HUMAN FACTORS IN AUTOMATED AND ROBOTIC SPACE SYSTEMS: PROCEEDINGS OF A SYMPOSIUM

Thomas B. Sheridan, Dana S. Kruser, and
Stanley Deutsch, editors

Committee on Human Factors

Commission on Behavioral and Social Sciences and Education

National Research Council

(NASA-CR-182495) HUMAN FACTORS IN AUTOMATED AND ROBOTIC SPACE SYSTEMS: PROCEEDINGS OF A SYMPOSIUM. PART 1 (National Academy of Sciences - National Research Council) 475 p CSCL 05H H1/54

N89-24792

Unclas
0124744

NATIONAL ACADEMY PRESS
WASHINGTON, D.C. 1987

This work relates to the National Aeronautics and Space Administration Contract Number NASW-4071. However, the content does not necessarily reflect the position or the policy of the government, and no official endorsement should be inferred.

Available from:
Committee on Human Factors
Commission on Behavioral and Social Sciences and Education
National Research Council
2101 Constitution Avenue, N.W.
Washington, D.C. 20418

SYMPOSIUM STERING GROUP

THOMAS B. SHERIDAN (Chair), Department of Engineering and Applied Psychology and Department of Mechanical Engineering, Massachusetts Institute of Technology

OSCAR GRUSKY, Department of Sociology, University of California, Los Angeles

ALLEN NEWELL, Computer Science Department, Carnegie-Mellon University

JUDITH REITMAN OLSON, Graduate School of Business Administration, University of Michigan

ROBERT C. WILLIGES, Department of Industrial Engineering and Operations Research, Virginia Polytechnic Institute and State University

HARRY L. WOLBERS, Space Station Program, McDonnell Douglas Astronautics Company, Huntington Beach, California

STANLEY DEUTSCH, Study Director
DANA S. KRUSER, Project Coordinator/Consultant
BEVERLY M. HUEY, Research Associate/Consultant
ELIZABETH F. NEILSEN, Research Assistant
MARGARET A. CHENG, Administrative Secretary

COMMITTEE ON HUMAN FACTORS
1986 - 1987

THOMAS B. SHERIDAN (Chair), Deparment of Engineering and Applied Psychology and Department of Mechanical Engineering, Massachusetts Institute of Technology
NANCY S. ANDERSON, Department of Psychology, University of Maryland
CLYDE H. COOMBS, Department of Psychology, University of Michigan
JEROME I. ELKIND, Information Systems, Xerox Corporation, Palo Alto, California
OSCAR GRUSKY, Department of Sociology, University of California, Los Angeles
ROBERT M. GUION, Department of Psychology, Bowling Green State University
DOUGLAS H. HARRIS, Anacapa Sciences, Santa Barbara, California
JULIAN HOCHBERG, Department of Psychology, Columbia University,
THOMAS K. LANDAUER, Information Sciences Division, Bell Communication Research, Morristown, New Jersey
JUDITH REITMAN OLSON, Graduate School of Business Administration, University of Michigan
RICHARD W. PEW, Computer and Information Sciences Division, Bolt Beranek and Newman Laboratories, Inc., Cambridge, Massachusetts
STOVER H. SNOOK, Liberty Mutual Insurance Company, Hopkinton Research Center, Hopkinton, Massachusetts
CHRISTOPHER I. WICKENS, Aviation Research Institute, University of Illinois
ROBERT C. WILLIGES, Department of Industrial Engineering and Operations Research, Virginia Polytechnic Institute and State University

HAROLD P. VAN COTT, Study Director
STANLEY DEUTSCH, Study Director (1984-1987)

SYMPOSIUM PARTICIPANTS

THOMAS B. SHERIDAN (Chair), Department of Engineering and Applied Psychology, Massachusetts Institute of Technology
DAVID L. AKIN, Assistant Professor of Aeronautics and Astronautics, Massachusetts Institute of Technology
ANTAL K. BEJCZY, Telerobot Research Program, Jet Propulsion Laboratory, Pasadena, California
BRUCE C. BUCHANAN, Knowledge Systems Laboratory, Stanford University
RAYMOND S. COLLADAY, Office of Aeronautics and Space Technology, National Aeronautics and Space Administration
KAREN S. COOK, Department of Sociology, University of Washington
RANDALL DAVIS, Associate Professor, Sloan School of Management, Massachusetts Institute of Technology
BARUCH FISCHHOFF, Department of Engineering and Public Policy, Carnegie-Mellon University
OSCAR GRUSKY, Department of Sociology, University of California, Los Angeles
PHILIP J. HAYES, Natural Language Systems, Carnegie Group, Inc., and Carnegie-Mellon University
WILLIAM C. HOWELL, Department of Psychology, Rice University
H. ANDREW MICHENER, Department of Sociology, University of Wisconsin
THOMAS M. MITCHELL, Computer Science Department, Carnegie-Mellon University
ALLEN NEWELL, Computer Science Department, Carnegie-Mellon University
RAYMOND S. NICKERSON, Bolt Beranek and Newman Laboratories, Inc., Cambridge, Massachusetts
JUDITH REITMAN OLSON, Graduate School of Business Administration, University of Michigan
PETER G. POLSON, Computer Laboratory for Instruction in Psychological Research, University of Colorado
WILLIAM H. STARBUCK, Department of Creative Management, New York University
LAWRENCE STARK, Telerobotics Unit, University of California, Berkeley
ROBERT C. WILLIGES, Department of Industrial Engineering and Operations Research, Virginia Polytechnic Institute and State University
HARRY L. WOLBERS, Manned & EVA Systems, Space Station Program, McDonnell Douglas Astronautics Company, Huntington Beach, California

CONTENTS

SESSION IV: COMPUTER-AIDED MONITORING AND DECISION MAKING

SESSION V: TELEPRESENCE AND SUPERVISORY CONTROL

SESSION VI: SOCIAL FACTORS IN PRODUCTIVITY AND PERFORMANCE

SESSION VII: THE HUMAN ROLE IN SPACE SYSTEMS

CLOSING REMARKS

FOREWORD

The Committee on Human Factors was established in October 1980 by the Commission on Behavioral and Social Sciences and Education of the National Research Council. It is sponsored by the Office of Naval Research, the Air Force Office of Scientific Research, the Army Research Institute for the Behavioral and Social Sciences, the National Aeronautics and Space Administration, and the National Science Foundation.

The principal objectives of the committee are to provide new perspectives on theoretical and methodological issues, identify basic research needed to expand and strengthen the scientific basis of human factors, and to attract scientists both inside and outside the field to perform needed research. The goal of the committee is to provide a solid foundation of research on which effective human factors practices can build.

In order for the committee to perform its role effectively, it draws on experts from a wide range of scientific and engineering disciplines. The committee includes specialists in the fields of psychology, engineering, biomechanics, cognitive sciences, machine intelligence, computer sciences, sociology, and human factors engineering. Participants in the working groups, workshops, and symposia organized by the committee represent additional disciplines. All of these disciplines contribute to the basic data, theory, and methods required to improve the scientific basis of human factors.

PREFACE

A steering group formed by the Committee on Human Factors was charged to identify the types of human factors research that, if funded and begun immediately, would be likely to produce results applicable to the evolutionary design of a National Aeronautics and Space Administration national space station to be launched in the 1990s. The steering group was instructed to consider human factors research relevant to such future space systems as the space station, lunar bases, and possibly interplanetary travel. The symposium, which was planned by the steering group and is reported in these proceedings, did indeed yield information applicable to future space systems. In addition, it provided information and offered insights of potential interest to many other civilian and military endeavors. It was our hope that this potential for transfer would occur.

I would like to thank the participants in this project for their time, effort, and contributions to the symposium. Individual authors accept primary responsibility for each paper and this authorship is acknowledged at the beginning of each paper. Steering group members deliberated, reviewed, and contributed to improvements in the content of each paper. I am especially grateful to them for their generous contribution of time both before and after the symposium.

The steering group, and the other principals in the production of this symposium, received a great deal of guidance and assistance from NASA personnel. On behalf of us all, I would like to thank Melvin Montemerlo and Michael McGreevy of the Office of Aeronautics and Space Technology, Richard Carlisle and Bryant Cramer of the Space Station

Office, and Owen Garriott, astronaut, for their extensive summaries of the space station planning activities during the initial October 1985 steering group briefings. Special thanks are also due to Jesse Moore, the director of the Johnson Space Center, Joseph Loftus, assistant director, David Nagel from the Ames Research Center, and the many NASA personnel who participated in the briefings of the steering committee held at the Johnson Space Center.

Finally, thanks are due to the people who have worked behind the scenes to ensure that the symposium was conducted, and the proceedings prepared, in an organized and timely manner. Appreciation is extended to Stanley Deutsch, study director at the time of the symposium, for his contributions to its planning; to Dana Kruser, project coordinator, for her efforts in the organization and execution of the symposium and assistance in the editing of this report; to Elizabeth Neilsen, research assistant, for her managerial and logistic support; to Beverly Huey, who also provided logistic support; to Christine McShane, of the Commission staff, for editorial support; to Margaret Cheng, who provided secretarial assistance in preparation for the symposium; to Marian Holtzthum, for secretarial assistance in preparing this document for review; and to Martha Seijas, for preparing the document for publication. I express my sincere thanks to each of these individuals for their significant contributions.

Thomas B. Sheridan, Chair

Committee on Human Factors

SYMPOSIUM SUMMARY

> We can follow our dreams to distant stars, living and working in space for peaceful economic and scientific gain. Tonight, I am directing NASA to develop a permanently manned Space Station and to do it within a decade.
> President Ronald Reagan, State of the Union Message, January 5, 1984.

In response to this presidential mandate, the National Aeronautics and Space Administration (NASA) is planning to launch a national space station in the early 1990s. To implement this commitment, and in concurrence with a congressional mandate, NASA is focusing serious attention on the use of automation and robotics in future space systems.

There is a tendency, particularly in the public sector, to view the emergence of new computer capabilities and automation and robotic technologies as a basis for replacing humans in space and thereby avoiding tragedies such as those of the Apollo 7 and the Challenger. However, it is unlikely that artificial intelligence comparable to human intelligence will be available to replace humans during the last part of the twentieth Century and the early part of the twenty-first. Therefore, people and automated systems will work together in space for the foreseeable future.

NASA is planning new research programs aimed at acquiring a better understanding of how computers, automation, and robotics can be made to work in partnership with people in complex, long-duration space system missions. These programs will address important questions concerning the relationship between what are called intelligent systems and the people who will use them as astronauts inside a space vehicle and in extravehicular activities, as scientists and technicians in space and on the ground, and as controllers on the ground.

Space offers significant challenges for the exploration and demonstration of human-computer-robot cooperation. Recognizing the size, complexity, and importance of this challenge, the Aeronautics and Space Technology Office approached the Committee on Human Factors for assistance. The specific question posed was "What research is, or should be, going on now that might produce new technologies that could, or should be, integrated into the space station after its initial operating capacity has been established?"

The committee responded to NASA's question by proposing to assemble a group of eminent scientists to address this issue and to present its views to the research community by means of a symposium on human factors research needs in advanced space station design.

Development of the Symposium

The Committee on Human Factors initially formed a small Steering Group composed of six researchers representing a broad range of relevant disciplines (i.e., human factors, artificial intelligence, expert systems, decision science, robotics and telepresence, and social science and space system design). The steering group was introduced to the task at hand through briefings from various NASA headquarters offices, including the Office of Aeronautics and Space Technology and the Space Station Office. Based on the information gathered during these briefings, the steering group then developed the following list of symposium topics and questions for consideration by prospective speakers.

- <u>System Productivity/People and Machines</u>
  - How can human performance and productivity be defined?
  - How can system productivity be measured and evaluated?
- <u>Expert Systems and Robotics and Their Use</u>
  - What are the requirements for reliability?
  - How can people, expert systems, and robots form an effective partnership?

- <u>Language and Displays for Human-Computer Communication</u>
  - How much structure does a computer language need?
  - What types of displays are most effective?
- <u>Telepresence and Supervisory Control</u>
  - What are the relative merits of various telepresence displays? (e.g., touch or stereopsis)
  - What can be done to increase the precision of control for remote manipulators?
- <u>Computer-Aided Monitoring and Decision Making</u>
  - What types of routine operations could be automated?
  - How will people use these types of aids?
- <u>Social Factors</u>
  - What factors affect group productivity and performance?
  - What are the potential effects of increased crew diversity with respect to such variables as gender, professional training, and interest differences?
- <u>Human Role in Space</u>
  - How should system functions be allocated in manned space systems?
  - Who or what instrumentality should take ultimate responsibility for system performance and safety, a human or a computer?

The general framework for the symposium was planned as follows. Each topic area would constitute a different session.

prepared especially for the symposium and would be followed by a formalcommentary on the papers by a preassigned discussant and would conclude with an open discussion. Members of the audience would be active participants and would be selected with this in mind.

The steering group identified and recruited three experts in each topic area: two authors and a designated discussant. The session on system productivity was an exception, having one author and one discussant. Before the symposium, all the prospective authors and discussants were invited to visit the Lyndon Johnson Space Center for briefings and discussions with key personnel involved in manned space flight research and development. Speakers and advisors were present from NASA headquarters, the Johnson Space Center, the Ames Research Center, and the Jet Propulsion Laboratory.

Following the extensive overview of NASA research efforts aimed at the space station effort provided by NASA personnel, symposium authors and discussants began preparing materials for the symposium. Individuals involved in each session worked together using an iterative peer review and revision approach in writing the papers and the formal commentary on them that was to be included in the symposium proceedings. Each group took responsibility for the completeness and technical accuracy of the material representing its area of expertise. Prior to the symposium, authors and discussants received a complete set of papers and commentary for each of the sessions.

The symposium was held at the National Academy of Sciences on January 29-30, 1987. Following the symposium, authors were asked to revise their papers and to suggest revisions to papers written by others based on the information and insights gained during the symposium.

The steering group did not consider its mandate to encompass the task of developing specific recommendations for research to NASA. The symposium presentations and commentary serve that purpose. However, the closing remarks of the keynote speaker and the chair, which appear at the end of these proceedings, stand as their personal interpretation of what was said that was the most important.

Symposium Abstracts

This section summarizes the contents of each of the symposium papers and provides the interested reader with an overview of the symposium program.

System Productivity: People and Machines

Productivity in the Space Station (Raymond S. Nickerson) The concept of productivity, while elusive, has been an important one in economics and engineering psychology and is frequently encountered in discussions of the space program and of the space station in means and how it has been assessed in earth environments. Several variables that have been shown to affect it are identified. Factors that are likely to have an impact on productivity in space are discussed, with emphasis on a variety of stressors that may be expected to characterize the space station environment. The paper ends with a set of recommendations for research.

Expert Systems and Their Use

AI Systems in the Space Station (Thomas M. Mitchell) Among the technologies that will help shape life in the space station, artificial intelligence (AI) seems certain to play a major role. The striking complexity of the station, its life support systems, and the manufacturing and scientific apparatus it will house require that a good share of its supervision, maintenance, and control be done by computer. At the same time, the need for intelligent communication and shared responsibility between such computer programs and space station residents poses a serious challenge to present interfaces between people and machines. Hence, the potential and need for contributions from AI to the space station effort are great.

This paper suggests areas in which support for new AI research might be expected to produce a significant impact on future space station technology. The paper focuses on two areas of particular significance to the space effort: (1) the use of knowledge-based systems for monitoring and controlling the space station and (2) issues related to sharing and transferring responsibility between computers and space station residents.

Expert Systems: Applications in Space (Bruce C. Buchanan) The technology of artificial intelligence (AI), specifically expert systems, is reviewed to examine what capabilities exist and what research needs to be conducted to facilitate the integration of humans and AI technology in future space systems. An expert system is defined as a flexible, symbolic reasoning program that uses heuristics to manipulate symbolic data in order to generate plausible answers to questions. Four goals are identified for expert systems: (1) performance (at a standard comparable to the best specialists); (2) reasoning (as opposed to straight "number crunching"); (3) understandability (the ability to explain why an answer is plausible and how it was generated); and (4) flexibility (the ability to deal with novel situations). Methodological techniques for achieving these goals are discussed, including modularity (keeping domain knowledge separate from decision rules, and independent clusters of domain knowledge separate from one another) and uniformity of language and constructs (both internally between segments of the program, and externally between the program and the intended users). The problems of collecting, representing, storing, maintaining, and manipulating domain knowledge are reviewed. Buchanan concludes that existing expert system technology is adequate for some problems but can be improved to use the very large knowledge bases required by a system as complex as the space station.

Language and Displays for Human-Computer Communication

Change in Human-Computer Interfaces on the Space Station (Philip J. Hayes) The planned longevity of the space station will require modularity in its design to allow components to be changed and updated as independently of one another as possible. This paper explores the issue of modularity in the design of human-computer interfaces for the space station. The need for modularity centers on the rapid rate of expansion in the kinds and combinations of modalities (typing, graphics, pointing, speech, etc.) available for human-computer interaction, and on the techniques available to effect their implementation and interaction. The paper assesses the appropriateness of current and forthcoming modalities according to task, user, and space station environment. A secondary factor that makes change in human-computer interfaces inevitable for the space station is the development of intelligent interfaces. The paper discusses methods of achieving intelligence in interfaces and in what circumstances it is desirable. The question of how to achieve the necessary changes in human-computer interfaces is considered, focusing on methods of obtaining a clean separation between the interface and the underlying space station system application. User interface management systems and interaction interface development environments are also addressed. The paper concludes with a set of research recommendations covering both research into new interface technology and methods for dealing with the consequent need for change in interfaces.

Cognitive Factors in the Design and Development of Software in the Space Station (Peter G. Polson) The paper describes major problems in the design of human-computer interfaces for systems on the space station and shows how systematic application of empirical and theoretical results and methodologies from cognitive psychology and cognitive science can lead to the development of interfaces that reduce training cost and enhance space station crew productivity. The paper focuses on four issues: (1) transfer of user skills; (2) comprehension of complex visual displays; (3) human-computer problem solving; and (4) management of the development of usable systems. Four solutions to the problems are proposed: (1) use of information processing models of tasks in the design process; (2) allocation of adequate resources to user-interface development; (3) use of user interface management systems; and (4) use of existing expertise in NASA.

Computer-Aided Monitoring and Decision Making

Robustness and Transparency in Intelligent Systems (Randall Davis) Building and operating a manned space station will give rise to problems of enormous complexity in an environment that is both hostile and unfamiliar. The complexity of the station and the novelty of the environment preclude the creation of an exhaustive list of contingency procedures. Unforeseen events will inevitably occur, requiring real-time interpretation, diagnosis, and response.

