KNOWLEDGE-BASED CONTROL OF AN ADAPTIVE INTERFACE

Final Report

NASA/ASEE Summer Faculty Fellowship Program-1989

Johnson Space Center

Prepared By:	Roy Lachman, Ph.D.
Academic Rank:	Professor of Psychology
University & Department:	University of Houston Department: Psychology Houston, Texas 77204
NASA/JSC	
Directorate:	Space and Life Sciences
Division:	Man-Systems Division
Branch:	Crew Station Branch
JSC Colleague:	Frances E. Mount, MS
Date Submitted:	August 25, 1989
Contract Number:	NGT 44-005-803

ABSTRACT

The analysis, development strategy, and preliminary design for an intelligent, adaptive interface is reported. The design philosophy couples knowledge-based system technology with standard human factors approaches to interface development for computer workstations. An expert system has been designed to drive the interface for application software. The intelligent interface will be linked to application packages, one at a time, that are planned for multiple-application workstations aboard Space Station Freedom. Current requirements call for most Space Station activities to be conducted at the workstation consoles. One set of activities will consist of standard data management services (DMS). DMS software includes text processing, spreadsheets, data base management, etc. Text processing was selected for the first intelligent interface prototype because text-processing software can be developed initially as fully functional but limited with a small set of commands. The program's complexity then can be increased incrementally. The intelligent interface includes the operator's psychometric profile, prior performance history, and a current keystroke performance stack within a conventional expert system database. Error detection and other productions in the form of conditionals between the operator's behavior and three types of instructions to the underlying application software are included in the rule base. A conventional expert-system inference engine searches the data base for antecedents to rules and sends the consequents of fired rules as commands to the underlying software. Plans for putting the expert system on top of a second application, a database management system, will be carried out following behavioral research on the first application. The intelligent interface design is suitable for use with ground-based workstations now common in government, industrial, and educational organizations.

INTRODUCTION

Integrated computer work stations are now under development at major U.S. corporations and at NASA. It is likely that in the 1990s, a variety of professionals will practice most aspects of their craft at an integrated work station. The most ambitious program, in this regard, is the Multipurpose Applications Console that will be used by the crew of U.S. Space Station Freedom. Scientific research as well as standard operations aboard the Space Station will require new approaches to human performance. Crews will operate and maintain equipment of unprecedented technological sophistication and complexity. Simultaneous monitoring of multiple systems, documentation access, data management, and command and control will be routinely conducted at integrated work-station consoles. Significant components of Space Station design now incorporate machine intelligence in the form of expert systems, robotics, and advanced automation. One aspect of machine intelligence in Space Station Program Requirements is of particular interest to cognitive psychologists: standard computer services with knowledge-based components and an interface that adapts to the unique characteristics of individual crew members.

This paper explores approaches to instantiating intelligence in the computer-human interface in the form of specific knowledge-based technologies. The initial effort to conceptualize an approach to this problem, formulate a design philosophy, and to develop a prototype is described. The relatively slow pace of research on adaptive systems and its significance is considered. Cognitive psychology provides one major orientation. The emerging fields of cognitive science and cognitive engineering are also a major source of concepts and methodology in this study of basic and applied problems in human and machine cognition.

NEEDS

The need is particularly acute for adaptive capacities in the integrated work stations of manned spacecraft. The logistics and costs of habitation aboard Space Station, for example, necessitates a minimum size of crew. Each crew member, therefore, will be assigned multiple responsibilities. Crew skill levels will vary with frequency of performance of particular tasks and native ability. The interface for a computer-based activity that is most suitable for peak skills can be highly unsuitable when that skill has diminished. Extended habitation in space may require that computers adapt to fixed and variable human characteristics. Intelligent machines, with knowledge-based control of the computer-human interface, are essential to the effective deployment of advanced computer-human interactive technologies as well as to raising productivity for computer based tasks that are now commonplace. A number of cognitive science problems are related to developing an intelligent interface that adapts its displays and command structure to each individual system user. Psychometric and performance data of an individual must provide the knowledge base as well as the rule-driving data base to adapt the control structure and command dialogue of work-station services to fixed characteristics and the current performance level of the user. Individual capacities measured previously and performance collected continuously at the operator-system interface have the potential for interpreting human performance, competency levels, errors and other unsuitable performance patterns. Information obtained unobtrusively from embedded performance measures and directly from queries to the human operator serves as the input to expert-system software modules. Elements of this information are the antecedents to the firing of rules in a production-system architecture. A knowledge-based work-station interface possesses the potential for reducing errors and significantly enhancing system performance and human productivity on Space Station and elsewhere.

