use the DYNAMO computer simulation language formulate the systems equations.

Understanding a system in terms of feedback loops is a major purpose of the System Dynamics approach. Implicit in this is the notion that the feedback loop is key to representing dynamic behavior. As defined the feedback loop is a closed loop of causes and effects. In this way, building a causal loop requires a sequence of causal hypotheses among variables or system elements and the identifying the direction of influence.

The flow diagramming stage of the modeling process requires the identification of system states and activities. In the Systems Dynamics framework, system states are represented as levels and activites or decisions are represented by rates. The value of a level reflects represent the values of system variables at points in time.

DYNAMO interprets system equations and simulates behavior over time. The smaller the time interval viewed, the more precise a view achieved of dynamic behavior of the system. The authors provide an elaboration of the use of DYNAMO by solving the differential equations developed by other researchers to quantify their conceptual model of organizational behavior (solutions were identical).

Tanksley, Major (P) David M. (November 1985). C2: finding the middle ground. Military Review, LXV, 11.

Major Tanksley discusses two sides to the command and control issue. On one side, he finds the technocrats who maintain that technological advances will provide many of the answers to C2 problems. The other side argues against technology in favor of the "great men" who win battles thrc gh individual genius.

The author maintains that the Army's theory of command and control must be based on ordinary individuals, not geniuses. Technology, rather than being an answer in and of itself, can be applied and provide both a thinking and educational foundation.

Major Tanksley believes that the narrow focus of C2 must be expanded into C3I to aid in removing the vertical stovepipe organizations and thought processes which are problematic to effective C2. He maintains that at the heart of future successes will be the ability of both humans and technology to appropriately filter information ("gleaning the crucial bits of information for decisionmaking").

Teston, Major Harvey A., Jr. (November 1985). Command and confusion at the NTC. Military Review, LXV, 11.

This article presents a description of the experiences of a battalion task force at the NTC. This task force was defeated and the author offers some insights into possible reasons. The performance deficiencies of the task force included: loss of time, poor METT-T analysis, poor TOC location, misunderstanding of the commander's intent, poor use of scouts, absence of flank coordination, poor position of the task force commander, and poor reporting. The deficiencies were the result of failures in command and control. A few of the deficiencies are discussed below.

The author notes that all planning errors were the result of the poor METT-T analysis. He notes that successful plans fit the circumstance, rather than trying to create the circumstances to fit the plan.

The commanders and staffs upon receipt of the warning order spent a great deal of time on terrain analysis and movement planning. This use of time precluded the commanders from actual terrain reconnoitering.

The positioning of the TOC offered excellent security. However, it was so far to the rear that communications and accessibility were hampered.

The commander's intent was not understood by the subordinates. There was a lack of guidance on disengagement criteria. Briefbacks were not conducted by the commander to insure understanding.

The author points out that insufficient training attention is given to command and control.

Williams, Antony (July 1986). An architecture for user interface R&D. IEEE Computer Graphics and Applications, 6, 7, 39-50.

Williams discusses the user interface work being accomplished at the Informatics Division of the Rutherford Appleton Laboratory. Underlying their work is the characterization of interaction between the human and computer as a dialog, analogous to conversations between humans. The research being conducted at RAL is trying to avoid time-sequential dialogs by "the spatial encoding of information in visual representations of the data and control objects supported by the application."

An architecture is presented within the context of the UNIX operating system. The architecture supports multiple concurrent activities through multiple windows. One objective of the proposed architecture is to support the development of a user interface management system (UIMS). Conceptually, the UMIS is modular, separating user interface concerns from application concerns. As Williams notes, the requirements placed on a general purpose UMIS are quite varied and to date, an interface of this type has yet to be demonstrated.

Williams, Tom (January 15, 1986). Graphics ICs increase on-chip functions and intelligence. Computer Design.

Williams offers an overview of the development of VLSI graphics controller chips and the off-the-shelf products available for the design of graphics systems. Williams notes that typically text and graphics are integrated (this is viewed as a consumer requirement).

Some approaches integrate text and graphics by reserving a separate window or scrolling area of the screen, such as Intel's 82730 text coprocessor. NEC's PD7220A partitions memory into graphics and text areas. Xerox PARC accomplishes a truer integration through Bit Boundary Block Transfer (BITBLT or Raster Ops). TI's TMS34061 and Hitachi's HD63484 integrate text and graphics through the use of Raster Ops. Raster Ops allows for the defining of an N x N array of pixles in the frame buffer, operations may be performed on the pixels in the array, and the results may be moved to an area of the frame buffer. Since operations are performed on a pixel-by-pixel basis, transfers do not require alignment on word boundaries in display memory.

Advances in graphic ICs are permitting designers to have more options for both configuring and controlling display memory. Examples of advances capabilities include Motorola's 68490 raster grqaphics display processor, NEC's 7220A, and TI's TMS34070 color palette IC.