The paper reviews the failure of a fuel cell during the second space shuttle mission in order to give an example of the kind of unanticipated event that can occur and examines the varieties of knowledge and engineering reasoning required to deal with such an event. Davis considers what might be required to have a computer assist in this task by giving it an understanding of "how something works". Some nonsolutions to the problem are discussed to demonstrate why existing technology is insufficient, and several research themes are then explored. The nature and character of engineering models are considered and it is suggested that their creation, selection, and simplification are key issues in the sort of understanding that should be created. Recalling the difficulties involved in the capture of Solar Max, the paper argues for the necessity of complete design capture and speculates about what it would take to create a design capture system so effective that it would be was almost unthinkable to create or modify a design without it. The paper also considers what can be done at the design stage to create models that are easier to use and more effective; that is, how to design in such a fashion that interpretation, diagnosis, and response are made less complex processes.

Decision Making--Aided and Unaided (Baruch Fischhoff) There are few aspects of space station design and operation that do not involve some decision making, whether it be choosing critical pieces of equipment, choosing to trust automated systems, choosing where to look first for the source of an apparent anomaly, or choosing the range of conditions for pre-mission testing. Knowing how people intuitively make such decisions provides a basis for determining where they need help, in the form of automated decision aids, specialized training, or designs that are robust in the face of fallible decision making. Although it has much in common with decision making in other contexts, space station decision making presents some special demands. These include: (1) the need to create a shared model of the space station and its support systems, which will coordinate the widely distributed decision makers capable of affecting its performance; (2) the need to make decisions with imperfect systems, whose current status and future behavior are incompletely understood; (3) the need to make novel decisions, responding to nonroutine situations. The human factors research needs in each of these areas are identified, using as a point of departure the literature of behavioral decision theory. Meeting these demands will require the sort of programmatic research effort that has distinguished NASA in the past.

Telepresence and Supervisory Control

Teleoperation, Telepresence, and Telerobotics (Thomas B. Sheridan)

The problems of integrating humans and automated or robotic systems in space environments are discussed, beginning with brief definitions of key terms like teleoperation, telepresence, telerobotics, and supervisory control. The early development of teleoperators is summarized, from the crude mechanical earth-moving and construction equipment available prior to 1945, to the industrial robots, equipped with primitive computer vision, wrist force sensing, and "teach pendant" control boxes that were in use by the early 1980s. The current status of teleoperator development is evaluated, and multifingered manipulators, touch sensing, and depth perception are cited as areas in which promising research is occurring. A need is identified for a formal theory of manipulation to guide the development of human-machine integrated sensory-motor control systems. Research needs are identified in the following areas: (1) telesensing (including resolved force, touch, kinesthesis, proprioception, and proximity); (2) teleactuating (including multi-degree-of-freedom end effectors, two-arm interaction, and multiperson cooperative control of teleoperators); (3) human-computer interaction in a computer-aided environment (including simulation, planning/decision-aiding, and command/communication/control). It is concluded that research in the areas discussed is critical for the development of teleoperator/telerobotic capabilities, which will permit the best relative use of both human and machine resources in future space systems.

Telerobotics for the Evolving Space Station (Lawrence Stark) In this paper, telerobotics is used to mean remote control of robots by a human operator using supervisory and some direct control. By robot is meant a manipulator/mobility device with visual or other senses. This is an important area for the evolving NASA space station. The paper suggests that triplicate or three way planning should be employed. It is important to carry out research to accomplish tasks: (1) with people alone, if possible, such as in extra-vehicular activities; (2) with autonomous robots (AR); and (3) with telerobotics. By comparing and contrasting the research necessary to carry out these three approaches, present problems may be clarified.

The paper describes an experimental telerobotics simulation suitable for studying human operator performance. Simple manipulator pick-and-place and tracking tasks allowed quantitative comparison of a number of calligraphic display viewing conditions. The Ames-Berkeley enhanced perspective display was utilized in conjunction with an experimental helmet mounted display system. A number of control modes could be compared in this telerobotics simulation, including displacement, rate, and acceleratory control using position and force joysticks. Communication delay was introduced to study its effect on performance.

The paper suggests that the impetus and support for telerobotics research technology should come from NASA and from private industry and that such research could also be conducted, with support from NASA, in university laboratories.

Social Factors in Productivity and Performance

<u>Social Stress, Computer-Mediated Communication Systems, and Human Productivity in Space Stations (Karen S. Cook)</u> The paper has two distinct but related foci. First, it considers the issue of stress and reviews the social psychological literature relating stress to individual and group functioning. Primary attention is focused on the link between stress and group productivity. The paper

identifies promising lines of research in the social sciences and poses issues that might be of particular interest to NASA for future research. Second, the paper considers a broad class of problems that arise from the fact that life aloft requires, almost exclusively, mediated communication systems. This section of the paper addresses the psychological and social aspects of mediated communication (primarily, computer-mediated communication systems) and its impact on individual and group performance or productivity. The concluding section of the paper proposes a critical set of research needs that NASA might take as recommendations for programmatic research. These complement research currently being supported by NASA's Human Factors Division. Emphasis is placed on what are termed critical social contingencies, namely, those psychological and sociological aspects of life as envisioned on space stations that, if not managed well organizationally, could create major problems for crew productivity and viability in space.

Control, Conflict, and Crisis Management in the Space Station's Social System (H. Andrew Michener) The paper discusses two social systems: (1) the space station social system in the year 1993 and (2) the space station social system as it may have evolved by the year 2000. Because neither of these social systems exists today, they cannot be investigated by empirical techniques; thus, the discussion in this paper is necessarily theoretical and conjectural. It is proposed that the year 2000 social system, in contrast with the 1993 system, will be larger in size and more differentiated in composition, will make greater use of on-board computerization (artificial intelligence), and will pursue different goals and subgoals. These changes will, in turn, create a year 2000 social system that is more complex, more differentiated into subgroups, and more decentralized with regard to decision making than the year 1993 system. It is suggested that several consequences will follow from increases in complexity, differentiation, and decentralization. Specifically, it is likely that: (1) the supervisory-control system on board the space station will shift from a hierarchical form to a heterarchical form; (2) the potential for, and severity of, interpersonal conflict will be greater; and (3) the logistics of responding to crises will be different. Each of these points is discussed in detail. The paper closes with suggestions regarding research that might usefully be conducted today in anticipation of these changes.

The Human Role in Space Systems

<u>The Roles of Humans and Machines in Space (David L. Akin)</u> The fundamental requirements for any self-contained device performing a useful function in space are identified as follows: (1) sensation (the ability to detect objects); (2) computation (the ability to formulate a plan of action); (3) manipulation (the ability to interact with, and to alter, the environment); (4) locomotion (the ability to maneuver within the environment); (5) support (power, cooling, etc.). The past and present roles of human and mechanical systems in fulfilling these functions in space activities are reviewed, with emphasis on the special contributions of people to

the performance of space systems. The need to take an earthlike environment into space in order to accommodate humans is also discussed, including the constraints of atmosphere, consumables, volume, work cycles, and gravity. It is concluded that there will continue to be necessary and sufficient roles for both humans and machines in space systems for the foreseeable future. Research needs are identified in the following areas: (1) development of a meaningful data base on human and machine capabilities and limitations in space environments; (2) identification of appropriate roles for humans and machines in space systems; (3) development of appropriate metrics of human and machine performance; and (4) an assessment of anthropocentrism (the tendency to design autonomous machines based on a human model).

Sharing Cognitive Tasks Between People and Computers in Space Systems (William H. Starbuck) The differences between people and computers are persistent and profound. Although computers' capabilities have been developing rapidly, computer simulation of human thought has had little success. However, the differences between people and computers suggest that combinations of the two can achieve results beyond the capabilities of each alone. For that reason, NASA should devote research to improving the interactions and synergies between people and computers.

Nearly all the research on human-computer interaction has focused on people who lacked thorough training and who had little experience with computers. Since most of these findings may not extrapolate to the well-trained and experienced operators of space systems, there is need for studies of such users. Five research topics seem especially interesting and important: (1) fostering trust between people and expert systems; (2) creating useful workloads; (3) anticipating human errors; (4) developing effective interface languages; and (5) using meaningful interface metaphors. Inherent in these topics is an implication that NASA should develop a user interface management system that will recognize the needs of different users, allow different users to express their personal preferences, and protect users' individuality. The paper concludes that to improve the quality of designs and to improve users' acceptance of designs, experienced astronauts and controllers should participate in the designing of interfaces and systems.

DRAFT
NOT FOR DISTRIBUTION, ATTRIBUTION OR QUOTATION

INTRODUCTORY REMARKS

THOMAS B. SHERIDAN, CHAIRMAN

Welcome to the Symposium on Human Factors in Automated and Robotic Space Systems. I will start by saying a few words about why we're all here. A bit over a year ago, actually before the Challenger accident, Melvin Montemerlo, the Manager of the Human Factors Research Program and Co-Manager of the Automation and Robotics Program for the Office of Aeronautics and Space Technology in NASA Headquarters, requested the Committee on Human Factors of the National Research Council to consider the needs for human factors research in evolutionary manned space stations. Mel asked the committee to look at future manned space systems beyond the Initial Operating Configuration (IOC); looking ahead into the late 1990's and beyond. (I might mention that Mel is on sabbatical leave and Mike McGreevy is currently managing the programs.)

It was clear to us that any new research started now could not have much effect on the design of the IOC, so we knew we had to speculate for a period beyond this first space station. It was also clear to us, as we thought about it, that if a single issue could be considered to have the most effect on human factors in the space program, it would be the computer. And while much of the public, even the Congress, and even some in NASA management, have come to think in terms of the astronaut versus the computer and automation and robotics, I believe the science and technology community and many in NASA know better. It's really the astronaut, or human beings, working together with the computer and automation and robotics in close cooperation, that will result in the greatest mission success. But simply to say that and to have it really happen, are, of course, not the same. We have a long way to go to piece it all together.

So we were asked to think about this major issue and to organize a symposium, composed of experts who, in our judgement, represented the most critical areas of human-machine interaction, even though we could not cover all of the major aspects of human factors. The committee decided that it would be most effective if it concentrated on human factors issues in relation to computers, automation, robotics, and the roles of people in the space stations of the future. A reason for selecting the symposium format was the opportunity that it would afford an exchange with other people in the scientific community (including NASA) and other organizations who might make cogent contributions to the discourse.

Let me identify the people who worked hard with the Symposium Steering Group to put this symposium together, the staff of the Committee on Human Factors: Dr. Stanley Deutsch, the Study Director for the committee; Dana Kruser, a consultant to the committee, who is largely responsible for having all of the symposium papers ready on time; Elizabeth Neilsen, the committee's staff assistant, whose support on the logistics was invaluable; and Beverly Huey, also a consultant, who helped us to meet our schedule in myriad ways. They will all be available during the meeting if you have any needs.

We ask you to listen to our thoughts, and possibly some irreverent comments about the space program and the research that's been done or should be done, and to participate in the discussion. One reason that the proceedings are available out at this time is so that we can capture your ideas and include them in the proceedings of the meeting.

I thank you for participating and I hope we can make this an interactive meeting.

Now, I want to introduce Dr. Raymond S. Colladay, the Associate NASA Administrator for the Office of Aeronautics and Space Technology, to say a few words about the NASA organization. I will then ask Dr. David A. Goslin, the Executive Director of the Commission on Behavioral and Social Sciences and Education (CBASSE), to say a few words about the National Academy of Sciences and the National Research Council. The Committee on Human Factors is located organizationally within CBASSE.

So, first, Ray Colladay.

DRAFT

NOT FOR DISTRIBUTION, ATTRIBUTION OR QUOTATION

SPACE STATION SYMPOSIUM: OPENING REMARKS BY RAY COLLADAY

I'm delighted to see that in spite of the snow here in Washington, there is such a good turnout. I was talking to Stan Deutsch before the meeting and he told me that attendance had to be restricted so that the group would be small and intimate to encourage good interchange and dialog. I'm pleased with that because it provides a focus on a subject that is extremely important to NASA. And I'm further pleased by the fact that Human Factors is being considered at this symposium in the context of Automated and Robotic systems, because that's precisely that way we should look at that subject. This reflects what NASA is trying to do to bring those disciplines together.

I think that when you look into the subjects which you are addressing in this symposium, you're going to see a discrepancy between our goals and our current capabiility, specifically in the NASA program. Your feedback in the discussions and in the proceedings of this meeting will be very important to us in planning the program and in trying to get our capability on track with our expectations and our vision. We have great plans for extending human presence in space. The space station is only the first step in that vision, which is taking shape right now as we contemplate lunar bases, expeditions to Mars, and other missions beyond the space station.

It is my pleasure to welcome you to this symposium on Human Factors in Automated and Robotic Space Systems, and I'd like to thank the National Research Council's Committee on Human Factors for their efforts in conducting this symposium, and for their valuable contributions over the years to NASA's Aeronautics and Space Human Factors research programs. The committee has helped us to formulate and develop the kinds of programs we need in this area.

The subject of this symposium is timely indeed. Yesterday was the first anniversary of the Challenger accident, a day of rededication to excellence in memory of the Mission 51-L Challenger astronauts. It was a day when, as a nation, we rededicated ourselves to the excellence that characterizes America. For our part, we at NASA are developing a clear vision of the future in space and are currently refining our research and technology development plans to ensure the health, safety, and productivity of humans in space throughout the coming decades. Although it was only formalized as a research discipline about five years ago, our Space Human Factors Program is built upon a long history of aeronautical human factors research, and extensive agency experience in life sciences research and manned space flight.

Something else is happening in the NASA program which pleases me, and that is the start of a new building for human performance research for the space program at our Ames Research Center in California. I intend this building to be the first leg of a major facility that combines human performance and automation research. We are, in fact, putting a building in place to reflect exactly the kind of merger of those disciplines that this symposium is addressing. We'll call it the Human Performance and Automation Laboratory. This will pull those disciplines together in a very realistic way, and will get researchers working in the laboratory in computer science, artificial intelligence, automation, and human factors.

I look forward to the results of this symposium. I think it will be exteremely helpful to us. We welcome this opportunity to interact with you and I wish you luck in the proceedings and the discussions that follow. Thank you very much.

DRAFT

NOT FOR DISTRIBUTION, ATTRIBUTION OR QUOTATION

KEYNOTE

HUMAN FACTORS RESEARCH FOR THE NASA SPACE STATION

Allen Newell

Carnegie-Mellon University

Pittsburgh, Pennsylvania

[pp. 7-15 DO NOT EXIST]

PRECEDING PAGE BLANK NOT FILMED

CONTENTS

DRAFT

NOT FOR DISTRIBUTION, ATTRIBUTION OR QUOTATION

KEYNOTE ADDRESS

HUMAN FACTORS RESEARCH FOR THE NASA SPACE STATION

Symposia are held for many reasons. This one is to do a task. I hope you are intellectually stimulated by what you hear and will take away some new knowledge that you do not already possess. I also hope the symposiats have enhanced their own knowledge by their studies and are gratified by the chance to advance their views from this earth-bound, but otherwise splendid, platform. But neither of these has much to do with the actual reason for this symposium. We are here to help shape the research that NASA will perform on how humans interact with the technology of the space station.

In a nutshell, and to state what I hope is already shared knowledge among us, operating in space poses extreme challenges. It is a hostile, aversive, constraining and unforgiving environment. Our intent as a species to make such operations successful and to continuously extend their scope in complexity, duration, and usefulness is epitomized in NASA, and other space agencies around the world. It takes its concrete form by the posing of specific projects, each more daunting than the last, but (skillfully we hope) set just within the bounds of the reachable. For us today that project is the space station, a project with an initial development phase prior to launch of a decade and a total lifetime of several more decades. Such projects force us to not only use the best available technology and science, but to extend them substantially. For us today, the question is what research is most needed that could have important payoff for the space station. It is not possible, of course, to encompass in a single effort all the technologies and sciences that support the space station. Thus, we focus on how humans and technology can cooperate to carry out the operations of the space station.

We will speak today almost entirely of the space station. That is proper, because we need projects to give as much form as possible to a future which is almost agonizably open. But, such far-future projects are emblematic of entire technological futures. Thus, behind the space station is to be seen an entire spectrum of future space systems, replete with automated and robotic devices, while also being a habitat for humans. Indeed, the space station is itself not a single envisioned system, but a projected series that stretches out in time and evolves in significant ways. Only occasionally will we have need to distinguish even between such relatively concrete visions as the IOC and SSOC. The research talked about here in the context of the space station is what we see as necessary to this entire technological future.

This is our task. Its success can be measured by the influence of this symposium on how humans and technology actually work together. Do the crews that run the space station, both on the ground and in space, have an easier, safer, more productive time than would otherwise have happened? We are only one player in the hundreds of individuals, groups and organizations that affect what goes into the space station, and a highly transient one at that. Our only leverage is the cogency of the ideas we put forth. Still, we fail if nothing down stream is different because of what we say here today. It is not enough to have an effect, it must be the right kind of effect at the right place and time. Furthermore, the effect depends not only on NASA decisions about its research program, but also on the quality of the research that is thereby enabled, and whether its results transfer into the operational space station -- a notoriously tenuous conjunctive chain. Still, though we talk here today of possible research, we hope for operational results.

My task, right now, is to get us launched -- to set the stage and provide the context for the papers you will hear over the next two days. I will only take a few minutes to do this. But some overview will help us to keep on track throughout the meeting.

THE SPACE STATION

Let's start with the space station itself -- although I am hardly the one to do so, with an audience that contains many with active responsibilities for it. Still, even I know enough to start with the obligatory picture (Figure 1). This is of course a fantasy, composed from the minds of many persons and living only there and derivatively in the minds of receptive audiences. In accordance with its fantasy character, it changes continually -- if not daily, at least monthly. The planners harden the fantasy with physical mockups that can be walked in and gawked at. That helps, but the time constants to realization are still of the order of half-decades.

This way of talking about the space station may induce a sense of fragility. That could be a good thing, if it brings with it an increased sense of commitment to making it happen. However, my actual objective is to induce a sense that much can change in the space station before it takes its place in the sky and, indeed, after it does. If we are to consider launching research in 1987 and expect it to have operational impact, then the time scale of that operational world must be sufficiently long and its character sufficiently malleable.

Planning -- even research planning -- must have some grip on reality. Thus, we need to focus on the hard constraints on the space station -- the ones that appear to hold no matter what, and on which we can build securely. Table 1 presents three handfuls -- already more than can be assimilated in an introduction. These constraints are what strike a technically observant human-factors specialist immediately upon hearing a briefing on the station. They are the constraints that shape the roles that humans must play and the tasks they must perform to make the space station function. What makes them unyielding is the limited state of our space technology, the primary goals set for the station, and the necessity of acquiring certain experiences as stepping stones to future technological frontiers. No matter how technology changes we must pass this way to move forward - not, of course, with the exact particularities of the space station we will build, but through something with the general characteristics listed in Table 1.

Many familiar things follow from this: the general strangeness of the weightless world and its frustrations; the isolation of the station group, coupled with the lack of privacy and the extent to which members are locked in; the public work-oriented, regimented world; the complete dependence on the efforts of others; the stress of continually living close to fatal errors. By and large, humans respond adaptively to all these conditions. Still, humans in space must spend their psychic resources to cope with these conditions, rather than spend it in other more productive ways.

One striking thing is how saturated with technology the life of the station will be. This is completely true of those stationed aboard, but is almost as true of those aground for their workaday world, although they get to go home to the grass each evening.