DESIGN PHILOSOPHY

The design and development of any engineering artifact, including software systems, entails an implicit design philosophy. It is useful to articulate the philosophy so that inadequacies can be recognized and corrected before changes become too costly. The current design philosophy is consistent with proven approaches reported in the literature (Rouse, Geddes, and Curry, 1987-88) and is summarized as follows.

- Develop operator centered design, leave the crew in charge
- Overcome human sensory and cognitive limitations
- Amplify existing human abilities
- Automate where human performance is below an acceptable level
- Sophistication of support for each application can vary from simple methods to large scale knowledge-based systems
- Develop support systems incrementally within a range of the application subsystems.
- Map expert-system technology on to the problem

The major justification for an operator-centered approach stems from the impossibility of ruling out the need for human intelligence to solve problems. Unanticipated events happen; only human intelligence can deal with unanticipated events. Automated systems including robotics and expert systems can only deal with events that were anticipated, and then only when plans were specifically enacted to deal with those events. There are numerous examples of the failure to anticipate potentially catastrophic events. The examples include the fuel cell failure on STS-2 and the false warning of an impending Soviet nuclear attack in 1960 due to unanticipated radar reflections from a moon rise.

An incremental approach with more than one application is the recommended method of development because cognitive variables interact with tasks and the tasks interact with each other to change difficulty levels, error types, and other performance parameters. The last item in the statement of design philosophy calls for the development of expert systems. Expert systems are the technology of choice for an adaptive interface because of their power and flexibility and their capacity to capture the heuristics of operator performance.

RESEARCH IN COGNITIVE PSYCHOLOGY

Contemporary theories of intelligence view adaptability as essential to intelligent behavior; this likewise must be part of any reasonable concept of machine intelligence. Understanding the processes by which adaptability is incorporated into a knowledge-based system requires more than AI system building; basic research in cognitive psychology is also needed. Research to isolate the heuristics that facilitate machine adaptation to the human user's fixed and variable capacities requires combining research in Artificial Intelligence and Psychology. Part of this project is devoted to the study of general human expertise and skilled performance in selected work-station tasks. Developing an adaptive interface requires inquiry into a machine's ability to recognize user and task properties. In order to effectively guide a person through a complicated task, a machine (or even a human helper) must possess specific sets of knowledge of the person and the task. The expert system knowledge base must possess detailed information as to the current stage of the task and the user's performance. At certain machine-state operator-input cycles, it is essential to determine the user's intentions, at least partially, for effective machine intervention. The system will provide more realistic and improved guidance if it possesses rules and facts dealing with user characteristics related to performance such as general task knowledge, aptitude, and motivation.