Winograd, Terry (1985). What does it mean to understand language? In Issues in Cognitive Modeling (Eds. A. M. Aitkenhead and J. M. Slack). Lawrence Erlbaum Associates, Hillsdale: NJ.

Winograd has long been involved in research in the area of designing computer systems to understand language. Winograd developed SHRDLU which demonstrated the potential for a seemingly natural exchange between a human and computer. This demonstration was successful in a limited domain with a carefully constrained dialog. Winograd notes a number of shortcomings of SHRDLU.

Winograd points out that there are two different starting points to understanding language, objectivity and subjectivity. Objectivity views an utterance as having meaning by its correspondence to a state of affairs. Subjectivity, on the other hand, views an utterance as having meaning by triggering processes within a hearer. The cognitive structure of the hearer is dependent upon prior history and current processing activity. There are inherent problems associated with either extreme.

Winograd suggests there are four domains in which language may be described:

Domain of linguistic structure. This domain looks for regularities in the patterning of structural elements in utterances and text.

Domain of correspondence between linguistic structures and the world. This domains looks for regularities in the correspondence between linguistic object structure and the states of affairs in the world those objects describe.

Domain of cognitive processes. This domain looks for regularities in the cognitive structures and processes of the person or machine that generates or interprets them.

Domain of human action and interaction. This domain is the focus of Winograd's current work. This domains looks for regularities in the "network of actions and interactions within a human society. An utterance is a linguistic act that has consequences for the participants, leading to other immediate actions and to commitments for future action."

Yager, Ronald R. (1985). Explanatory models in expert systems. International Journal of Man-Machine Studies, 23, 539 - 549.

The problem space considered in this article concerns the degree to which a set of elements may be used to explain a set of evidence. Yager offers a measure for such a calculation. Yager's approach is in contrast a rule-based approach and is extends the work of Reggia, Nau, and Wang (using a set, theoretic approach) by allowing for possibilistic associations.

Consider that the problem is in realm of medical diagnosis. A set of objects is termed a manifestation set (M). A second set of objects is termed the disorder or causal set (C). A fuzzy relation, R, which is a measure of the possibility of an element of set C being the cause of an element of set M (the greater the value of R the easier it is for the C element to be the cause of the M element). From the knowledge engineering perspective, R is the knowledge base which is independent of any particular case. A final element in the problem is a fuzzy subset, E, which relates to the evidence of an element of M being manifested in a particular case.

The problem addressed by Yager relates to a subset of C which is represented as D (for diagnosis). In this manner, D suggests the subset of disorders that may be present in a particular case. The determination of D is related to the degree to which the elements in D explain or account for the evidence in E. Yager offers mathematical justification for his measure along with a brief discussion of the problem of reducing requirements placed on the diagnosis set.

Zemankova, Maria and Abraham Kandel (1985). Implementing imprecision in information systems. Information Sciences, 37, 107 - 141.

The authors perspective is that data base management systems frequently serve as input to human problem solving and decision making in uncertain environments. Further, since humans deal with uncertain, imprecise, incomplete, or vague information, the need exists for automation to allow the representation and manipulation of imprecise information in an effort to model human reasoning.

In answer to the need to represent and manipulate imprecise information the authors present a fuzzy relational data base (FRDB) system. The target problem is the handling of uncertainty or vagueness in natural language queries. The FRDB is organized into three categories:

Value data base which stores actual data values.
Explanatory data base which consists of a collection of semantic relations or functions explaining how to compute the degree of compliance of data values with user's query.
Transla- -- rules.

The explanatory data base permits individualization of the FRDB while keeping the actual value data base constant. In this way, the explanatory data base reflects the subjective knowledge profile of a given user and may evolve over time to account for changes in user needs and expertise level.

The authors see the value of FRDB as an information base for decision support systems. Their rationale stems from that the fact that the problems dealt with in DSS are typically possibilistic. The authors also note the potential application to pattern recognition.

Zissos, Adrian Y. and Ian H. Witten (1985). User modelling for a computer coach: a case study. <u>International Journal of Man-Machine Studies</u>, 23, 729 - 750.

This article reports the findings of a prototype coach for the EMACS text editor. Basically, a coach monitors the user's interaction with a system and offers advice on system use. The coach focuses on the detection of "bad plans" (i.e., inefficient use) and identification of important options not used by the user.

The authors identify "bad plans" as a difficult option to handle well. Efficient use of "bad plans" detection requires the cataloging of a large number of plans to be effective. A task they regard as both difficult and tedious. To make "bad plans" detection meaningful may also require contextual information.

One issue raised by the authors but not addressed in their research concerns ways to present advice. They note that recognizing inefficiency is one thing, but presenting advice on compensating for the inefficiency is a different matter, one which requires further research.