Another striking thing is that the residence time-scale is long enough so that many functions have to be accommodated that can be avoided in shorter flights. The station appears to be a microcosm of life - so many activities must occur that one can find any problem or task one looks for, or at least a close analog. Now, in fact, this is not quite so. Many functions, such as raising a family, becoming educated, moving to a new home, and planning retirement, do not show up at time scales even as long as months. And to those concerned with the man-machine system in the modern fighter plane, where the focus is on actions in the subsecond range, the station will appear downright leisurely. That the space station occupies a middle range in the total timescale of human action is a significant simplification -- as we will discover when we have to plan permanent space or lunar stations. But even so, from the perspective of a human factors analyst, the space station has moved a long ways toward total living and not just temporarily occupied workspace. Along with that has come an almost un-enumerable collection of tasks that humans must perform, and the need for designing the artificial environment in which to perform them.

Still the tasks must be enumerated. One of the great liabilities of technological environments is that they don't take care of themselves -- not yet and for some time to come. The tasks to be performed in the station and between ground and station must be enumerated and explicitly planned for. What we fail to enumerate here below is in parlous state up above. There will always be true stories about the novel activities of intelligent astronauts, solving life-critical problems or having fun in ways we could not predict. Bless them for that. But let no one argue back from that blessed fact to the need for less preparation. And preparation implies explicit task description and enumeration. NASA, of course, has gone to great lengths to do this. Table 2 provides some reminders of what those enumerations cover.

The left hand column simply lists the various subsystems involved, so one gets some notion of diversity. With respect to each of these there are many actual tasks to be performed. To enumerate them is to descend into the technological gritty of each type of system. But various types of activities that go into these tasks can be identified, which is what the right hand column shows. These generic activities come in indefinite variety as well, in terms of what must actually be accomplished, with what initial knowledge, and against what constraints. Finally, I have put across the bottom what is perhaps the most important factor, namely, that the time scale over which these tasks endure stretches from less than a second to about four months -- seven powers of ten. Each task in its individuality fits into this time-stretch at some point. But every duration contains tasks of every type.

There are two points and one conclusion to be made from all this. First, I would impress upon you that there are an almost unimaginable variety of tasks, which contain almost any combination of task demands one cares to contemplate. Second, the vast majority of these tasks are to be accomplished by some combination of humans and technology. To be sure, at the top ultimately there is a pure human, if only a congressman; and at the bottom there is a pure machine, if only a pushbutton making an electrical contact. It follows that we can consider today only a selection of all the problems. We will of course seek for research that is generic in its character and that will impact large classes of these tasks. But much that is important will not even be mentioned.

THE TECHNOLOGY OF INTERACTION

The classical situation of human factors has been that an industrial or military organization develops some machine to do some task. The human-operator aspects of controlling this machine and of being trained to do so are dealt with in due course. In the best of cases, this occurs early enough to permit modest alteration of the engineering of the interface. But in the main, the technology of the machine is autonomous and fixed.

With the advances in artificial intelligence and computer science in general, and in computer interfaces in particular, the situation is changing -- and changing in several ways simultaneously. First, the machines are becoming more complex, yet capable of more autonomy and intelligence at the same time. Second, the interfaces themselves are becoming more intelligent so that they can aid the user and operate cooperatively with him. Third, all interfaces are becoming alike in their utilization of a common hardware and software technology. Finally -- and of a different order entirely -- the technology on which all this is based in itself undergoing rapid evolution, so that all the features just mentioned are not new fixities that can be depended upon, but are themselves on the move. All of these current truths have double force for the space station, which is located a long ways in the future. Let us focus on each of them in turn.

Machines are controllable arrangements of matter and energy that do things to the physical world. (Thus, tools are machines.) The ability to be controlled is of their essence, for it is what changes them, as micro parts of the world, from a thing that can be taken advantage of (as to drink from a brooklet happened upon) to a thing that can be used at will (as to turn on a faucet whenever thirsty). So machines bring with them the problem of the human-machine interface, and necessarily those interfaces are dynamic and continue throughout the duration of use.

As machines become more capable, through the rational foresight of their designers and the skill of their builders, the tasks that machines can do without human intervention increase. Although the real measure is in the total range of useful tasks they can accomplish with acceptable reliability, an appropriate indicator is the length of time machines can go without interaction with humans. With this increased scope comes inevitably the problem of who should do a task, the human or the machine. Formally, this is exactly the same as the problem of whether this human or that should do a task, or whether this machine or that. However, because of the category difference, the human-machine question is taken to have a more profound character and it becomes the focus of scientific attention. It is a surrogate, of course, for our need to understand the advancing capabilities of machines.

That question is finally about to change its form radically. The advances in computers and computation have now been driving exponentially for forty years. All parts of that advance are significant for us today, in part because they all interrelate. The driver of it all, we always say, is the cost/performance of the computing devices and the level of their integration. But by this time that itself depends on software design systems with quality graphics. So it is all one ball of wax. Nevertheless, the parts where the advances touch us the most here today is in robotics, artificial intelligence and the technology of the human-computer interface. Through these, the amount of intelligence that can be incorporated into machines is now reaching the place where the problem of assignment of functions to men or to machines no longer holds any charm. The question must be phrased -- How can humans and technologies cooperate to attain a set of system-level goals.

The situation at the interface between the human and some machines provides a good example of the increase in the capabilities that are available, with a concomitant increase in the complexity for those of us who design and understand these systems. As machines increase in capability, interfacing to them becomes a complex task in its own right and requires substantial knowledge about what is required to communicate knowledge back and forth -- languages, protocols, communication over intermediate links, the status and location of the communicants, and on into the night. The solution is to have special agents that have this knowledge or know how to acquire it, in short, intelligent interface agents. But such agents imply that knowledge about how things work will be distributed -- of what good are such agents unless they relieve other parts of the system of the responsibility for having certain knowledge and skills? But this reinforces the point made earlier that it no longer makes any sense to cast the problem of how humans work with technology in exclusive terms of who controls whom. Rather, it must be in how agents embodying distributed sources of knowledge cooperate.

One more point about the technology and I am done with it. If NASA had to settle for the level of intelligence in current robotic and expert systems, this symposium would have a very different character. We have, of course, come a long way in computer science in the last forty years and this is plainly evident in existing robotic and intelligent systems. But the changes are proceeding very rapidly and substantially more capabilities can be expected to be available in another five years or in five years more again. This introduces uncertainty into our proceedings, for we must not only talk of what new research might bring, but must place this against a background that will increase in possibilities no matter what NASA does. But this same motion also adds to the sense of excitement of the new powers that are possible in the space station. The space station, by being a project measured in terms of decades, both suffers and benefits in the extreme from this motion of technology.

RESEARCH OBJECTIVES

Given the picture just sketched of tasks and technology, the question of the day is what research should be done. The substantive answers to that question are the responsibility of the speakers of this symposium. I would only ask you to keep three general considerations in mind.

First, the research topics raised here range widely -- from artificial intelligence, to the human-computer interface, to telerobotics, to issues of social organization. These are not just a congeries, brought together to obtain coverage. They are all facets of how humans are to interact with the primary technology of the space station, and what technologies are involved in that interaction. A research program needs to address all these aspects in some coherent way, and not treat them as separate questions.

Second, we have had to sample - to focus on some issues and to neglect others. But the research program needs to consider the full range of phenomena. It is in research plans, and the study efforts that support them, that one engages in the compulsive attempts to taxonomize the domains and worry seriously about coverage and missing elements. A symposium is to make clear the fruitfulness of research areas and to show that there are exciting research questions. Attempts at completeness and evenhandedness would only dull the senses.

Third, with more glibness than honesty, I have just shifted an immense burden from the symposium speakers to the symposium participants -- or at least some of them. For, of course, the domain of research is so broad that coverage is a chimera. That is especially true if one thinks of research as devoted to getting answers to specific questions about a specifically configured space station. Such answers must be obtained -- that is what engineering requires. And in the present context it is human engineering and even organizational engineering. A research program that is in effect a systematic and planned program of human and organizational engineering, with the resources to do some background studies, cannot possibly provide the coverage that is necessary. Thus, the research program must be aimed at discovering conceptual, theoretical and technical tools that will permit the human and organizational engineering of the space station to proceed with greater efficiency and accuracy. Only if a research program advances the theoretical state of the art, including therein systematic organizations of data that permit answering a multitude of questions, will it serve NASA in the decades it takes to achieve the space station.

THE INSTITUTIONAL CONTEXT

Thus far, like a good cobbler, I have stuck to my last, discussing the substantive issues. But it is important to say something about the institutional context in which the symposium occurs. Blessedly, I need not take my point of departure from the current spasm of reflection, critique and renewal that has been the fallout of the Challenger disaster. For our timescale is too long for that to count as more than a transient. At least that is true if NASA can continue in its planful ways, which it shows every sign of doing. Thus, in setting out the institutional context I will not talk about the microstructure of command and timing that will, in fact, have the lion's share of responsibility for whether any trace of this symposium's efforts survives these two days. Instead, I will point to larger entities.

Let us start with NASA. It is, of course, the primary player. It is its space station, after all. Its primary view of human factors considerations has got to be simply as an ingredient to make the space station better -- as a factor of production, in the economist's sense. That view leads inevitably to working backwards from specific questions about the space station to specific studies to answer them. After all, in the logic of planful organizations: To get X, set up a plan for getting X. Furthermore, the cogency of a plan can only be apparent if it explicitly and recognizably puts down each step, from what is available initially to the obtaining of X. This leads to a thoroughly applied effort and one characterized by short-range goals with tight loops of justification. Such a logic is certainly appropriate in part -- after all, if NASA doesn't do the studies to deliver the answers it needs on the nitty-gritty of the space station, who else will? But the timescale of the space station is long enough so that other attitudes are appropriate as well. NASA can change the available science enough to make a difference to the space station itself. And to do that the research must be launched on a broader and freer path, letting it pick its way among the interesting questions of today to the different questions of tomorrow. The issue for NASA then is whether it will rise above the immediate applied questions of human factors -- to which the safety and productivity of the astronauts will force attendance in any event -- to the faith that major gains for the space station can be attained from supporting basic long-term research.

Each of us has our own stories of where such long range research by an institution has made immense differences to the downstream operation of that institution. Not being a NASA insider, my stories of that ilk do not come from NASA. But even to an outsider it is apparent that there must be a whole book full of such stories. After all, space science is an almost new science, even though, as always with science, it has a whole tangle of historical roots in early rocketry, astronomy, and more. And space science is practically a creature of NASA, so NASA must know all about the gains from bringing a new science along.

Nevertheless, it may be worth recounting briefly one of my own stories. This is DARPA's creation of the field of artificial intelligence and expert-systems technology. DARPA did not start artificial intelligence, that occurred in the mid 1950s. But only a few years afterwards, in the early 1960s, DARPA began its open support of that part of computer science. It did so in an essentially free spirit and mixed with the many other things it was also supporting, such as time sharing, graphics, multiprocessors (Illiac IV) and networking. The support was substantial, but was far from being the dominating item in the mix of DARPA programs. The important aspect, from the present point of view, is that DARPA started its support in 1962. By 1972, a mere decade later, the first expert systems had begun to emerge -- Dendral and Mycin. By 1982, only one more decade, the commercialization of expert systems had begun. Today, five years later, though still a green and nascent technology, it has become the property of us all. It has become integral to much of DOD's own future and is now integral to our discussions here. But for almost all the first twenty years, DARPA was essentially the only support for artificial intelligence[1]. Thus, we see that an agency can bring into existence wholly new techniques and ideas for its own use downstream. It cannot usually be done in less than a decade. But in timescales that are commensurate with the space station, such things are possible. And their payoff is incalculable.

The second major player is the collection of scientists and engineers who will conduct the research. This is not a homogenous group. Most immediately, the scientific cadres within NASA concerned with human factors and artificial intelligence are to be distinguished from the scientists in the universities and research organizations across the country. Each clearly plays a different role, although, in the style of the times, strong attempts exist to weld these into a more continuous community, with the establishment of places such as the NASA Research Institute at Stanford University.

The more important inhomogeneity is among the social institutions we call professions and disciplines. Focus narrowly on the human-science issues concerning the space station, and ignore totally the half-hundred natural-science and engineering disciplines concerned with the physical structures in the space effort. However, a gaggle of disciplines are still gathered around this seemingly narrow focus. Alphabetically, they are: artificial intelligence, cognitive psychology, computer science, human factors, industrial engineering, organization theory, robotics, social psychology, sociology. I have no doubt overlooked some, but all these, at least, are represented among the speakers of this symposium. The inhomogeneity here arises from two sources. First, the issues of the space station involve multiple technologies, and the relevant human phenomena are so diverse that they necessarily make contact with different human sciences. But second, multiple human-science disciplines focus on the same phenomena, but do so from different perspectives. In particular, the emergence of the computer as a mass phenomena has raised the problem of human-computer interaction to prominence. At least four disciplines - artificial intelligence, cognitive psychology, computer science (mostly graphics and interface programming) and human factors -- are currently engaged in forming an interdiscipline called human-computer interaction (HCI). The effort is currently focused on the individual in interaction with the computer via a system of interaction mechanisms (displays, keyboards, pointers, etc.). It is acknowledging, though only gradually, social and communicative dimensions. The conceptual and disciplinary turbulence involved in all this is both part of the inhomogeneity of the current scene and revelatory of it. HCI is only one part of the human-related issue of the space station, though a significant one.

The NASA situation that we discuss at this symposium provides an opportunity for these disciplines. They can, of course, treat the NASA problems as if they were just another collection of interesting situations in which to ply their investigatory trade. Our nation -- blessedly, once again -- is extraordinarily pluralistic. Thus, NASA research contracts and grants can be taken as providing additional micro-research opportunities in a larger mix. This is one view and an important one.

But the NASA situation provides a larger opportunity, or at least it does if NASA chooses to make that opportunity available. The space station provides a unique focus for the development of the science of how humans interact in a technology-saturated environment. By reaching so far ahead of the degree of saturation in the rest of current society, it offers a chance to study a world well ahead of its time. It is a unique opportunity in this historical moment, although it will become less so as the saturation of the rest of the world proceeds.

It is important to realize that in applied sciences technological foci have an immense influence on the character of the science. One has only to think of the influence on human factors of its being nurtured by the aircraft industry, while being relatively ignored by other industries. Thus, NASA has a fleeting opportunity to bend the twig of HCI to a long-term concentration on aspects especially relevant to NASA's interests.

ENVOI

The ingredients of the symposium has now been assembled before your very eyes -- the space station; the tasks of human-technology interaction; the technologies that are both the object of that interaction and the means to make it work; the orientation towards the research that needs to be done; and the institutional setting within which this symposium must make its contribution. Let us now move to the substantive papers.

NOTES

1. My friends from ONR and from NIH-supported AIM (AI in medicine) may be a little annoyed at so sweeping a claim; yet it remains close to true.

TABLE 1 The Hard Constraints that Apply to the Space Station

1. Long lifetime of the station (decades).
2. Medium term crew residence on board (months).
3. Small group of residents aloft (less than ten, to begin with).
4. Large group of operators (non-residents) aground (hundreds).
5. Very small amounts of resources available per resident.
6. Very small amounts of space available per resident.
7. Infrequent physical communication (months).
8. Continuous, but limited-bandwidth communication.
9. Time delay of station communication of .5 to 2 secs.
10. Modest time constants of action (minutes to hours).
11. Weightlessness.
12. Continuous, high task load.
13. Continuous high threat-level of many potential errors.
14. Continuous public exposure.
15. Completely artificial environment.

TABLE 2 Types of On-Station Tasks for the Space Station

Subsystems	Generic functions
Power	Power handling
Guidance & navigation	Checkout
Communication & tracking	Mechanical actuation
Data handling	Data handling and communication
Propulsion	Monitoring/control
Environmental control and life support	Computation, decision and planning
Thermal	Fault diagnosis and handling
Structures/mechanisms	Sensing
Crew systems	
Payloads (experiments, manufacturing, observations)	

|--------|--------|--------|--------|--------|--------|--------|--------|--------|

.1 sec 1 sec 10 sec 100 sec 10^3 sec 10^4 sec 10^5 sec 10^6 sec 10^7 se

1 sec 1 min 1 hour 1 day 1 month

NEWELL
KEYNOTE

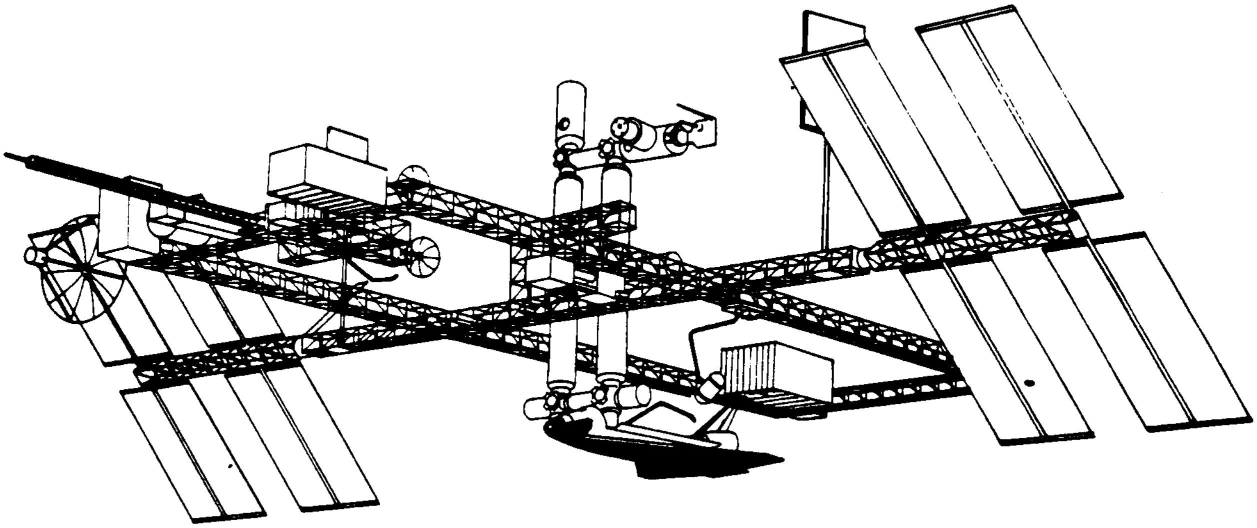

FIGURE 1 Artist's construction of an evolutionary space station.

SESSION 1:

SYSTEM PRODUCTIVITY: PEOPLE & MACHINES

Paper: Raymond Nickerson, Bolt Beranek and Newman Laboratories, Inc.

Discussant: Robert Williges, Virginia Polytechnic Institute & State University

DRAFT

NOT FOR DISTRIBUTION, ATTRIBUTION OR QUOTATION

PRODUCTIVITY IN THE SPACE STATION

Raymond S. Nickerson

BBN Laboratories, Inc.