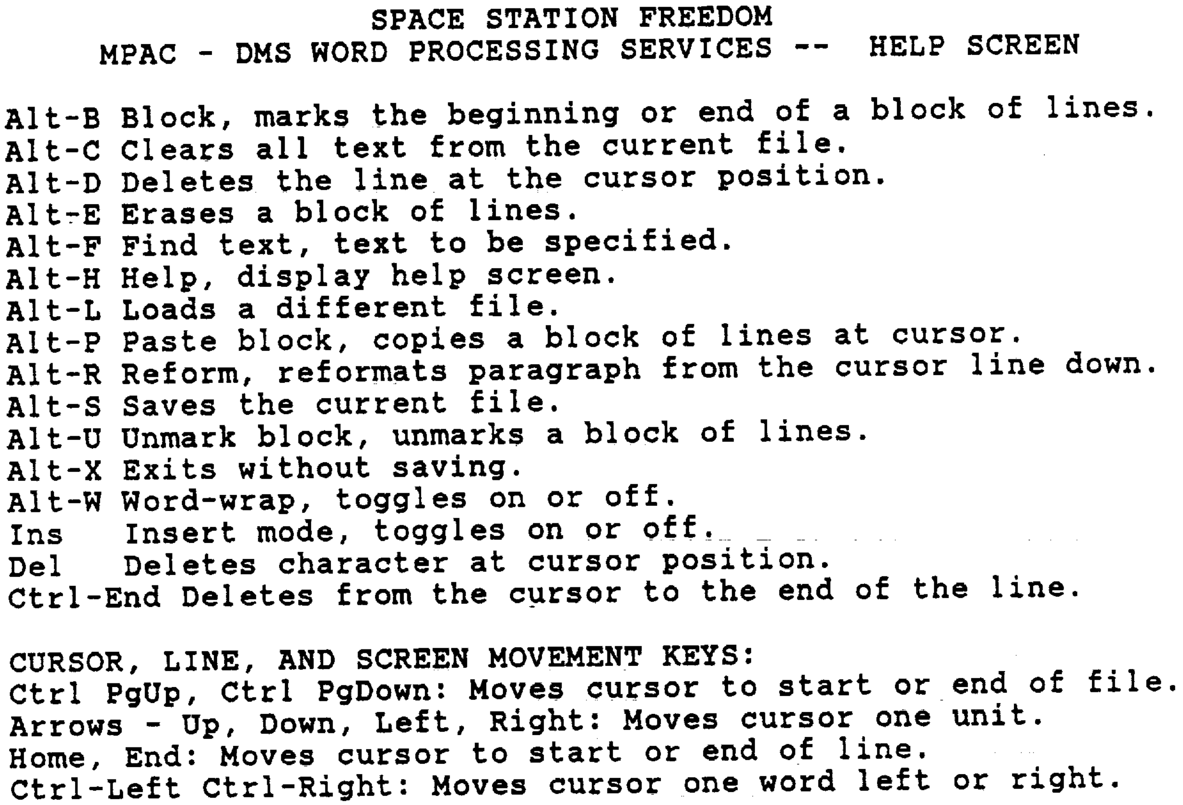
Precisely formulating the user's task state knowledge, and the user's intentions relative to the current machine state (i.e. what the user is trying to do right now) is the goal of continuing cognitive research that is important to the project. Machine-detectable user information from keystrokes is currently the primary basis for formulating rules that, when fired, produce the adaptation of a work station's display and command structure. It is important to detect increasing operator expertise. Anderson, Farrell, and Sawyers (1984) have shown that the development of expertise entails a shift from the use of declarative knowledge to the development of problem specific productions that are subsequently used in new situations. The acquisition of productions can be detected by a decrease in latency and recurring keystroke patterns. Automaticity, increases in quantity of knowledge, schema driven problem solving, and changes in chunking patterns are all indicative of growth of expertise (Chi, Glaser, and Reese, 1982). Several of these indices may be detectable by changes in keystroke patterns and response latencies to specific machine states.

DATA MANAGEMENT SERVICE SIMULATIONS

Work-station hardware and software components in current Space Station specifications (NASA SSP 30261, 1989) are to be designed for modularity. The software will contain a number of modules for crew operations support. These will provide, among other things, Data Management Services. DMS will support a full spectrum of computer services including document retrieval, word processing, and data base access. Two DMS functions are selected for simulation, namely, text editing and data base management. The "simulated" text editor has been developed with a limited command set. Figure 1 (next page) shows the help screen with the limited command set currently in use. The editor is an extension of software developed to enhance comprehension of technical text (Lachman, 1989b).

THE EXPERT SYSTEM ADAPTIVE INTERFACE

The core idea of an adaptive interface is to reverse the usual role of human and computer and to equip the computer with mechanisms to learn about the user and adapt accordingly. Several approaches to human performance, task characteristics, and adaptation methodology have been proposed in the computer-human interface literature. The approaches include statistical analyses of performance, the GOMS model (Card, Moran, and Newell, 1983), generative grammar (Hoppe, 1988; Payne and Green, 1986), production systems (Polson and Kieras, 1985) and hybrid formalisms (Tyler, 1988). The first of these approaches, statistical analysis of keystroke command selection and error performance, is insufficient by itself to drive an autoadapting system. But it is included as part of a more effective design, the current expert-system driver for work-station interfaces. A generative grammar is formally equivalent to a production system (c.f. Lachman, 1989a) but lacks the latter's extent of implementation and demonstrated utility. A psychometric approach alone is also incapable of effectively informing adaptation software but is potentially useful in combination with other methodology, primarily



```
                    SPACE STATION FREEDOM
     MPAC - DMS WORD PROCESSING SERVICES --  HELP SCREEN

Alt-B Block, marks the beginning or end of a block of lines.
Alt-C Clears all text from the current file.
Alt-D Deletes the line at the cursor position.
Alt-E Erases a block of lines.
Alt-F Find text, text to be specified.
Alt-H Help, display help screen.
Alt-L Loads a different file.
Alt-P Paste block, copies a block of lines at cursor.
Alt-R Reform, reformats paragraph from the cursor line down.
Alt-S Saves the current file.
Alt-U Unmark block, unmarks a block of lines.
Alt-X Exits without saving.
Alt-W Word-wrap, toggles on or off.
Ins   Insert mode, toggles on or off.
Del   Deletes character at cursor position.
Ctrl-End Deletes from the cursor to the end of the line.

CURSOR, LINE, AND SCREEN MOVEMENT KEYS:
Ctrl PgUp, Ctrl PgDown: Moves cursor to start or end of file.
Arrows - Up, Down, Left, Right: Moves cursor one unit.
Home, End: Moves cursor to start or end of line.
Ctrl-Left Ctrl-Right: Moves cursor one word left or right.
```