Cambridge, Massachusetts

DRAFT
NOT FOR DISTRIBUTION, ATTRIBUTION OR QUOTATION

TABLE OF CONTENTS

[P.42 DOES NOT EXIST]

PRECEDING PAGE BLANK NOT FILMED

DRAFT

NOT FOR DISTRIBUTION ATTRIBUTION OR QUOTATION

INTRODUCTION

What is productivity? How do we measure it, predict it and control it on earth? To what extent can that knowledge be extrapolated to a space context? What do we not know about productivity on earth that might be found out -- and is worth finding out -- through research? How might the expected findings be applied to space? How should the research be directed to ensure its applicability to space? Are there important questions about productivity in space that earth-based research is not likely to help answer?

I wish I could promise to answer these questions here. Unhappily, I cannot. These are the kinds of questions that I have had in mind, however, in preparing this paper. In what follows I will focus first on the notion of productivity and on how it has been measured and manipulated in earth environments, and then turn to the question of productivity in space, or more specifically, the Space Station. The paper ends with a set of recommendations for research.

WHAT IS PRODUCTIVITY?

Productivity is an elusive concept. It seems straightforward enough when one begins to consider it. It is easy to think about the productivity of chickens or dairy cows in terms of eggs laid or milk produced per unit time; here we are dealing with output in a very literal sense. And it does not tax one's imagination to think about comparing the output of the one producer with that of the other. To do this we need a way to describe eggs and milk quantitatively in the same terms, which is not difficult. Since eggs and milk are valued as foodstuffs, we could describe them both with respect to their nutritional ingredients. But quantifying productivity only in terms of output is not very useful from an economic point of view, and as it relates to chickens and cows as producers it would be grossly unfair to the chickens; we must also take into account how much chickens and cows consume in order to produce a given amount of nutritive capital by means of eggs and milk respectively. And to round out the picture we must factor into the equation not only what the producers eat, but other resources upon which their continuing production depends. To do all this we may find it convenient, since not all the factors that must be considered are nutritional, to quantify everything in monetary terms. But this gives us no serious problem. The situation is still fairly simple conceptually: chickens and cows produce foodstuffs that can be given a monetary value, and to do so they consume resources that have a monetary cost; productivity can be thought of in terms of the value of what is produced and the cost of producing it. This all makes intuitive sense.

When one tries to apply the same type of thinking to human productivity, one has no trouble as long as the human activity involved is analogous to laying eggs and giving milk, in the sense of producing tangible goods that can be used to satisfy basic human needs, and consuming resources in the process of doing so. The picture gets less clear quickly, however, when what is produced is not so tangible -- perhaps not even readily identifiable -- and not easily quantified in monetary terms. How does one measure the productivity, for example, of the teacher, the scientist, the poet, the philosopher, the salesperson, the physician, the corporate executive, the athlete, the entertainer - or the astronaut?

Lack of definitional precision has seldom been a great deterrent to the use of words, and "productivity" is no exception in this regard. It is a popular word in economics, and like "truth" and "beauty," connotes something much to be desired, whatever it means. Within the literature pertaining to space exploration, one finds references to increases in the productivity of spacecraft crews resulting from changes in displays, control procedures or other variables, but seldom is it clear exactly what this means. The word is also seen throughout the human factors literature more generally; although Muckler (1982) has commented that the unconstrained way in which it is used here makes its meaning difficult to discern in this context. In practice, productivity is often used more or less as a synonym for performance; if performance improves, by nearly any criterion, productivity is said to go up; if performance degrades, productivity is said to go down.

Sometimes the word <u>is</u> given a precise quantitative meaning by virtue of the variables that are involved in its measurement. Indices of productivity are typically expressed as a ratio where the numerator is some measure of output (what is produced or the value of same), and the denominator is some measure of input (what is used up in the production process or the cost of same). What constitutes input and output, and how they are quantified, differs considerably from case to case, however; and changes in productivity indices over time can sometimes be difficult to interpret (Baily, 1986). Moreover, often the word is used as though it were intended to connote a quantitative entity, but there is no clue as to what the input and output variables are or how they could be measured.

Two concepts that are closely related to productivity are those of <u>production</u> and <u>efficiency</u>. Productivity implies production, or more specifically, product and producer. Productivity is an attribute of a producer; and a producer, by definition, is one who produces something. What is produced may be tangible (paper clips, a household appliance, an airplane) or intangible (an educational service, entertainment). A producer may be a person, a person-machine system, a team, a factory, an industry, an economic sector (agriculture), a nation, the world.

But although productivity and production are closely related concepts they are not the same. As we have noted, productivity is usually expressed as a ratio of some measure of output or product value to some measure of input or production cost, and the goal, in most cases, is to make this ratio as high as possible. Production usually refers only to output quantity. Given these connotations, it is easy to imagine production increasing or decreasing independently of changes in productivity. If, for example, a manufacturer produced 10 percent more items in a given year than in the preceding year, but doing so required a 15 percent increase in the number of employees, we might say that production increased while the productivity of the employees declined.

The concept of efficiency, like that of productivity, relates output to the resources consumed in obtaining it. Efficiency has to do with getting the most out of given resources; the challenge is to organize a production process so as to minimize wasted effort. A process is said to be made more efficient when the unit costs of output are decreased or when the consumption of a fixed amount of resources yields a greater output than before.

Techniques for measuring the efficiency of assembly line workers were among the earliest contributions of engineering psychology to the manufacturing process and have been used extensively in the work place. These have typically involved analyzing production tasks into observable components. The development of task-analysis techniques has received considerable attention from human factors engineers (Van Cott and Kincaid, 1972; Woodson, 1981). Such techniques have been more readily applied to psychomotor tasks than to tasks that are primarily cognitive in nature or even those that have major cognitive components. Attention has been focused increasingly, however, on the problem of analyzing cognitively-demanding tasks, as an increasing percentage of the tasks performed by people in the work force are defined more by cognitive than by psychomotor demands.

We cannot hope to settle terminological issues here. Moreover, definitions are of limited utility when dealing with terms that are widely used, with a variety of connotations, within a field. For present purposes, productivity will be taken to be very close, but not quite identical, in meaning to efficiency. An entity (person, group, system) will be considered highly productive when it uses its resources to maximum advantage in accomplishing its goals. One can be efficient in the sense of not wasting resources simply by using those resources very sparingly, but that type of efficiency could be counterproductive if resources are husbanded to the point of precluding getting the task done. To be productive one has to use one's resources and use them well.

As a working definition of productivity I will use: effective and efficient use of resources in accomplishing a goal. The emphasis is on both effectiveness and efficiency. A productive system is one that gets the intended job done and does so with a minimum of wasted effort and resources. I do not mean to split hairs here in making a distinction between efficiency and productivity; if one's idea of efficiency incorporates effectiveness, then I see no objection to thinking of efficiency and productivity as more or less synonymous. Effort and resources can be wasted as a consequence of many factors, such as poor training, lack of motivation, mismanagement, faulty organization, misscheduling, and a host of others. Productivity will be said to increase when either more is accomplished with no increase in consumed resources or the same objectives are attained with a smaller expenditure of resources.

These are still somewhat imprecise notions, but not so imprecise as to be useless. In the Space Station context, as elsewhere, when modifications in design or operating procedures have big effects on productivity, there probably will be no difficulty in getting a consensus that productivity has really been improved. When tasks are performed more easily, more reliably, and with fewer costly errors, most interested observers will probably be willing to describe what has happened as an increase in productivity, and even if not, they are likely to agree that changes for the better have occurred. It seems to be generally assumed, if only tacitly, that anything that improves human performance (increases speed, accuracy, reliability) probably increases human productivity. This appears to me to be a reasonable assumption, and a very useful one. Frequently in this paper, the discussion focuses on variables that influence performance, the justification being the assumption that what affects performance for better or worse will affect productivity in a comparable way.

ASSESSING PRODUCTIVITY

It is helpful in the present context to distinguish between the problem of determining what the level of productivity is at any given time and that of determining whether productivity is changing, or has changed. One might assume that the second problem is more difficult than the first, inasmuch as a measure of change, or difference, is derived from the more fundamental measure of absolute value: to determine whether productivity is more or less this week than it was last, one simply takes the difference between this week's measure and last week's. But this is so only if one wishes to know the magnitude of the difference. If one is content to know only the direction of the difference, it may not be necessary to know the individual magnitudes, at least if the magnitude of the difference is relatively large. One does not have to know the precise weight of each of two objects to know which one weighs more, especially if the difference is sizeable.

Productivity as a Percentage of Capacity

Productivity is sometimes quantified in terms of performance relative to a maximum. When this is done, maximum output or performance is used as the standard against which to evaluate the actual output or performance, whether the performer is an individual, a system (say a factory), or an economy. Thus one might encounter the claim that the productivity of a given industry in a particular region is currently at about 70 percent, which would mean that that industry is operating at 70 percent of what, under certain assumptions, is the maximum possible. Economists often refer to how close to capacity factories and other manufacturing facilities are operating. The ability to specify how close to capacity some entity is operating presupposes a metric in terms of which to quantify the operation. Determining what constitutes maximum capacity can sometimes be a complicated and controversial process. Further, maximum must be understood as maximum within a particular context. The maximum output of a given factory, for example, could mean maximum obtainable with the present tooling, layout, manpower and stock; alternatively it could refer to what would be obtainable if one or more of these constraints on output were relieved.

As applied to individual human beings, capacity connotes the best (which often, but not always, equates to most) one can do in a given situation, the limit of human performance - or, more accurately, the limit of the individual performer. Conceptually, there are two ways to determine capacity in any given instance: one is to derive it from theoretical considerations; the other is to measure performance under ideal conditions. Neither works very well. While information theory once provided a basis for the hope of defining capacity theoretically, it proved to be a false hope, and psychologists have not yet found or developed an alternative that can do the job. Ideal conditions for performing a given task - which would have to include an optimally motivated performer - have proved also to be easy to conceptualize but difficult if not impossible to actualize.

Differential Productivity

Differential productivity in a business context is sometimes measured in terms of changes in the number of employees or amount of employee time required to get a fixed amount of work done, or conversely by changes in the amount of work accomplished by a fixed staff. Thus a retail company is said to have doubled the productivity of its bill collection departments when it managed, by computerizing its operation, to place the same number of calls with a 50% reduction in staff. And the productivity of an insurance company is described as increasing fivefold when the number of policies issued per employee per year increased by a factor of five (Bowen, 1986).

Studies of individual human productivity in specific job situations have often focused on the performance of individuals relative to the performance of other individuals on the same task. It is possible to say that A is more productive than B without saying anything very precise about how productive either individual is relative to a larger frame of reference. Measures of white-collar productivity typically do not yield absolute quantities, but do permit comparisons among similar organizations (Drucker, 1986).

In the Space Station program, attention will probably be focused primarily on <u>differential</u> productivity (the cost of attaining some production objective in space relative to that of obtaining it on earth; or the cost at one time relative to that at another). While it would be interesting to be able to relate productivity to some theoretical maximum in this context (e.g. by relating production to some measure of capacity), it is not clear how to do that. Fortunately, it is not necessary to be able to quantify maximum productivity in order to determine whether one is moving toward or away from it.

That is not to suggest that assessing differential productivity is likely to be an easy task. Several investigators have commented on the variability of measurements of productivity, especially those that relate to individual human productivity, and on the resulting need to make many measurements over a considerable period of time if reliable numbers are to be obtained (Muckler, 1982). It is especially difficult to measure productivity in intellectual tasks, inasmuch as methods for assessing cognitive performance are not well developed. When a person is staring out of his office window, it may be impossible to tell whether he is idly daydreaming or is engrossed in "productive" thought. And even if he were known to be daydreaming, it would not follow necessarily that that time was lost from a productivity point of view. One widely held view of problem-solving distinguishes an "incubation" period in the problem-solving process during which progress is made on a problem in spite of -- perhaps because of -- the fact that the individual is not consciously focusing on the problem to be solved -- and there are numerous examples of scientists and other thinkers reporting insights that have occurred when they were not actively engaged in working on the problem.

Whatever methods are developed for measuring productivity must take quality -- as well as quantity -- of output or work into account in some way. In manufacturing operations, product quality affects measures of productivity to the degree that items that fail to meet a preset standard become rejects. The importance of quality control in this sense is obvious and the difficulties that some industries (e.g. the manufacturing of computer microchips) have had are well known. This type of linkage between quality and quantity is a fairly gross one however. Differences in quality tend to be ignored so long as the quality is not sufficiently low to necessitate rejection. In nonmanufacturing activities the relationship between quality and quantity is even more tenuous, in spite of the fact that here one might expect qualitative differences in output to be both large and important. Quality will certainly be an important consideration in the Space Station context. The quality of the experiments that are done, for example, will be at least as important as the number.

Workload and Its Assessment

In a complex system the operation of which depends on functions performed by both people and machines and, especially, by people and machines in interaction, high productivity will require that workloads be at or near optimal level. Significant overload will reduce productivity through increases in the frequency of human error; significant underload will mean wasted resources at best and possibly direct negative impact on productivity resulting from boredom, inattentiveness or other difficulties arising from feelings of being underutilized or unimportant to the operation. Workload and its assessment will be important considerations, therefore, in efforts to understand, measure, or control productivity in space.

As in the case of efficiency, the workload carried by an individual is much easier to measure when the task is primarily physical than when it has major cognitive components. As Wierwille et al. (1985) point out, a major consequence of the increasing automation of modern systems is a shift in the role of the human operator away from manual control and toward monitoring and performance evaluation, and this has complicated considerably the problem of quantifying the operator's workload. How can we hope to determine how hard -- how close to capacity -- an individual is working when most of what he is doing is mental activity that is not directly observable?

The measurement of mental workload has been recognized by human factors researchers as a major challenge to the field and this recognition has stimulated considerable activity (Chiles and Alluissi, 1979; Eggemeier, 1980; Kalsbeek, 1968; Moray, 1979; Parks, 1979; Sheridan and Simpson, 1979; Singleton et al., 1971; Williges and Wierwille, 1979). Work in the area is still in the exploratory and formative stages, however, and there has not yet emerged a theory or even a widely agreed upon set of concepts and measurement procedures that are needed to provide a sense of stability and coherence.

An indication of the magnitude of the problem and of the current status of work on it is provided in the Proceedings of a NATO Conference on Mental Workload published in 1979. Johannsen (1979:3) opened the conference with the observation that "there exist too many conflicting ideas about the definition and measurement of workload", and expressed the hope that the conference would produce a consensus among participants on a definition and on a procedure for workload assessment. In his preface to the conference proceedings, Moray (1979:VIII), the organizer, acknowledged that these hopes were not realized, but noted that participants from various disciplines did come to "very similar conclusions about the validity, usefulness, and promise (or lack of each) for a wide variety of methods for approaching the assessment of workload in the human operator". It is unfortunate that the proceedings does not contain a summary of these conclusions. It does contain, however, a report from each of five participant groups, classified as experimental psychology, control engineering, mathematical modelling, physiological psychology and applications.

The experimental psychologists summarized their conclusions this way: "The concept [mental workload] reflects a genuine dimension or dimensions of human experience in daily work...it is a concept absolutely required for the adequate analysis and description of such tasks [tasks that are not necessarily physically demanding but that are experienced as exhausting and stressful nonetheless] and for predicting, at the design stage, the future performance of such [automatic and semi-automatic man-machine] systems... On the other hand the concept is at present very ill-defined with several probably distinct meanings... There is no satisfactory theory of 'mental workload'" (Johannsen et al., 1979:101). Johannsen et al stress the multidimensional nature of workload, and deny the appropriateness of trying to quantify it as a scalar variable. They specifically rule out the possibility of meaningfully comparing different tasks with respect to workload, except when the tasks are very similar in structure.

The conclusions drawn by the experimental psychologists in the NATO workshop clearly caution against any expectation that the problem of workload measurement will be resolved soon. They are equally clear, however, in supporting the view that workload is an essential concept if we are to understand the role of human beings in modern systems and design tasks that impose reasonable demands on their capabilities. It could prove to be an especially important concept in the context of the Space Station because of the unusual cognitive demands that that environment will represent. A detailed understanding of those demands -- insofar as possible in anticipation of the deployment of the station -- surely must be a primary objective of the human factors effort in this program.

One of the approaches that has been used to identify performance measures that are sensitive to workload has been to take a variety of candidate measures in situations in which workload is intentionally varied and see which of them vary with workload manipulation (Casali and Wierwille, 1983; Hicks and Wierwille, 1979; Wierwille and Connor, 1983; Wierwille et al., 1985). Much of this work has been done in flight simulators. Candidate measures that have been studied include opinion scales (subjects' ratings of the task in terms of specified descriptors), physiological measures (heart rate, respiration rate, pupil diameter, eye-blink frequency, eye-fixation fraction), measures of performance on secondary tasks (time estimation, tapping regularity), and measures of performance on the primary task. A limitation of this approach is that viable measures, at best, reflect differences in workload; they do not provide an indication of how hard or how close to capacity one is working in any particular case.

Many of the studies of pilot workload have made use of post flight questionnaires. Because this approach is heavily dependent on memory, Rehmann et al. (1983) explored the possibility of having subjects report how hard they are working periodically while performing a task. Workload judgements did change in this case with controlled changes in task difficulty, but this measurement technique has the disadvantage that it could interfere with the performance of the primary task, especially when the latter is very demanding.

The intrusiveness of the measurement process has been a major drawback of many approaches to workload assessment, and especially those that make use of a secondary task (Rolfe, 1971; for a summary of nearly 150 studies using secondary tasks see Ogden et al., 1979). One way to avoid the use of an intrusive task and also dependence on the subject's memory is to monitor physiological indicants of workload that can be obtained automatically. Isreal et al. (1980) have argued that some of the physiological measures that have been tried -- galvanic skin response, heart rate variability, and pupil diameter -- reflect changes in autonomic nervous system activity and so are sensitive to changes in emotional state independently of their origin. As a physiological measure that is more likely to be indicative unambiguously of changes in the cognitive demands of a task, they propose the event-related brain potential and, in particular, its late positive or p300 component. Wickens (1979) also has argued for the use of evoked potentials. Isreal et al. (1980) present data from one experiment supporting the idea that this measure does vary with task demands and that obtaining it need not interfere with the primary task. While it would be imprudent to conclude from these data that electro-physiological monitoring of workload will be effective in the Space Station, the possibility deserves further exploration.

Varying workload for experimental purposes is probably not feasible within the Space Station context, or at least the amount of this type of experimentation that can be done will probably be very limited. It will be essential to attempt to have workloads be as close to ideal as they can be made from the very beginning. Of course when evidence indicates that an initially established workload is not ideal, the workload should be changed in the indicated direction, and keeping track of such changes can provide some of the data that would have been obtained from controlled experimentation. The goal must be to minimize the need for such changes, however, which requires being able to predict the effects of different workloads from data obtained in earth environments.

DETERMINANTS OF PRODUCTIVITY

There seems to be a consensus among investigators that productivity is a function of many variables, and that attempts to affect it that focus on one or a small subset of those variables and ignore the others run the risk of doing more harm than good (Muckler, 1982; Sutermeister, 1976). Among the determinants of productivity that would have to be included in any extensive list are the following.

Human Capabilities and Limitations

A great deal of information has been compiled about human capabilities and limitations and is available in various engineering psychology handbooks. What is known in this regard clearly sets bounds on what human beings in general can be expected to do in specific task situations. Individual differences are also germane to the question of human productivity. People differ widely with respect to both physical and mental capabilities, and the productivity of individuals is bound to vary with the degree to which their individual capabilities match the demands of specific tasks. Aptitude testing and job screening and selection procedures are based on these assumptions.

Task Demands

Evidence supports the intuitively appealing idea that people work best when the demands upon them are neither too great nor too small. This is one form of the "inverted-U hypothesis" regarding the relationship between workload and performance, which holds that performance of a given task is optimal for a workload level that is intermediate between one that is excessively high and one that is so low as to promote boredom (McGrath, 1965; Welford, 1973, 1974). The detrimental effects of overloading are somewhat better documented than are those of underloading (Weiner, 1975; Weiner et al. 1984). The possibility that underloading can affect performance negatively takes on special significance, however, in the context of systems in which humans function primarily as supervisors of automated processes.

Motivation

One can hardly doubt that motivation affects performance. It is clear in particular that performance suffers when motivation is very low. What is less clear is how performance is affected when motivation becomes extremely high. Modest increases in motivation that is relatively low at the outset will almost certainly lead to improved performance, but what happens when motivation that is already very high is increased still further? Is there such a thing as trying too hard? Wanting too badly to succeed? Some investigators believe there is, and that when motivation is extremely high it has a debilitating effect. This is another form of the inverted-U hypothesis mentioned above; except that in this case the performance determinant of interest is motivation rather than task demands. It may be that the detrimental effects associated with motivation becoming too high are better attributed to anxiety over the possibility of failing; fear, especially when it becomes panic, undoubtedly can cause performance to deteriorate. According to this view, if motivation becomes arbitrarily high but is not accompanied by such fear, we would not necessarily expect performance to fall off. The distinction between very high motivation and fear of failure may be an important one in the Space Station context; it would be helpful to have a better understanding of the roles of these variables as determinants of productivity and performance.

Physiological State

Fatigue has long been recognized as a factor in reducing productivity in many settings (Simonson and Weiser, 1976). Indeed it has been defined operationally as a decrease in performance as a consequence of prolonged activity (Kalsbeek, 1971). Much of the research on this topic has focused on the problem of scheduling rest breaks in such a way as to minimize fatigue (Bechtold et al., 1984; Ganaro and Bechtold, 1985). The tasks involved in these studies have often been physically strenuous and the results are of limited applicability to tasks that are primarily cognitive in nature. Exceptions include studies of the performance of aircrews over extended periods (Cameron, 1971, 1973). A major question of relevance to productivity in the Space Station is how productivity might be affected by the various physiological effects that can be expected from prolonged living in the Space Station environment. Little is yet known about the physiological consequences of living in such environments for longer than a few weeks at a time.

Training

Performance, especially of complex tasks, obviously improves with training and practice. An aspect of the relationship between training and performance that is especially important relative to the Space Station context has to do with the obscuring of differences by ceiling effects. The fact that one has, through practice, gotten to the point of being able to perform a task without error is not compelling evidence that one has really mastered the task. The true test may come when that task must be performed under stress or in concert with competing demands on ones resources. To make the point another way, the fact that two people perform a given task equally well under accommodating conditions is not good evidence that they will perform it equally well under stress.

Capabilities and Limitations of Machines

Just as the capabilities and limitations of the humans in a complex system help determine the productivity of the system as a whole, so do the capabilities and limitations of the machines involved. Unlike the capabilities of human beings, those of the machines that are available for use in the Space Station can be expected to evolve even over the next few decades. Initial plans for the use of technology in the Station take this fact into account. Plans to use artificial intelligence, for example, explicitly note the unlikelihood that this technology will be used extensively for operational purposes during the initial years of the program. However, provision is being made for its incorporation as the technology matures to the point of being reliably applicable. We would expect that as machine capabilities are extended and improved, a major consequence would be increased productivity of the Space Station as a whole. Whether this proves to be the case and, if so, exactly how remain to be seen.

Person-Machine Function Allocation

An important determinant of system productivity, as distinct from both human productivity and machine productivity, must be the way in which system functions are assigned to people and to machines. Several methods for function allocation have been developed (for a review, see Price et al., 1982); but none of them is widely used by system designers (Montemerlo and Cron, 1982; Price, 1985). Investigators have argued that it is not realistic to expect it to be feasible to allocate function by formula anytime soon, if ever, because the problem is too complex and situation-dependent (Price and Pulliam, 1983). Allocations typically are made in an ad-hoc fashion on the basis of human judgment, aided perhaps by efforts of engineering psychologists, beginning with Fitts (1951), to distinguish between generic functions that machines do better than people and those that people do better than machines. While the number of functions that people can perform and machines cannot is likely to grow ever smaller with continuing advances in machine intelligence, it is likely to be some time before machines can match people in their ability to integrate information in so many forms from so many sources; to respond as effectively and adaptively to such a wide range of unanticipated situations; to make judgments of relevance, reliability and importance; to draw upon a large store of common sense, as well as technical, knowledge; and to follow imprecise instructions and work toward high-level goals. And if machines acquire such capabilities in time it does not follow that they should assume these functions in all cases. The question of what functions can be automated and that of what functions should be automated may have different answers. This fact has not received the attention it deserves. There may be reasons not to automate functions that are automatable from a technological point of view. These include reasons of cost effectiveness, human preference, and the need to maintain human skills at a high level in case they are needed in the event of system failure. One function that we can presumably assume will be a human one indefinitely is that of high-level goal-setting. Value judgments, including judgments of what goals are worth working toward, will hopefully remain the purview of human beings, no matter how clever the machines become. This probably means also, at least for the foreseeable future, retaining the role of deciding to what extent the behavior of the clever machines is consistent with those top level goals.

Design of Person-Machine Interfaces

In very complex systems like the Space Station, many functions are performed neither by people nor by machines independently, but by people and machines interactively. This being so, the adequacy of the designs of the interfaces through which information passes between the machines and their users will be a major determinant of productivity of the people, the machines, and the Space Station as a whole. The design challenge for the Space Station is complicated by the fact that the intent is to accommodate a large fraction of the anthropometric spectrum. It is here, in the design of interfaces, that human factors researchers and engineers are likely to have the greatest impact on productivity. A great deal has been learned about interface design as a consequence of human factors research in other contexts (Nickerson, 1986). A significant general conclusion to be drawn from this research is that designers' intuitions uninformed by human factors research are often wrong. A second similarly general conclusion is that small differences in interface design can often have very large effects. This area deserves a great deal of emphasis in the Space Station program.

Organizational Factors

Gunn (1982:115) has claimed that, in the case of manufacturing, the major opportunity for improved productivity is not to be realized by mechanizing the work of making or assembling products, but rather "in organizing, scheduling, and managing the total manufacturing enterprise. The most important contribution to the productivity of the factory offered by new data processing technology is its capability to link design, management, and manufacturing into a network of commonly available information". Gunn's emphasis on the importance of a single integrated information system, serving various needs of a manufacturing operation, applies with as much, if not greater, force to the Space Station context. Information will be the life blood of the Station. How the information that supports the various functions will be organized and accessed will be a critical aspect of the Station's design. Problems of organization, access, updating, protection, and representation abound. How these problems are addressed is certain to have implications for productivity, which is not to suggest that those implications will be easy to make explicit.

Scheduling Factors

Scheduling is a particularly important problem for any operation that involves numerous interdependent processes that proceed partly in series and partly concurrently. The problem is exacerbated by the fact that an unanticipated delay in the onset or completion of any given process may have implications for the timing of other processes. Small perturbations can ripple and grow into major problems producing inefficiencies at best and sometimes serious difficulties. Dynamic rescheduling of multi-process operations of any complexity usually requires computer involvement. Producing the scheduling algorithms, however, is still a human activity and one that requires a great deal of ingenuity, if major inefficiencies are to be avoided.

Social and Interpersonal Factors

The linkage between social or interpersonal factors and productivity may be indirect, but there can be no doubt of its importance.

Interpersonal difficulties among people who must work cooperatively as a group can seriously impair the smooth functioning of the group; conversely, when the members of the working group genuinely like each other and enjoy working together, there can be equally substantive positive influences. Interrelationships outside the working situation, and sudden changes in them, can also have profound effects. A new emotional relationship, illness or death of a loved one, an unresolved dispute with a friend or acquaintance are obvious cases in point. Such factors can affect performance not only through changes in morale or motivation, but also by diverting attention from the demands of one's job.

The above list of determinants of performance could easily be extended, but it is representative of factors that have been studied. Much is known about how these factors relate to performance and thus to productivity in earth environments. Much remains to be learned too, however, and while the themes may seem familiar, the new context of space gives the problems new dimensions. While all of these factors are likely to prove to be important in space, none represents a greater opportunity and need for research than those that involve the way people will relate to and interface with machines, especially in view of the rapidity with which the capabilities of the latter are changing.

Anticipated Functions and Uses

The Space Station is expected to serve a variety of functions. These include serving as a laboratory for scientific experimentation and data gathering, manufacturing and processing of materials (e.g., crystals and pharmaceuticals), servicing of satellite and other space vehicles, providing a staging platform for other space missions, and serving as a base for constructing large structures for use in space. The station is viewed as being important not only to scientific and commercial enterprises but to the further development of space technology. Eventually the station is expected to serve as an extraterrestrial control and service center for numerous unmanned satellites orbiting in a variety of inclinations and altitudes. Serving as a control and maintenance center would include deploying, retrieving, repairing, and reconfiguring other satellites or spacecraft (JSC,1979, NASA-ASEE, 1985). Considerable interest has also focused on the role the Space Station could play as a development and evaluation platform for automation, robotics, knowledge-based systems and other emerging technologies that make intensive use of computer-based resources.

Preliminary Design and Operation Considerations

The station is expected to evolve in at least two ways. As a physical plant it will increase in size and become more complex as modules are added and desirable modifications are identified. Operating procedures will also change as a consequence both of experience gained in operating it and of technological developments. In the interest of facilitating the evolution of the physical plant as new desiderata are identified, design plans call for modularity and expandability.

The living-working modules are an interconnected set of 4 pressurized cylinders, each of which measures 35-40 feet in length and 15 feet in diameter. The sizes of the modules are constrained by the requirement that at least the initial ones be prefabricated to fit in the cargo bay of the space shuttle. Two of these modules are to be living quarters and two are to be laboratories. Each living module will accommodate 6-8 people. A fifth module similar in design is called a logistics module and will be used for transporting equipment and supplies between earth and the station. Each of the modules is equipped with detachable units to facilitate reconfiguration, servicing and replacement.

Safety is, of course, a major concern. And this problem has the added dimension that mishaps that would have relatively minor consequences on earth could be disastrous in space. The possibility of fire in the spacecraft is a major worry for obvious reasons. This concern dictates many aspects of spacecraft design. Among the safety provisions that have been specified in preliminary design documents are: safe exit from any of the pressurized modules; isolatability of each module from the others; sufficient food, waste management, control and communications, and life support facilities in any three-module cluster to sustain crew and make rescue possible. Concern for safety also dictates that much of the training regimen focus on possible malfunctions.

In addition to the issue of safety, that of habitability is receiving considerable attention (Clearwater, 1985; Clearwater and Kasper, 1986). This issue becomes much more important for missions of extended periods than for those of a few days' duration (Wise 1985, 1986). The question is how to use color, texture, lighting, spatial arrangements, window placements, and other design features, within the constraints of other requirements, to make the various Space Station modules, and especially the living modules, pleasant places in which to spend long periods of time.

It is intended that the Space Station be as self-contained as possible. Consequently, much attention is being given to recycling of supplies, such as water, and to on-orbit maintenance and repair. Because the kind of constant and extensive ground control monitoring that has characterized short duration missions is not feasible for a permanent station, much attention is also being given to the objective of giving the station crew a high degree of autonomy and independence in its day-to-day operation. And because the intent is to make the station attractive to the private sector and useful for commercial ventures, the operating policies will have to take account of the desires of the station's clientele. There is a strong interest in assuring human productivity in the Space Station environment, which stems in part from the anticipated high cost of manned flight.

Uniqueness of the Space Station

Newell, in the preceding paper, has highlighted thirteen "hard constraints" that may be expected to hold independently of the specifics of the station's design. The list makes clear the enormous challenge the Space Station program represents. It also points up the fact that the uniqueness of the space station environment stems not so much from any given constraint or small subset of them, but from the set as a whole. For any given constraint, one can point to one or more other environments or situations with which we have some experience that shares it (e.g. nuclear submarines, submersible laboratories, off-shore oil platforms, polar excursions, scuba and deep-sea diving, incarceration -- prisoners of war -- and time spent at sea by shipwreck survivors). Some of these environments or situations share several of the constraints in Newell's list, but none of them shares the entire set. This is an important point. Suggestive evidence regarding the expected effects of some specific constraints in the Space Station may be found in the results of studies of other environments that share those constraints; and situations that have been studied include extended submarine patrols (Weybrew, 1961; 1963) and wintering-over parties in the Arctic and Antarctic (Gunderson, 1963, 1974; Gunderson and Nelson, 1963). But extrapolating what is known about the effects of any given constraint or even small subsets of them may overlook important effects of interactions. It is not prudent to assume, in the absence of supportive evidence, that the effects will simply add. It is easy to imagine conditions under which constraints that individually would have minor effects would, in combination, produce major ones.

Many of the constraints in Newell's list have implications for productivity, either directly or indirectly through, say, morale. Multi-month crew residences and infrequent physical communication outside the station, for example, could result in feelings of isolation, deprivation or boredom, or interpersonal tensions among the personnel. Limited resources and space could become uncomfortably restrictive in time. Weightlessness can produce nausea, headache, stuffiness and other physical discomforts, as well as spatial disorientation.

If challenged to extend Newell's list of constraints to incorporate other characteristics of the Space Station environment that are likely to be especially important from the point of view of productivity, my candidates for consideration would include the following:

- High degree of interactivity, and especially cognitive coupling, between crew and equipment.
- Computer mediation of control actions and displays.
- Criticality of information systems.
- Need for aiding or augmenting of human thinking for troubleshooting and decision making.
- Importance of human-machine interface designs.
- Need for continual concern for safety.
- Need for ability to deal with unanticipated contingencies.
- Shared responsibility of flight-control decisions between ground and flight crews.

- o Need for some operating procedures and principles to be negotiated with customers; in some cases, perhaps, while in orbit.
- o Heterogeneity of Space Station inhabitants (different languages, different cultures, different professions, different amounts of technical training and flight experience).
- o Importance of satisfying ways for inhabitants to spend free time.
- o Stress.

Each of these characteristics deserves attention as a variable that could have significant implications for productivity. Consider, for example, the second one. In the Space Station most of the control actions that are identified by humans will actually be effected by computers and most of the information provided to the human operators will be provided via computer-generated displays. Focusing only on displays, for the moment, it is easy to see how the ubiquitous computer mediation represents an important departure from more conventional displays. A major concern in the operation of any high performance vehicle is that of keeping the operator(s) aware of those aspects of the system's state that are critical to its operation. In conventional aircraft most indications of system state (altitude, bearing, airspeed, fuel reserve, etc.) are indicated by fixed displays each of which is dedicated to a particular indicant; when the pilot wants to check the plane's altitude, he looks at the altimeter, which is always in the same spot and displays always and only altitude: a little area of the cockpit is totally dedicated to the objective of keeping the pilot aware of how far off the ground he is. In the Space Station, most of the information that crew members receive will be delivered on computer driven displays that are used for more than one purpose. Display functions that were once implemented in hardware will now be implemented in software, and the type of information that is available in a specific spot on a control console will vary from time to time, depending on what piece of software is controlling the display device at the moment. This shift from hardware to software implementation of display functions has some implications for the problem of keeping crew members aware of system state.

Productivity in the Space Station

Productivity can have several connotations relative to the Space Station. It can refer to the impact of the Space Station program as a whole on the GNP or GWP. It can refer to the use of the Space Station by industry in production and manufacturing. It can refer to the performance of individual humans or person-machine complexes. Also, there may be a diversity of goals relating to the measurement and control of productivity in the Space Station. It may be desirable, for example, to measure the productivity of an individual, a person-machine system, a team, or an entire station over some specified period of time. One goal might be to achieve some targeted productivity on average over extended durations. Another might involve being able to achieve peak productivity for short periods when needed.

Impact on National or Worldwide Productivity

Considerable emphasis is being put on the potential commercial uses of the Space Station and the assumption that it will have beneficial long-range effects on the economy of the participating nations. The 1986 report of the National Commission on Space, Pioneering the Space Frontier, proposes that the space program have three mutually-supportive thrusts:

- Advancing our understanding of our planet, our Solar System, and the Universe;
- Exploring, prospecting, and settling the Solar System;
- Stimulating space enterprises for the direct benefit of the people on Earth (p. 5).

The third of these thrusts is directly relevant to the idea that the space program could have implications for national and international productivity.

Whether productivity gains will be realized will depend, of course, on whether the savings due to better quality control more than offset the cost of getting materials to and from space and any other increases resulting from conducting the operations in a space environment. To have a significant impact on national or international productivity will require a continuing operation of considerable size. The impact on certain industries could be significant relatively quickly, however, if the cost effectiveness of space-based manufacturing is conclusively demonstrated.

The space program could also affect productivity on earth in a variety of ways. Exploration of the earth's atmosphere and surface with photography (e.g. LANDSAT) and other sensors can produce information that can affect productivity by producing a better understanding of weather patterns, energy sources, climatic trends, and so on.

Industrial Productivity in Space

The combination of zero-G and vacuum in space is expected to facilitate production processes for which it is critically important to contron for convection forces or airborne impurities. Among the materials and products that are of interest in this context are "shaped crystals, semi-conductors, pharmaceuticals, biologicals, strategic materials, plastics, films, oils, alloys and mixtures, ultra pure metals, composites, glasses, membranes, metal foam, fibers, microspheres, ceramic/metal, and matrix materials" (NASA-ASE, 1985:9). A major industrial interest in space is the prospect of growing superpure crystals (e.g. gallium arsenide) for semiconductors in an environment free of convective turbulences. Interest in conducting such operations in space stems from the assumption that the quality of the products will be much easier to control (Chaudhari, 1986). It is expected to be possible to grow much larger crystals, for example, and to have a much smaller reject rate.

Individual Productivity in Space

Individual productivity -- the effectiveness and efficiency with which the individual participants in the Space Station program carry out their assignments -- is of special interest to the human factors community, inasmuch as the other types of productivity are contingent to no small degree on how well individuals function in their various roles. All of the determinants of productivity mentioned earlier in this chapter represent important considerations for the Space Station, as they do for any complex system. The following are also among the more significant issues relating to individual productivity that are very likely to arise in this context.