Figure 1. Print out of the help screen showing the current command set of the text editor. The help screen is called by Alt-H.

consisting of a production-system expert system (Buchanan and Shortliffe, 1984; Clancey, 1985; c.f. Lachman, 1989a). The Carnegie Group (Hayes, 1988) recommends that an independent knowledge base be used to drive the interface for expert systems. The rationale for this approach includes ease of rule modification and a guaranteed consistency between the interface and the internal system in how problems are viewed. The rule-based expert system that drives the work-station interface is based on the Carnegie recommendations (Hayes, 1988). The rule-based system exercises partial control of command syntax, command execution, and the content of screen displays and advisory messages. Finally, it should be noted that despite the various articles appearing on adaptive interfaces over the last decade, no one has reported a fully functional system. This is due, in part, to the necessity of a full scale knowledge engineering effort with all the associated costs. The current approach is feasible because it starts with a scaled down but fully-functional system and proceeds incrementally.

The use of high-level domain experts has not been required for prototype development. The DMS services selected for prototyping with a knowledge-base are common computer tasks with an established literature (Carroll, 1987; Lachman, 1989b). The literature and common knowledge concerning text editors are sources of information for construction of the rule base. This is supplemented by a limited amount of direct observation of subjects performing the targeted task at a console.

Once a tentative rule base is established, the adaptive system requires facts concerning the service in use, the state of the system, and the current performance of the user. Relevant facts are stored in the data base of the expert system and are compared continuously to the antecedents of rules. When a match is obtained between the user's command entry and a rule, that rule will fire. The firing of any rule either adds a new fact to the data base or issues a command to the underlying DMS system. Firing of a set of rules designed to affect the application triggers commands that adapt the display and command structure of the interface. The user then may be presented with automatic assistance or with direct advice. The facts are obtained from several sources, including nonobtrusive and embedded speed and accuracy measures of keystrokes or mouse movements at the work-station console; rule-selected automatic queries to the operator; unsolicited operator inputs and requests for help within predefined parameters; and software monitoring of the particular state of the DMS service being used.