- Redundancy and backup: Many of the functions performed by the Space Station crew will be sufficiently important that provision will have to be made for backup in case an individual becomes incapacitated. The necessity to rely on backup capabilities could have implications for productivity, depending on the adequacy of the backup procedures and the extent to which reliance on them has a ripple effect to other functions.

- Use of aids, intelligent and otherwise: There will be a need in the Space Station to augment human cognitive abilities in various ways. Decision-making aids, troubleshooting aids, memory aids, will be needed in various contexts.

- Error recovery: It must be assumed that in a manned Space Station human errors will occur. The standard approach to minimize disastrous consequences arising from such errors is (1) to attempt to build in fail-safe procedures so as to make it difficult to commit the errors in the first place and (2) to buffer operator actions -- postponing their effects -- so that when an error is made, there is an opportunity to correct it. There is an obvious tradeoff here between safety and short-term productivity.

Fail-safe procedures and provisions for failure recovery are likely to slow operations down. In the long run, however, their costs may be more than offset by what they save if they prevent errors with serious consequences.

- Information accessibility: The operation of the Space Station is expected to be highly procedurized. While crew members may be assumed to have had extensive training in how to deal with various contingencies that may arise, it is not safe to assume that all the information they will ever need is stored in their heads. Availability of precisely the right information at specific moments could prove critical not only to productivity, but in some instances to safety or even survival. A recent report from a NASA sponsored workshop identifies a system that explains or assists in the use of other tools as perhaps the single most important tool from the standpoint of EVA autonomy and recommends the development of a real-time maintenance information retrieval system that could provide astronauts information on demand relating to EVA tasks as they are being performed (NASA-ASEE, 1985).

- Life-support systems: Although very great progress has been made in improving the design of space suits over the years of the space program, the suits currently in use for extra-vehicular activity still greatly restrict the wearer in various ways.

- Morale: Excepting complications arising from motion sickness, morale has not been a major problem affecting performance of crews in flight in the space program thus far. But the publicity surrounding the flights and the relative brevity of their durations have probably sufficed to keep the morale generally high. When people are in space for months at a time and the work becomes less of an adventure and more of a job, it will not be surprising if morale becomes an issue, and one that could affect productivity, from time to time.

In addressing these and related issues, it is useful to bear in mind that while the Space Station will differ from other environments in numerous ways, many of the issues that relate to productivity in this environment are of more general interest because of their relevance to earth environments as well. The question of how various types of information are best represented on computer-driven displays is a very general one, for example. And it takes on considerable practical significance in view of the fact that 40 to 50 percent of all American workers are expected to be using electronic terminal equipment on a daily basis by 1990 (Giuliano, 1982). Unquestionably designers of Space Station displays should benefit from the many ongoing efforts to package information more effectively for use in office, industrial, and other earthbound contexts; we expect also, however, that efforts to get the Space Station displays just right -- because being almost right may not be good enough in this context -- will yield knowledge about display design that will advance the state of the art in a general way.

The Evolving Role of Humans in Space

There has been and continues to be a debate about the advantages and disadvantages of a space program that includes manned spacecraft as opposed to one that does not. That debate will not be rehearsed here, beyond noting that opponents of a manned program have argued that having humans in space is unnecessary for many aspects of space exploration and providing for their safety delays the program and increases its costs (e.g. Van Allen, 1986a, b) whereas proponents of a manned program have presented a variety of arguments in favor of it, among them our inability to provide machines with some human abilities that are seen as critical, especially in responding to unanticipated events. Of particular relevance in the present context is the argument that has been made that the presence of humans in space will contribute positively to the productivity of the program as a whole. In this paper a manned program is taken as given. The problem then becomes that of designing a Space Station environment and operating procedures that will insure both the safety of the crew and the success of its missions.

The human's role in space has expanded and diversified over the life of the space program (Loftus et al., 1982). In the earliest flights the role was primarily that of passenger in a highly automated or ground-controlled vehicle. As experience was gained and the flights became more ambitious the crews took on more of the responsibility of piloting the spacecraft. Still later, the crew's role was enlarged to include functions unrelated to piloting, such as performing scientific experiments and repairing malfunctioning equipment.

Specific tasks that have been performed by crew members include monitoring of the various spacecraft subsystems (guidance and control, propulsion, environmental control, and life support); guidance and control during rendezvous and docking; landing and taking-off of lunar module (about 10,000 key strokes are required to complete all elements of a lunar landing mission, according to Loftus et al., 1982); assembly, maintenance and repair (especially of scientific instruments); aiming of scientific instruments and conducting of experiments; monitoring of data quality; and housekeeping.

The ability of the crew to perform maintenance and repair operations and to handle unexpected subsystem failures of various types has been demonstrated in several missions, including Gemini, Apollo 13, Skylab, and Spacelab (Garriott, 1974; Garriott et al., 1984). Especially in the Skylab and Spacelab programs crewmen on numerous occasions were able to repair malfunctioning equipment that was essential to the planned experiments. As Garriott et al. (1984) have suggested, the importance of the function should be reflected in the training of the crew designed to familiarize them with the equipment and how to repair it.

The ways in which the crews participated in the research activities of the Skylab and Spacelab programs have also been reviewed by Garriott (1974) and Garriott et al. (1984). An important idea emerging from these reviews is the following one. To the extent that crew members are to act in behalf of scientific investigators located on the ground, this function may go more satisfactorily if there has been more opportunity for the crew members to work with the scientists prior to the space mission.

As the human's role has expanded and diversified, the need for specialized capabilities and talents on space crews has increased, and consequently the crew members are less and less interchangeable. In the Shuttle program, specialization is recognized explicitly in the terminology, which distinguishes between mission specialists and payload specialists. In prolonged flights, like those anticipated for the Space Station, there will be an even greater need for certain specialized skills than has been the case heretofore. It may be necessary, for practical reasons, to have specialists who are also able to function effectively as crew members outside their area of specialty.

An important problem in planning the crew requirements for the Space Station will be that of assuring that collectively the crew has all the knowledge and skills that success and safety will require. What is difficult about this task is specifying the knowledge and ingenuity that will be required to deal with whatever unexpected contingencies arise. While it is not possible, of course, to anticipate everything that could happen, one step that can be taken in this direction is to attempt to identify the major <u>types</u> of problems that could arise (e.g. problems in the station's electrical system, medical problems among the crew, etc.) and to make sure that there is expertise within the crew to deal with problems in those areas.

Some of the activities the Space Station's crew will perform will take place outside the spacecraft. Such extravehicular activities (EVAs) may include the changing of focal planes and other servicing of the Hubble Space Telescope (HST), the Gamma Ray Observatory (GRO), the Advanced X-Ray Astronomy Facility (AXAF), and the Shuttle Infrared Telescope Facility (SIRTIF). (For a tabular summary of extravehicular activity on spaceflights through the Skylab III, see Loftus et al., 1982.) A major component of the cost of EVA activity stems from the large amount of time required to make the transition from the environment inside a pressurized space capsule to that outside it (Howard et al., 1982). Pressure inside the Space Station is 14.7 psi; that in the pressurized suit is 4.3 psi (King and Rouen, 1982). Because of the magnitude of this difference it is necessary for astronauts, in order to avoid the bends, to clear out the nitrogen in their body tissues by breathing pure oxygen for 3 or 4 hours before exiting the spacecraft. This procedure could be eliminated if the pressure maintained by the suit were above approximately half that maintained inside the cabin; thus immediate exit upon donning a space suit would be possible if either suits were designed to maintain 8 psi and cabins were kept at 14.7 psi as they currently are or cabin pressure was maintained at about 8 psi and suits at 4.3 psi, as they now are (NASA-ASEE, 1985).

Extravehicular activity represents a special challenge with respect to productivity for a variety of reasons, including the following:

- o Severe constraints on mobility and dexterity imposed by the pressurized space suit.
- o Limited visibility due in part to restrictions on head movements from the helmet and space suit.

- o Greatly reduced tactile feedback to the hands because of pressurized gloves.
- o Free floating or tethered (and easily tangled) tools.
- o Limited voice communication with in-station crew.
- o Problems associated with personal hygiene and comfort; most serious perhaps are the problems of defecation for males and defecation and urination for females, but the general problem surfaces in numerous other, perhaps less serious, guises as well: it is very difficult to scratch one's nose or any other itch in an EVA suit.
- o Problems of eating and drinking.

To the degree that the Space Station is an automated system that is monitored by human beings and dependent on manual override in case of subsystem malfunctions, it will pose the same kinds of challenge as other systems of this type. One such challenge is that of assuring that the human monitors are adequate to the task. The monitoring and controlling of dynamic systems are quite different tasks, and there is some evidence that people who have not had experience as manual controllers are less effective at detecting small changes in system dynamics than are those who have (Kessel and Wickens, 1983; Wickens and Kessel, 1979, 1980; Young, 1969). Another challenge relates to the dependence on human monitors for back up in case of system failure, and that is the problem of maintaining the human skills needed to perform complex functions that are very seldom performed under normal operating conditions. How does one keep crew members highly skilled at complex tasks that they seldom, if ever, have to perform? According to Jones et al., (1972), the most important functions aboard present spacecraft involve diagnosis and decision making, and retention of diagnostic and decision making skills represents our greatest gap in knowledge about task retention at the present time.

A major challenge for extended space missions, especially those involving long periods of time simply getting to a destination (e.g. interplanetary travel) will be to keep a crew and other inhabitants of the space vehicle occupied in meaningful ways when there is little essential work relating to piloting or maintenance of the vehicle to be done. Work that is invented just for the sake of killing time is unlikely to be very satisfying. It will be important for individuals to perceive their tasks as serving some useful purpose. Some time will have to be spent in doing housekeeping chores and some will be viewed as leisure, but it will undoubtedly prove to be necessary to have significant fractions of most days occupied with activities that are perceived as important to the mission or to other valued goals. Scientific experimentation and research could occupy much of this time, at least for those individuals who are scientists by profession or who would derive satisfaction from participating in scientific work.

The problem of leisure time is considerably more complicated for extended missions than for those of short duration. In the former case, one must be concerned not only with provision of short periods of free time at frequent intervals (e.g. daily) but also with the need for something analogous to holidays or weekends and vacations on earth, and with the question of how to ensure that individuals find it possible to spend that time to good advantage both from their point of view and that of the mission.

The Close Coupling of Humans and Computers

In 1983, the Space Systems Division of the NASA Office of Aeronautics and Space Technology convened a summer workshop (co-sponsored by the American Society for Engineering Education) at Stanford University to study the role of autonomy in space. The workshop report was issued in 1985, and has been referenced here as NASA-ASEE, 1985. Participants in the workshop included professors from universities across the country. "The workshop sought to generate recommendations on autonomous systems and human functions as well as on a space technology program directed toward symbiotic use of machines and humans"....."The principle objectives of the 1983 summer study were to examine interactions of humans and highly automated systems in the context of specific tasks envisioned for the space station, to search for optimum combinations of humans and machines, and to develop methodologies for selecting human-machine systems" (NASA-ASEE, 1985:2).

Participants in the workshop concluded from their study "that machines will not replace humans in space and that artificial intelligence (AI) systems will not have major impact on initial station design." To be sure, some aspects of the operation of the Space Station -- maintenance of orientation, control of in-station environment, pointing of antennas and solar panels -- will be done completely automatically, at least under normal circumstances. Moreover, the role of automation and artificial intelligence will increase as these technologies mature. But for the foreseeable future, and perhaps indefinitely, a great many aspects of the operation of the Station and of the performance of various missions will require the interaction of people with machines.

An increasingly common mode of interaction will involve supervisory control, which is viewed by some as intermediate between the use of teleoperators on the one hand, and robots on the other (Thiel and Kurtzman, 1983). In the case of teleoperators, the human has a "virtual" hands-on relationship but at a distance, as it were. In the case of robots, the relationship is of a qualitatively different type and may be remote both with respect to distance and time. The robot is given a capability by its designer to function relatively autonomously, albeit in accordance with principles incorporated in its design. In the case of supervisory control, the human is linked to the machine in real-time, but controls its operation only at a relatively high level. The human provides generic commands, which the system then translates into lower-level commands to the effectors that will, if all goes well, get the job done. How generic the commands are that the human operator provides depends on the system. The higher the level, the closer one comes to robotics, and at some point the distinction between the two modes disappears.

The fact that so many of the functions in the Space Station will be performed by people and machines in interaction means that the design of the various workstations and person-machine interfaces will be of central importance. There exists a substantial literature, much of it in design-guide form, that is highly relevant to this problem and that should be a major resource for designers of Space Station workstations and displays. But because the Space Station will be extending the frontiers of technology in several ways, designers will also have to consider questions for which answers have not yet found their way into design guides, and in some cases may not have yet been asked. Moreover, as Loftus et al. (1982) point out, the ultimate design objective of any manned spaceflight program is never that of optimizing the crew-to-spacecraft interface, but rather that of achieving overall program effectiveness; and given the numerous constraints within which such programs must function, this may mean that compromises will be necessary in various interface designs. Decisions about such compromises, and selections among various possible tradeoffs, should be made with the best understanding possible of their implications.

Among the issues relating to workstation and interface design that will be of special concern in the Space Station context are the following:

- How to design multifunction input-output devices so as to preclude confusion among functions.
- How to lay out the various display and input devices so as to ensure ease of location, interpretation and use.
- How to design the control and feedback interfaces for teleoperator systems.
- How to design the various input-output procedures (command and query languages, menus, abbreviations, symbols) so as to maximize their usefulness and minimize human error.

Many of the human factors issues relating to the design of workstations and interfaces will center on the question of how to get information -- precisely the right information in a useable format and at the appropriate time -- from a person to a machine or from a machine to a person. So in addition to the important questions of the physical designs of displays and input devices, there will be many issues relating to the design of methods and procedures for interacting with information per se. When will it make sense to use query languages as opposed to menus for searching a data base? Query languages put a greater learning burden on the user than do menus, but probably are faster for experienced users, because menus typically force one to go all the way down a tree step by step even when one knows precisely what one wants to ask at the beginning.

When menus are used, how should they be structured? This question subsumes a host of others, and although the lower-level questions sometimes seem to have intuitively obvious answers, research often reveals them to be more complicated than they appear. Consider the apparently simple matter of deciding how many items to show on a single node of a menu hierarchy. For a system with a given number of possible end points, there is a tradeoff between the number of options one sees at a given node in the hierarchy and the number of nodes required to get from the start to the finish. This breadth-versus-depth tradeoff has been the focus of some research (Dray et al., 1981; Miller, 1981; Seppala and Salvendy, 1985). While the results have not led to an unequivocal conclusion, there seems to be some agreement that menus that have very few items per level (say less than four) tend generally to be inefficient (Lee and MacGregor, 1985; Seppala and Salvendy, 1985). The situation is complicated, however, by the fact that how much breadth one can handle effectively will probably depend on how much experience one has had with the system. This may be an argument in favor of permitting a menu structure to modify itself to match the experience level of its user.

Much research effort is currently being devoted to the development of natural-language front ends for information systems. It seems likely that natural language systems with limited but useful capability will be available by the time the Station is operational. This is not to suggest that the reality of natural language capability will make other modes of interaction obsolete. The assumption that natural language would be the preferred mode of interaction with a data base in all cases is not beyond question; there is some evidence that more structured and constrained query languages may give superior performance in certain instances (Small and Weldon, 1983; for a review of human factors considerations that pertain to the design of query languages, see Ehrenreich, 1981).

Speech is also becoming increasingly feasible as a mode of communication between people and machines and could find at least limited use in the Space Station. The technology for synthesizing speech is improving steadily and although the best synthetic speech is still noticeably different from human speech and typically somewhat less intelligible, people get quite good at understanding it with only modest amounts (a few hours) of listening (Schwab et al., 1985). Speech understanding by computer is not so far along, but progress there is also being made. The technology for isolated word recognition probably is sufficiently mature to be used in a Space Station context, and more ambitious uses of speech understanding technology may be feasible by the time the Station becomes operational.

Stress and Performance In Space

Efforts to anticipate how humans will perform on extended space missions have focused on certain ways in which the space environment differs from more familiar environments on earth and on various types of stressors that could have either acute or cumulative long-term effects. Some of the characteristics of the Space Station environment may themselves be stressors, if not continuously, at least under certain conditions. It will be convenient, therefore, to begin this section with a brief discussion of stress in general terms and then to consider specific environmental characteristics or stressors that might be expected to affect performance and hence productivity significantly.

Effects of Stress on Performance

Stress is likely to be a factor in the Space Station and to affect productivity in several ways. First, under the best of circumstances the Station and its personnel are always at risk. While we would not expect individuals to spend every waking moment worrying about safety, it would be surprising indeed if there were not a constant underlying sensitivity to the tenuousness of the situation; this might be considered a type of chronic stress. Second, from time to time, individuals or the entire occupancy of the Station may be stressed acutely as a consequence of an unanticipated event or situational change. Third, stress may also be caused by factors that are relatively long lived, but not necessarily chronic. These include confinement and social isolation, sensory-motor restriction, interpersonal frictions, dissatisfactions with certain aspects of one's duties or the Station's operating procedures, and anxieties about events or situations on earth. The list of possibilities is easily extended.

According to Sharit and Salvendy (1982) most of the definitions of stress that one finds in the literature reflect biases related to the scientific orientation of the writers and fail to capture the many-faceted nature of the phenomenon. Fidell (1977) has noted that some authors who have written about stress have avoided defining the term (e.g. Broadbent, 1971; Welford, 1974) presumably on the assumption that the word is intuitively meaningful: most of us know what it means to be stressed from personal experience.

In his review of effects of stress on performance, Fidell (1977) classified stressors as physical, physiological, psychological, and social. Lazarus and Monat (1977) used the last three of these categories but not the first.) In the first category Fidell included thermal (heat, cold, humidity) mechanical (vibration, acceleration, fluid pressure) and sensory (noise, glare, odor, deprivation) and ingested or inhaled substances (drugs, noxious fumes, insufficient oxygen). As physiological stressors he listed musculoskeletal fatigue, sleep deprivation, age, disease, and illness. As psychological stressors he distinguished between cognitive (information or perceptual under/overload) and emotional types (fear, anxiety, insecurity, frustration). The social stressors in his list were occupational factors (e.g. career pressures) organizational structures, major life events, crowding, and solitude. Fidell also pointed out that stress is sometimes thought of as an effect and sometimes as a cause. It is assumed to be an effect, for example, of a perceived threat to one's safety or the imposition of a task that exceeds one's ability to perform. On the other hand, it is sometimes identified as the cause of poor performance or of otherwise inexplicable behavior. It is also sometimes viewed as the cause of certain types of medical problems such as ulcers, colitis, and cardiac arrhythmias.

Effects of stress on performance are not easy to summarize. Mild to moderate stress for short durations can have a beneficial effect in many situations, possibly as a consequence of increased alertness and the energy spurt that comes with the greater-than-normal production of adrenaline and other hormones. Excessive stress can produce deterioration of performance. Frequent experience of stressful events tends to be accompanied by atypically high incidence of illness of various sorts (Norman et al., 1985). A relatively unexplored aspect of effects of stress on performance relates to how performance changes after a temporary stressor has been removed.

The study of effects of stress is further complicated by the fact that people adapt or accommodate to stressors, especially if they are only moderate in degree and relatively invariant over time. Noise, for example, can be a stressor, but people who work in a continuously noisy environment seem to adapt to it so that its effects as a stressor diminish greatly or disappear. Unexpected substantive change in the level or characteristics of the noise, however, may have disruptive effects.

Leventhal and Lindsley (1972) distinguish between danger control and fear control as two types of concern that one may have in a threatening situation. Concern for danger control is focused on the threatening situation and on how to rectify it. Concern for fear control is focused on the fear response and on how to keep it in check. Both are legitimate concerns and training in preparation for extended space missions should take both into account.

Stress is likely to be an important factor in the Space Station and its effects on productivity could be substantial. Moreover several stressors may be operating simultaneously, producing complex interactive effects, and the stressors will be interacting also with other variables in ways that cannot be foreseen. In the remainder of this section, several of the stressors that could be especially important in the Space Station environment are briefly noted. Exactly how these factors, especially in combination, will affect performance and productivity is not known; that their effects will be substantive, however, seems highly likely.

Weightlessness

Weightlessness has been emphasized as a major feature of a spacecraft environment that could give rise to physiological problems such as altered fluid and electrolyte balances, and deconditioning of specific systems such as the cardiovascular, musculoskeletal, metabolic, and neuroendocrine systems (Lindsley and Jones, 1972). Problems of these types have not yet been shown to be sufficiently severe to preclude prolonged space missions; on the other hand, how they will manifest themselves in long duration missions remains to be seen.

In retrospect many, perhaps most, of the observed short-term effects of weightlessness on human functioning probably were predictable, but many of them were not predicted. In thinking about what it would be like in a weightless environment, one may find it easy to imagine being able to float freely in space and fail to realize that it will also be difficult to stand on the floor, sit in a chair, or maintain any fixed position without restraints. Who would have thought to ask whether it would be possible to burp? Or whether it would be difficult to bend down to tie one's shoes?

Unfamiliar Motion

Closely related to weightlessness are the various types of motion that can produce motion sickness (Kennedy, 1972). Even astronauts who have had training intended to reduce the probability of motion sickness have experienced such sickness during space flight, usually during the first few days of a mission, although nausea has typically not precluded crew members from carrying out essential activities (Garriott, 1974). There is some indication that dizziness is more likely to be induced in situations that permit individuals to move around in large spaces than in those in which they are more confined (Berry, 1969, 1970). When severe, motion sickness can be debilitating.

Motion Restriction

On the opposite end of the spectrum from the concern for unfamiliar motion is that for motion restriction. A variety of restrictive conditions on earth have been studied with a view to determining their physiological and psychological effects. These include immobilization from a plaster cast, bed rest, and prolonged confinement in submarines, space cabin simulators or other chambers (Fraser et al., 1972). Among the most apparent physiological effects of long-term restriction of activity appear to be cardiovascular and musculoskeletal deconditioning, including some bone decalcification. Other possible effects include electrolyte imbalances and hemolytic anemia.

As measures that can be taken to prevent or counter the deconditioning effects of motion restriction, Fraser et al. (1972) list the following: adequate free living space (200-250 cubic feet per person at a minimum, up to 600-700 cubic feet per person as the "optimal, maximizing habitability in the light of other requirements"), adequate exercise, applied pressure (to control for fluid volume loss and orthostatic intolerance of deconditioning), artificial gravity (seen as expensive and therefore less practical than other approaches), and hormones and drugs (primarily to control fluid loss).

Sensory and Perceptual Restriction

What is known about the effects of sensory and perceptual deprivation or restriction on human performance has been summarized by Schultz (1965) and Zubek (1973). Eason and Harter (1972) have also reviewed the literature on this topic through 1972 and attempted to extract from it information that would be relevant to the prediction of human performance in prolonged space flight. (Sensory deprivation or restriction connotes an absence or marked attenuation of sensory input to one or more modalities; perceptual deprivation or restriction suggests reduction in patterned stimulation.) Eason and Harter noted that the studies available for their review did not include any in which the period of confinement or isolation exceeded a few weeks. Russian investigators have done studies on effects of confinement in which subjects spent as long as one year in relatively isolated environments but details have not been available.

The data from these studies are fragmentary at best and do not constitute a coherent set of findings. Results of individual studies are often mutually contradictory, some showing negative and some positive effects of deprivation on subsequent perception or performance. As they relate to long duration space missions, Eason and Harter (1972:101) see the findings as "rather heartening, for they suggest that the effects of severe sensory or perceptual restriction, isolation, and confinement are so minor, except in a few instances, that they are difficult to demonstrate with any degree of consistency not only from one laboratory to another but often within the same laboratory".

Eason and Harter caution against making predictions about an astronaut's sensory, perceptual and motor functions during long-range missions on the basis of experiments involving relatively short-term isolation. The results of such studies do provide a basis for raising questions and suggesting directions for research that can be relevant in the space flight context, and had they yielded solid evidence of large effects of isolation on sensory or motor functions, they would have raised some concerns about potential effects in the Space Station program. "As it turns out, the results of studies summarized in this paper suggest that only minimal and relatively insignificant changes in sensory and motor function are likely to occur during long-duration missions" (Eason and Harter:103). Eason and Harter point out that in extended space flight, boredom from repetition of stimulation may turn out to be a more important determinant of performance than sensory deprivation as such. They note, however, that past studies have been too limited in various respects to provide a basis for confident predictions about possible effects of confinement and isolation in space flight and urge further study of these variables under conditions that will assure the applicability of the results.

DRAFT
NOT FOR DISTRIBUTION, ATTRIBUTION OR QUOTATION

Sleep Interference

Sleep disturbances and irregularities take many forms. The most obvious departure from a typical sleep-wake cycle is total sleep deprivation -- going for extended periods of time without any sleep. Other types of irregularity include unusual cycles (e.g. 4 hours of sleep, 4 hours of work), change in phase in the normal sleep-wake pattern (e.g. shifting from a work-in-the-day-sleep-at-night pattern to a sleep-in-the-day-work-at-night pattern), disruption of the quality of sleep (fitful or shallow sleep; decrease in stage-3 and stage-4 sleep) resulting from environmental disturbances, psychological stress or other unusual factors. Studies of shift workers have shown that changing from day to night shift typically results in a reduction (1 to 2 hours) in the duration of the main sleep period, an increase in average total amount of sleep per 24 hour period -- due to naps taken outside the main sleep period and extra sleep on rest days -- and a change in the quality of sleep (Akerstedt and Gillberg, 1981; Tilley et al., 1981; Tilley et al., 1982). Indicants of quality include time to sleep onset, number of awakenings, number of body movements, and number of changes in sleep stage (Johnson et al., 1972).

How sleep disturbances affect performance is not understood well. Data suggest that sleep loss is likely to have deleterious effects on tasks for which sensory stimulation tends to be low and the rate of data handling is not under the individual's control (e.g. monitoring or vigilance tasks) and to have less effect on the performance of complex intellectual tasks involving problem solving and logical analysis (Johnson et al., 1972). Somewhat independent of the question of the effects of sleep disturbances on performance is that of their effect on moods and attitudes. Insomnia is often linked to depression, tension, and irritability. Whether there is a cause-effect relationship and, if so, which way it goes, are not known for certain.

Determination of optimal work-rest cycles will involve consideration of a variety of factors, technological, psychological and social. How often and how long people will need (or want) to sleep will depend in part on the demands of their jobs, and in part on the conditions of the sleeping environment. Requirements for sleep are likely to differ from person to person. With respect to social factors, there is some evidence that crews prefer to be on the same work-rest cycle insofar as possible, and work and get along better when this is the case.

The importance of rest periods interspersed among work tours has been known at least since Taylor's (1947) early studies. Exactly how rest breaks should be scheduled, however, or how this should depend on the nature of the work being done, has not been established very precisely. It is not even clear that it is always optimal for work breaks to occur on a fixed periodic schedule.

Any attempt to understand the relationship of sleep disturbances and stress will illustrate the problem of distinguishing cause from effect.

Sleep disturbances, such as those caused by unusual work-rest cycles or the need for prolonged wakefulness to deal with an emergency situation are seen as sources of both physiological and psychological stress. On the other hand, stress originating from other sources can be the cause of insomnia or other sleep-related difficulties.

Boredom and Other Motivational Problems

It is somewhat paradoxical that one of the major concerns about such a risky venture as extended space flight should be a concern about boredom. However, boredom and various attendant complications could be among the most serious problems that have to be faced. Although surprisingly little empirical work has been done on boredom (Smith, 1981), it has been identified as a significant problem for people living in restrictive environments with monotonous schedules for weeks or months at a time. It is believed to have detrimental effects on motivation and morale and to lead to increased frequency of complaints of headache and other physical problems. The tendency for motivation to decrease over a period of extended confinement is a common report from studies of small groups in isolated environments (Smith, 1969).

Behavioral evidences of a loss in motivation include diminution of one's ability or willingness to engage in sustained purposeful activity. There is some evidence that declining motivation has a physiological correlate in a decreasing frequency of alpha rhythm in the EEG wave (Zubek et al., 1969). This is an interesting finding because it suggests the possibility of using alpha rhythm as a means of monitoring individuals' momentary cognitive state and of predicting how productive they are likely to be in specific work situations.

Many studies have failed to find a decrement in ability to perform some types of cognitive tasks -- and in some cases have even found an improvement in that ability -- as a consequence of spending substantial amounts of time in confined environments. However, Johnson et al. (1972) note the possibility that studies that measure performance under the circumstances in which motivation might be expected to be low often risk artifactual results by virtue of the possibility that the experimental task itself, if unusual within the context, may be sufficiently arousing and rewarding to improve temporarily the subjects' motivational state. After reviewing the pertinent literature, Johnson, Williams and Stern concluded that very little is known about how to reduce monotony and boredom during long periods of group confinement.

Social Isolation

Isolation can mean a variety of things. Brownfield (1965) identifies four: spatial confinement; separation from persons, places, or things that one values highly; reduction or restriction of sensory stimulation; and reduction in the variability and structure of stimulation. The first, third and fourth of these connotations have already been mentioned. Unfortunately, effects of isolation often cannot be distinguished from those of confinement, motion restriction and social crowding, because these conditions typically occur together; nevertheless, it is believed that social isolation could prove to be among the most important stressors in the context of prolonged space missions. Some concern has been expressed that it, combined with some of the other characteristics of the space environment such as weightlessness, empty time, and distortion of the usual balances among sensory inputs, may lead to an increased frequency of daydreaming and fantasizing and a progressively more subjective orientation (Leventhal and Lindsley, 1972). Studies of groups that have spent extended periods (months) in relative isolation have shown that individuals tend over time to withdraw and become more psychologically remote from other members of the group (Haythorn et al., 1972). According to Sells and Gunderson (1972:204), extended isolation and confinement of small groups on earth (e.g. at scientific stations in Antarctica) can increase the probability of "irritability and depression, sleep disturbances, boredom, social withdrawal, dissatisfaction, and deterioration in group organization and cohesion". Enriching the stimulus environment can counteract this tendency to some degree, but the stimuli must be meaningful and of interest to the people involved. There is some evidence that part of the withdrawal complex is a decreased tendency to avail oneself of whatever opportunities for stimulation the environment provides.

Special problems may arise when an individual especially close to a person on an extended mission becomes seriously ill (e.g. a child, spouse, or parent) and it is impossible for the person to return to earth, or if unanticipated events of major significance occur on earth during a prolonged mission. The effects of such happenings on attitudes and morale could be substantive. It is easy to imagine other examples of events on earth that could prove to be stressors to people in space. Inasmuch as communication between earth and the station will probably be primarily through ground control stations, at least for some time, information that could have a detrimental effect on the morale of members of the Space Station crew could be withheld from them. Consideration of such a policy raises a serious ethical issue, however, and would probably not be tolerated in any case. There are many reasons for maintaining frequent, if not constant, communication with earth. Not least among these is the need for inhabitants of the station to communicate frequently with people other than themselves.

Excessive Workload

Excessive task demands can be a source of stress and can lead to serious performance decrements. When even moderate task demands are coupled with the constant possibility of catastrophic errors, long term exposure to the situation can produce a variety of stress-related symptoms. One inherently stressful job that has been the focus of considerable attention by researchers, and the general public as well, is air traffic control (Cobb and Rose, 1973; Crump, 1979; Finkelman and Kirschner, 1981; Hailey, 1968). The stress in this case probably stems in large part from the facts that errors in performance can result in human fatalities and that most aircraft accidents are due to human error (Danaher, 1980).

Task demands in the Space Station are unlikely to be excessive for sustained periods of time, although they could be high at critical mission junctures and could become excessive during emergencies. Perhaps more important is the ever-present possibility of human error having a catastrophic result. Every attempt will be made, of course, to ensure that the operating procedures are fail-safe and that any errors that can be anticipated are recoverable, but some degree of uncertainty in this regard is bound to remain, and with it some level of task-induced stress.

Acute Medical Problems

With respect to the control of medical problems within a spacecraft, the emphasis has to be first on prevention (Fraser et al., 1972). Having taken all reasonable preventive measures, however, the chance that medical problems will arise on any long-duration mission is high. Within the Space Station there will be the possibility of many of the same types of physical injuries arising from accidents with equipment that might occur on earth. In addition there are certain types of mishaps that are relatively unique to the space environment; these include the aspiration of particles that float in the weightless environment of the station, effects of prolonged exposure to atypical mixes of atmospheric gases or pressures, exposure to high-Z particles -- high energy particles of high atomic number -- or other forms of radiation, and heat disorders resulting from malfunctioning of a pressure suit during EVA. Fraser et al note also the possibility that some medical problems that would be very easy to treat on earth could become significant in space, either because of inadequate treatment facilities (e.g. acute appendicitis) or because the medical problem has been complicated by virtue of various ways in which the body has adapted physiologically to the weightless environment (e.g. reduction in blood volume due to weightlessness).

Other Sources of Stress

Other features of space flight that could also be problematic include the absence of normal terrestrial time references, and possibly altered magnetic fields (Fraser et al., 1972). Changes in lines of authority that could prove necessary from time to time could pose challenges for social stability of the spacecraft community. The need for privacy could be an especially important one in extended space flight; the ability to have some time and place wholly to oneself on a fairly regular basis may prove especially important in this environment. Sharing of sleeping quarters and other personal space over long periods of time can increase the frequency and seriousness of interpersonal frictions. Habitability of the spacecraft will increase in importance with increases in the durations of space missions. The difficulty of maintaining a habitable environment will also increase with mission duration.

It will be particularly important that inhabitants of the Space Station be able to resolve, quickly and expeditiously, any interpersonal conflicts that arise. Presumably selection procedures will disqualify from participation in space missions individuals for whom the probability of interpersonal disputes or frictions is determined to be high. It will be important for those who do qualify to receive such training as is available regarding how to avoid various types of interpersonal disputes, and how to resolve them when avoidance proves to be impossible.

Individuals react differently to the same stressors, depending on motivation, familiarity with the situation, appropriateness of training, degree of confidence in own ability to cope, degree of confidence in supporting colleagues and accessible resources, and other factors. There is some evidence that the magnitude of physiological reaction (e.g. increased pulse rate) to psychological stress is likely to be less for individuals who are aerobically fit than for those who are not (Holmes and Roth, 1985). Tests that provide a reliable indication of how individuals will react to the types of stressors they are likely to encounter in the Space Station environment would be useful both for purposes of selection and for identifying specific training needs. Development and validation of such tests are worthwhile goals. Similarly, development of more effective methods of increasing tolerances to specific stressors and of improving the ability of individuals to function effectively in spite of them should be continuing objectives.

Loftus et al. (1982:II-34) note that stress does not seem to have led to performance degradation so far in the spaceflight program. They attribute failure to observe such degradation "to substantial overtraining of flight crews for the tasks they must perform, diverse and interesting stimuli present in the real environment contrasted with minimum stimulation environment in simulations, and stronger motivation in flight crews compared with test subjects". It would be unwise to extrapolate the relative unimportance of stress as a determinant of performance in the early space program to the future, however; the much longer durations of the missions and the inclusion of participants who are not professional astronauts are two major differences that could make stress of various types much more consequential.

CONCLUSIONS AND RECOMMENDATIONS

The Space Station program is an ambitious undertaking. Establishing a permanently manned facility in space will be expensive and risky, but the long-range benefits for humankind that could result from success in this endeavor are surely very great. Keeping the program moving forward without unpleasant surprises and major setbacks will require intensive planning, continual evaluation of plans, replanning based on the results of evaluations, and compulsive attention to details of countless types.

In the remainder of this paper, I shall identify what appear to me to be some of the major needs, especially research needs, relating to productivity in the Space Station. At the beginning of this paper, it was noted that the term <u>productivity</u> is used in a variety of ways and often without a very precise connotation, and that except in certain highly-structured situations, how to quantify productivity unambiguously is not clear. If high productivity is to be an explicit objective of the space program, some consideration must be given to how it is to be measured or otherwise assessed in this context. Assessment will be desirable at various levels -- that of the overall Space Station program, that of specific missions, that of specific crews during designated periods of time, and that of individuals performing specific tasks.

For present purposes, it is assumed that enhancements — increases in the efficiency, accuracy, reliability — of the performance of humans or human-machine systems are very likely to improve productivity by nearly any reasonable definition and measurement technique. The recommendations that follow are predicated on that assumption. Research that is alluded to in some of these recommendations is already underway, in NASA laboratories and elsewhere. I am aware of some of these efforts, but there undoubtedly are many of which I am not. Inclusion in this list signifies only my opinion that the topic deserves attention; if it is getting it already, so much the better. While all of these recommendations are considered important to the Space Station program, they are not all uniquely applicable to it. Some of them are similar to recommendations that would apply to the design and development of any complex system that will have people interacting with computer-based tools in non-trivial ways (Nickerson et al., 1984).

- There is a need to organize the information that has been obtained from research on earth or from data gathered in previous space flights that is relevant to human performance in space. This information should be organized and indexed so as to make it highly accessible to scientists and engineers in the space program.

- It would be useful also to commission the compilation of an encyclopedia of ignorance about productivity, and performance more generally, in space. The primary objective should be to identify as many as possible of the important unanswered questions about performance in space. Questions should be prioritized with respect to urgency, and classified in terms of the kind of research that could lead to answers.

- What information will be required by specific members of the Space Station team at specific times needs to be determined. This includes determining what information should be presented spontaneously, and in such a way as to capture the intended receiver's attention, what information should be available explicitly on some display all (or most) of the time, and what information should be available but presented only on request.

- Possible and most-likely patterns of communication or information flow both within the Space Station and between the station and earth need to be understood better.

- More effective means of providing EVA access to data-base information pertinent to EVA tasks are needed.

- An inventory of tasks that people will be expected to perform in the Space Station should be compiled.

- Procedure descriptions should be evaluated for accuracy and clarity.