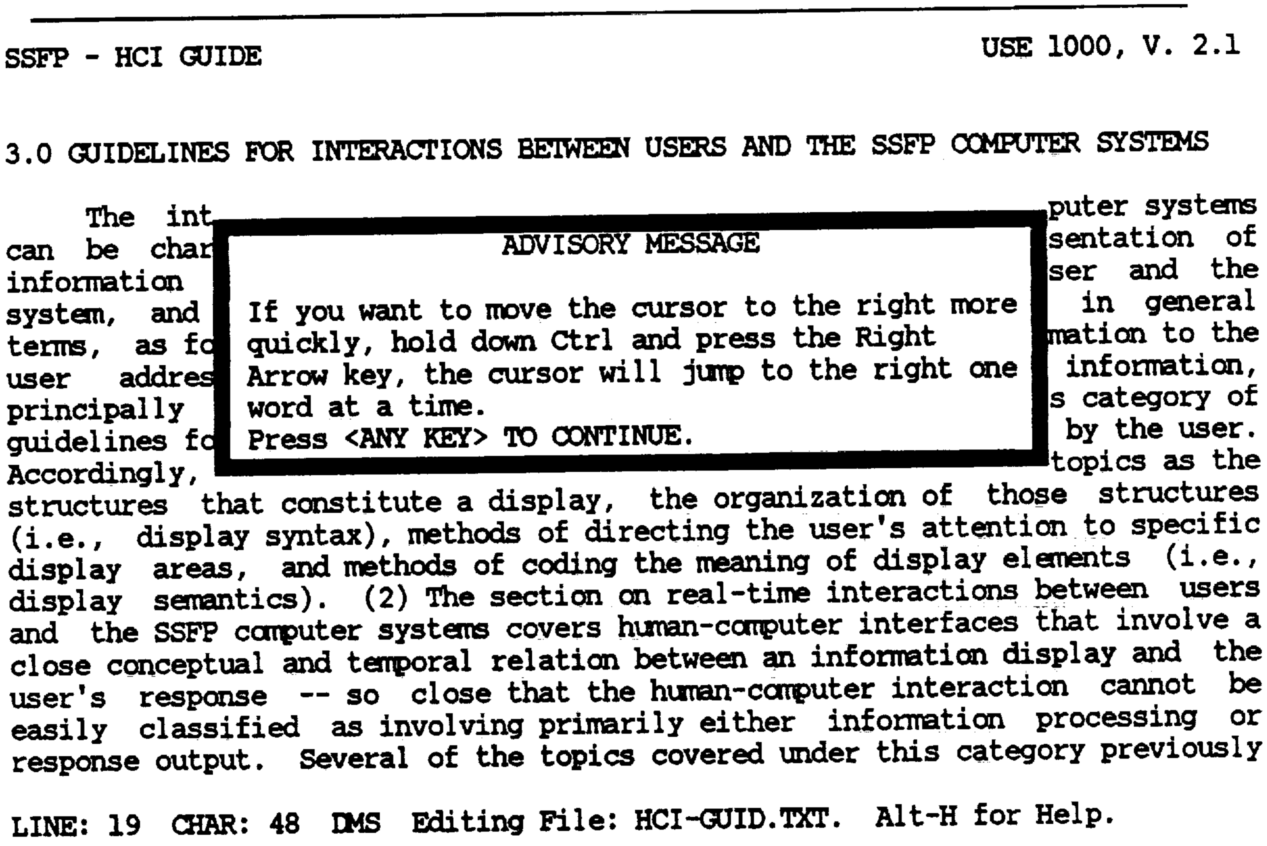
An example of the action produced by the firing of one type of rule is shown in Figure 2. The advisory message is windowed over the visible screen of the text editor by Rule 7 which has the following form.

```
RULE 7

IF RESPONSE(last) = RIGHT.ARROW.KEY
AND IF RESPONSE(last - 1) = RIGHT.ARROW.KEY
AND IF RESPONSE(last - 2) = RIGHT.ARROW.KEY
AND IF RESPONSE(last - 3) = RIGHT.ARROW.KEY
AND IF RESPONSE(last - 4) = RIGHT.ARROW.KEY
AND IF RESPONSE(last - 5) = RIGHT.ARROW.KEY
AND IF LINE.LENGTH(CURRENT.LINE) - CURSOR.POSITION > 10

THEN PRINT MESSAGE RIGHT.WORD
```

The rule states that if the right arrow key was pushed during the last six keystrokes and if the cursor is now 10 or more character positions from the last character on the line, then print the window message shown in Figure 2.



```
SSFP - HCI GUIDE                                             USE 1000, V. 2.1

3.0 GUIDELINES FOR INTERACTIONS BETWEEN USERS AND THE SSFP COMPUTER SYSTEMS

   The int                                                    puter systems
can be char              ADVISORY MESSAGE                     sentation of
information                                                   ser and the
system, and  If you want to move the cursor to the right more  in general
terms, as fo quickly, hold down Ctrl and press the Right      mation to the
user   addres Arrow key, the cursor will jump to the right one  information,
principally  word at a time.                                  s category of
guidelines fo Press <ANY KEY> TO CONTINUE.                     by the user.
Accordingly,                                                  topics as the
structures that constitute a display, the organization of those structures
(i.e., display syntax), methods of directing the user's attention to specific
display areas, and methods of coding the meaning of display elements (i.e.,
display semantics). (2) The section on real-time interactions between users
and the SSFP computer systems covers human-computer interfaces that involve a
close conceptual and temporal relation between an information display and the
user's response -- so close that the human-computer interaction cannot be
easily classified as involving primarily either information processing or
response output. Several of the topics covered under this category previously

LINE: 19 CHAR: 48 DMS Editing File: HCI-GUID.TXT. Alt-H for Help.
```

Figure 2. An advisory message that is windowed onto the screen by Rule 7. The rule is described in the text.

The rule could have executed an application command instead of sending a message to the operator. Rule 8, for example, states that if the last five keystrokes moved the cursor five words to the right and there are at least 15 characters left on the line, then execute the command to jump the cursor to the end of the line. The assumption is that the operator wanted to get to or near the end of the line and forgot the command. The command execution can be accompanied by an advisory message explaining what was done and tutoring the operator on the appropriate keyboard input for the command. Obviously, command execution rules as well as rules that alter the screen display and command entry features require behavioral research to determine their effects on productivity.

Thus, an important part of the task of the knowledge-based system is to determine an astronaut's intentions when entering commands for a particular DMS service. One aspect of predicting user intentions is straightforward. If the expert system knows (has in its data base) the context for commands that are currently being entered, then the redundancy in the command structure will allow determination of the user's intentions. For a command sequence with less than total redundancy, the point is to capture the rules used by an expert or knowledgeable observer in determining the user's intentions. A domain expert cannot perfectly determine other people's intentions; this determination can only be done partially. A machine can do no better than the expert whose rules are being implemented. But people can partially predict the intentions of others and provide appropriate directions and excellent advice; an expert system machine implementation can do as well.

The positive outcome of the this proof-of-concept study and prototype development does not mean that large scale deployment is unproblematic. Intensive and well designed behavioral experimentation is essential. In particular the strategy for sampling of subjects must be developed meticulously for realistic generalization to specialized and global populations. Up scaling to highly complex commercial systems and integrating a very large rule base present significant technical problems. But, these can be solved.

To summarize the second half of this article, the architecture of the adaptive interface is similar to the three component architecture of a conventional expert system. First, there is a database with the operator's psychometric profile, prior performance history, and the output of the current keystroke performance stack. Second, there is a rule base with error detection rules and productions in the form of conditionals between the operator's behavior and three types of instructions to the underlying application software. Third, there is an inference engine that searches the data base for antecedents to rules and sends the consequents of fired rules as commands to the underlying software. This structure appears sufficient for the construction of an adapting interface.

REFERENCES

Anderson, J. R., Farrell, R., and Sauers, R. (1984). Learning to program in LISP. *Cognitive Science, 8,* 87-129.

Card, S. K., Moran, T. P., and Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, NJ: Erlbaum.

Buchanan, B. G. and Shortliffe, E. H. (Eds.) (1984). *Rule-based expert systems*. Reading, MA: Addison-Wesley.

Carroll, J. M. (Ed.) (1987). *Interfacing thought: Cognitive aspects of human computer interaction*. Cambridge, MA: MIT Press.

Chi, M. T. H., Glaser, R., and Rees, E. (1982). Expertise in problem solving. In R. Sternberg (Ed.), *Advances in the psychology of human intelligence*. Hillsdale, NJ: Erlbaum.

Clancey, W. J. (1985). Heuristic classification. *Artificial Intelligence, 27*, 1-67.

Hayes, P. J. (1988). Using a knowledge base to drive an expert system interface with a natural language component. In J. A. Hendler (Ed.), *Expert systems: The user interface*. Norwood, NJ: Ablex Publishing.

Hoppe, (1988). Task-oriented parsing - A diagnostic method to be used by adaptive systems. In *Proceedings of CHI '88* (pp. 241-247). New York: ACM.

Lachman, R. (1989a). Expert systems: A cognitive science perspective. *Behavior Research Methods, Instruments, and Computers, 21,* 195-204.

Lachman, R. (1989b). On-line reading and comprehension aids for expository text. *Human Factors, 31,* 1-15.

NASA SSP 30261 (1989). *Preliminary draft: Architectural control Document, Data Management System*. Draft Version 2, Revision D. Space Station Freedom Program Office.

Payne, S. J. and Green, T. R. G. (1986). Task action grammars--A model of the mental representation of task languages. *Human-Computer Interaction*, 2, 93-133.

Polson, P. G., and Kieras, D. (1985). A quantitative model of the learning and perform-ance of text editing knowledge. In Proceedings of CHI '85. New York: ACM.

Rouse, W. B., Geddes, N. D., and Curry, R. E. (1987-88). An architecture for intelligent interfaces: Outline of an approach to supporting operators of complex systems. Human-computer Interaction, 3, 87-122.

Tyler, S. W. (1988). SAUCI: A knowledge-based interface architecture. In Proceedings of CHI '88 (pp. 235-240). New York: ACM.