- Criteria need to be established regarding what aspects of the Space Station's operation should be automated. The rule that anything that can be automated (effectively, safely) should be automated is not necessarily a good rule. There may be some functions that can be done acceptably by either people or machines that should be done by people. Issues of morale, perception of control, and skill maintenance must be considered as well as that of technical feasibility.

- More research is needed on the question of how much "intelligence" to build into teleoperator or telerobot systems, and how much to rely on remote control by humans.

- The design of computer-based aids for trouble shooting, problem solving and decision making, and of the protocols for interacting with them deserves considerable attention.

- Efforts to advance the state-of-the-art of aiding human operators through the use of "intelligent", or "expert-system" software should be supported: potential applications in the Space Station program include fault detection, identification, and repair; planning and plan revising; and crisis management.

- The knowledge of astronauts and space professionals must be codified to provide the basis for the development of expert systems and knowledge-based aids.

- The phasing of expert system technology into operational situations as its evolution warrants will represent an ongoing challenge into the indefinite future.

- Possible problems involved in having crew members share responsibility of high-level cognitive tasks with "smart" software or expert systems need to be identified; policies should be established for deciding when to trust a system and when to override it.

- Design of the various interfaces through which Space Station personnel will interact with the numerous systems and subsystems on board is among the most critical problems to be solved, from a human factors point of view. There is a body of literature relating to the design of workstations and displays that should be consulted; however, much remains to be learned about how best to represent and present information in various Space Station contexts. This topic deserves a continuing effort of research focused on the identification of display formats, information coding dimensions, and input techniques that are especially well suited to the Space Station environment and the demands of specific tasks that are to be performed.

- Proposed or planned displays and work stations should be evaluated in terms of conventional human factors criteria: lighting, glare, flicker, contrast, character/symbol legibility/interpretability, functional-positional relationships, clutter, and so on.

- Display configurations and symbology must be designed and evaluated; this includes determination of content and format of specific-purpose displays. Display coding dimensions must be selected so as to minimize confusion arising from multiple functions of a given display space.

- A better understanding is needed of when to use menus and when to use command languages as input methods. The menus and languages to be used must be designed, evaluated and refined.

- There is a need to identify situations in which voice could be used to advantage as an input or output medium, given the probable state-of-the-art of voice recognition and production technology over the next decade or so.

- Further work is needed on the design of control and feedback interfaces for remote manipulators, teleoperators, and semi-autonomous systems. The problem is complicated when the distance between the devices and their operators is great enough to cause significant communication delays.

- The need for high resolution, stereo visual feedback from teleoperator systems should be studied and the feasibility of its use explored.

- More effective helmet-mounted displays for use in EVA should be a continuing research objective.

- The technology for tracking eye fixation and movement, and hand and finger position and movement could have applications in the Space Station, but need to be developed further.

- The technology needed to make a virtual-interface approach to teleoperator control a practical reality requires further exploration.

- Acquisition of anthropometric, range of motion, strength, and force and torque application data, with and without pressurized suits, should be continued.

- The ability to measure and monitor mental workload could be useful, especially for the establishment of crew responsibilities in the Station's day-to-day operation and in high-activity situations. But techniques that are to be used in operational contexts must be unintrusive, and this rules out the applicability of many of those that have been used to study mental workload in the laboratory.

- A catalog of possible human errors (of both commission and omission) that could have non-trivial consequences in the Space Station should be developed; potential errors should be rated as to seriousness and probability of occurrence, and the results used to develop safeguards and error detection and recovery procedures.

- A detailed study of human errors that are actually made in the Space Station environment will be very useful, as it has been in other contexts (Meister, 1966; Swain, 1970, 1978).

- Methods of assuring the maintenance of critical skills that are typically used only in the event of a system malfunction or failure must be developed.

- Effects of prolonged living in restricted environments on work performance, social behavior and mental state deserve further study. More specifically, attempts should be made to identify aspects of such environments that are the major determinants of behavioral, cognitive or emotional effects.

- Special attention should be given to the types of interpersonal tensions and conflicts that are likely to arise in the Space Station environment and the development of effective techniques for relieving or resolving them.

- The question of how to occupy long periods of time during which the operational demands of the spacecraft are minimal deserves considerable attention. The maintenance of motivation, alertness and social stability during extended stretches of being, in essence, passengers on an automatically piloted craft represents a significant challenge.

- Presumably, productivity in space can be enhanced by factors that contribute to the maintenance of high levels of alertness, motivation and general physical and mental well being. We need to understand better how these variables depend on such factors as appropriate diet; regular physical exercise; the opportunity to engage in interesting and valued activities in free time; frequent communication with earth, not only regarding mission matters, but regarding those of personal interest; adequate variety in job responsibilities; adequate rest; and extensive use of error detection and failsafe procedures (especially those that can be automated).

- We need also to learn more about the relationships among certain performance or psychological variables (attention, vigilance, perception, memory, learning, thinking, and judgement) and indicants of physiological state (EEG, evoked potential, contingent negative variation, heart rate, blood pressure, respiration, skin temperature, galvanic skin response). To the extent that variables in the latter category can be shown to be reliable indicants of the quality of specific types of human performance, consideration should be given to the development of unintrusive ways of monitoring them, at least at critical times, and using the results of the monitoring to enhance performance in various ways (Johnson et al., 1972). Although techniques exist for doing such monitoring, they tend to be sufficiently intrusive to interfere with the monitored individuals' performance of their primary tasks and to be less reliable than is desired. A continuing goal of research should be the development of less intrusive and more reliable techniques for monitoring cognitive state.

- o The ability to monitor -- and in particular to detect significant changes in -- physiological and psychological states could prove to be especially important in long-term space missions. State changes that could be important to detect include both temporary fluctuations in alertness and long-range changes in general physical condition, motivation and mood.

- o Biofeedback technology is still in its infancy, however the evidence is clear that people can learn, within limits, to control certain physiological functions that had been thought to be completely automatic. Further study of biofeedback techniques is warranted with a view to their possible application in the Space Station for purposes of controlling tension, facilitating good quality sleep, and otherwise tuning physiological states to enhance either performance or rest.

- o Studies of the mental models that crew members or perspective crew members develop of the Space Station and its hardware and software componnents could help determine what kinds of models are acceptable for conveyance to future participants in Space Station missions.

- o There is a need for better rapid prototyping capabilities especially for prototyping candidate interface designs.

- Procedures and policies must be established for acquiring data in space that can be used to relate productivity and performance to the numerous variables that are believed to affect them in significant ways.

- It is not likely that predictions about performance of humans in space can be very accurate very far into the future. A reasonable goal is the development of a predictive model, based on what is currently known from data collected on earth and from studies of performance in space to date, with the intent of modifying that model continually as further relevant data are obtained, especially from experience in space. Conditions in space exploration will change and the durations of stays in space will increase, so the model will have to evolve to accommodate those changes. On the assumption that the changes that occur will be evolutionary and relatively continuous, one can hope for a model that is highly predictive of the situation that is current at any given time and reasonably predictive of the situation as it is anticipated to be in the near-term future.

DRAFT

NOT FOR DISTRIBUTI...
ATTRIBUTION
OR QUOTATION

REFERENCES

Akerstedt, T., and Gillberg, M.

1981 Sleep disturbance and shift work. In A. Reinberg, N. Vieux, and P. Andlauer, eds., Night and Shift Work: Biological and Social Aspects. Oxford: Pergamon Press.

Baily, M. N.

1986 What has happened to productivity growth. Science 234:443-451.

Bechtold, S. E., Ganaro, R. E., and Summers, D. L.

1984 Maximization of labor productivity through optimal rest-break schedules. Management Science 30:1442-1458.

Berry, C. A.

1969 Preliminary clinical report of the medical aspects of Apollos VII and VIII. Aerospace Medicine 40:245-254.

1970 Summary of medical experience in the Apollo 7 through 11 manned spaceflights. Aerospace Medicine 41:500-519.

Bowen, W.

1986 The puny payoff from office computers. Fortune 5:20-24.

Broadbent, D. E.

1971 Decision and Stress. New York: Academic Press.

Brownfield, C. A.

1965 Isolation, Clinical and Experimental Approaches. New York: Random House.

Cameron, C.

1971 Fatigue problems in modern industry. Ergonomics 14:713-720.

1973 A theory of fatigue. Ergonomics 16:633-648.

Casali, J. G., and Wierwille, W. W.

1983 A comparison of rating scale, secondary-task, physiological, and primary-task workload estimation techniques in a simulated flight task emphasizing communications load. Human Factors 25(6):623-641.

Chaudhari, P.

1986 Electronic and magnetic materials. Scientific American 255(4):136-144.

Chiles, W. D., and Alluisi, E. A.

1979 On the specification of operator or occupational workload with performance-measurement methods. Human Factors 21:515-528.

Clearwater, Y.A.

1985 A human place in outer space. Psychology Today 7:34-43.

Clearwater, Y. A. and Kaspar, P.

1986, Humanizing space. Professional Careers. 10/11:5-8.

Cobb, S., and Rose, R. M.

1973 Hypertension, peptic ulcers and diabetes in air traffic controllers. Journal of the American Medical Association 224:489-492.

Crump, J. H.

1979 A review of stress in air traffic control: Its measurement and effects. Aviation, Space and Environmental Medicine 50:243-248.

Danaher, J. W.

1980 Human error in ATC system operations. Human Factors 22(5):535-545.

Dray, S. M., Ogden, W. G., and Vestewig, R. E.

1981 Measuring performance with a menu-selection human computer interface. Pp. 746-748 in Proceedings of the Human Factors Society 25th Annual Meeting.

Drucker, P. F.

1986 The Frontiers of Management. New York: Truman Talley Books

Eason, R. G. and Harter, M. R.

1972 Sensory, perceptual, and motor factors. Human Factors in Long-Duration Spaceflight. Washington, DC: National Academy of Sciences.

Eggemeier, F. T.

1980 Some current issues in workload measurement. Pp. 669-673 in Proceedings of the Human Factors Society 24th Annual Meeting.

Ehrenreich, S. L.

1981 Query languages: Design recommendations derived from the human factors literature. Human Factors 23(6):709-725.

Fidell, S. A.

1977 Effects of stress on performance. The C^3 System User Vol I: A Review of Research on Human Performance as it Relates to the Design and Operation of Command, Control, and Communication Systems. BBN Report 3459.

Finkelman, J. M., and Kirschner, C.

1981 An information-processing interpretation of air traffic control stress. Human Factors 22(5):561-567.

Fitts, P. M. (Ed.)

1951 Human Engineering for an Effective Air-Navigation and Traffic-Control System. Columbus, OH: Ohio State University Research Foundation.

Fraser, T. M., Busby, D. E., Adey, W. R., and Galambos, R.

1972 Physiological and Medical Factors. Human Factors in Long-Duration Spaceflight. Washington, DC.: National Academy of Sciences.

Ganaro, R. E., and Bechtold, S. E.

1985 A study of the reduction of fatigue impact on productivity through optimal rest break scheduling. Human Factors 27:459-466.

Garriott, O. K.

1974 Skylab report: Man's role in space research. Science 186:219-225.

Garriott, O. K., Parker, R. A. R, Lichtenberg, B. K., and Merbold, U.

1984 Payload crew members' view of Spacelab operations. Science 225:165-167.

Giuliano, V. E.

1982 The Mechanization of Office Work. Scientific American 247:(3)148-164.

Gunderson, E. K. E. (Ed.)

1974 Human Adaptability to Antarctic Conditions. Washington, DC: American Geophysical Union.

Gunderson, E. K. E.

1963 Emotional symptons in extremely isolated groups. Archives of General Psychiatry 9:74-80.

Gunderson, E. D. E. and Nelson, P. D.

1963 Adaptation of small groups to extreme environments. Aerospace Medicine 34:1111-1115.

Gunn, T. G.

1982 The mechanization of design and manufacturing. Scientific American 247:(3)114-130.

Hailey, A. H.

1968 Airport. New York: Doubleday and Company.

Haythorn, W. W., and McGrath, J. J., and Hollander, E. P., and Latane, B.

1972 Group processes and interpersonal interaction. Human Factors in Long-Duration Spaceflight. Washington, DC.: National Academy of Sciences.

Hicks, T., and Wierwille, W.

1979 Comparison of five mental workload assessment procedures in a moving-base simulator. Human Factors 21:129-144.

Holmes, D. S., and Roth, D.

1985 Association of aerobic fitness with pulse rate and subjective responses to psychological stress. Psychophysiology 22(5):525-529.

Howard, R. D., Miller, R. H., Minsky, M. L., and et al.

1982 Space Applications of Automation, Robotics and Machine Intelligence Systems (ARAMIS). Phase I, Structural Assembly Expansion. Volumes 1-5, Space Systems Laboratory and Artificial Intelligence Laboratory, SSL Reports #25-82 - #29-82, Massachusetts Institute of Technology, Cambridge, MA.

Isreal, J. B., Wickens, C. D., and Chesney, G. L.

1980 The event-related brain potential as an index of display-monitoring workload. Human Factors 22(2):211-224.

Johannsen, G.

1979 Workload and workload measurement. Pp. 3-11 in N. Moray, ed., _Mental Workload: Its Theory and Measurement_. New York: Plenum.

Johannsen, G., Moray, N., Pew, R., Rasmussen, J., Sanders, A., and Wickens, C.

1979 Final report of experimental psychology group. Pp. 101-114 in N. Moray, ed., _Mental Workload: Its Theory and Measurement_. New York: Plenum.

Johnson, L. C., Williams, H. L., and Stern, J. A.

1972 Motivation, cognition, and sleep-work factors; central-and autonomic-nervous-system indices. _Human Factors in Long-Duration Spaceflight_. Washington, DC.: National Academy of Sciences.

Jones, E. R., Chiles, W. D., and Voas, R. B.

1972 Human factors and operational requirements. _Human Factors in Long-Duration Spaceflight_. Washington, DC.: National Academy of Sciences.

JSC

1979 _Space Operations Center, A Concept Analysis_. JSC-16277. NASA/Johnson Space Center.

Kalsbeek, J. W. H.

1968 Measurement of mental work and of acceptable work: Possible applications in industry. International Journal of Production Research 7:33-45.

1971 Sinus arrhythmia and the dual task method in measuring mental load. In W. T. Singleton, J. G. Fox, and D. Whitfield, eds., Measurement of Man at Work. London: Taylor and Francis.

Kennedy, R. S.

1972 A Bibliography of the Role of the Vestibular Apparatus Under Water and Pressure: Content-Oriented and Annotated. M4306.03.5000BAK9, Naval Medical Research Institute Report No. 1.

Kessel, C. J., and Wickens, C. D.

1983 The transfer of failure-detection skills between monitoring and controlling dynamic systems. Human Factors 24(1):49-60.

King, K. R., and Rouen, M. N.

1982 Extravehicular Mobility Unit. NASA-JSC-18201. Satellite Services Workshop 1:44-72.

Lazarus, R. A., and Monat, A.

1977 Stress and coping -- Some current issues and controversies. In A. Monat and R. S. Lazarus, eds., Stress and coping. New York: Columbia University Press.

Lee, E., and MacGregor, J.

1985 Minimizing user search time in menu retrieval systems. Human Factors 27(2):157-162.

Leventhal, H., and Lindsley, D. B.

1972 Subjective states. Human Factors in Long-Duration Spaceflight. Washington, DC.: National Academy of Sciences.

Lindsley, D. B., and Jones, E. R.

1972 Physical factors. Human Factors in Long-Duration Spaceflight. Washington, DC.: National Academy of Sciences.

Loftus, J. P., Jr., Bond, R., and Patton, R. M.

1982 Astronaut activity. Workshop Proceedings: Space Human Factors, 1, Leesburg, VA, II-28-II-64.

McGrath, J. J.

1965 Performance sharing in an audio-visual vigilance task. Human Factors 7:141-153.

Meister, D.

1966 Human factors in reliability. In W. G. Ireson, ed., Reliability Handbook. New York: McGraw-Hill.

Miller, D. P.

1981 Depth/breadth tradeoff in hierarchical computer menus. Proceedings of the Human Factors Society 25th Annual Meeting. Santa Monica, CA: Human Factors Society.

Montemarlo, M. D., and Cron, A. C., eds.

1982 Space Human Factors Workshop, Volumes 1 and 2. NASA/Office of Aeronautics and Space Technology OAST, Code RTH-6, Washington, DC.

Moray, N.

1979 Preface. Pp. V-VIII in N. Moray, ed., Mental Workload: Its Theory and Measurement. New York: Plenum.

Muckler, F. A.

1982 Evaluating productivity. In M. D. Dunnett and E. A. Fleishman, eds., Human Performance and Productivity, Vol. 1, Human Capability Assessment. Hillsdale N.J.: L. Erlbaum Associates.

NASA-ASEE

1985 Autonomy and human element in space. Proceedings of the 1983 NASA/ASEE Summer Faculty Workshop at Stanford University.

National Commission on Space

1986 _Pioneering the Space Frontier_. New York: Bantam Books.

Nickerson, R. S.

1986 _Using Computers: Human Factors in Information Systems_. MIT Press (Bradford Books).

Nickerson, R. S. and others

1984 _Research Needs on the Interaction Between Information Systems and their Users: Report of a Workshop_. Washington, DC: National Academy Press.

Norman, G. R., McFarlane, A. G., and Streiner, D. L.

1985 Patterns of illness among individuals reporting high and low stress. _Canadian Journal of Psychiatry_ 30(6):400-405.

Ogden, G. D., Levine, J. M., and Eisner, E. J.

1979 Measurement of workload of secondary tasks. _Human Factors_ 21:529-548.

Parks, D.

1979 Current workload methods and emerging challenges. In N. Moray, ed., _Mental Workload: Theory and Measurement_. New York: Plenum.

Price, H.E.

1985 The allocation of functions in systems. Human Factors 27:33-45.

Price, H. E., Maisano, R. E., and Van Cott, H. P.

1982 The Allocation of functions in Man-Machine Systems: A Perspective and Literature Review. NUREG-CR-2623, Oak Ridge, TN: Oak Ridge National Laboratories.

Price, H. E., and Pulliam, R.

1983 Control room function allocation - A need for man-computer symbiosis. Proceedings of the 1982 IEEE Computer Forum. Denver Co: Institute of Electrical and Electronic Engineers.

Rehmann, J. T., Stein, E. S., and Rosenberg, B. L.

1983 Subjective pilot workload assessment. Human Factors 25(3):297-307.

Rolfe, J. M.

1971 The secondary task as a measure of mental load. In W. T. Singleton, J. G. Fox, and D. Whitfield, eds., Measurement of Man at Work. London: Taylor and Francis.

Schultz, D. P.

1965 Sensory Restriction: Effects on Behavior. New York: Academic Press.

Schwab, E. C., Nusbaum, H. C., and Pisoni, D. B.

1985 Some effects of training on the perception of synthetic speech. Human Factors 27(4):395-408.

Sells, S. B., and Gunderson, E. K.

1972 A social system approach to long-duration missions. Human Factors in Long-Duration Spaceflight. Washington, DC.: National Academy of Sciences.

Seppala, P., and Salvendy, G.

1985 Impact of depth of menu hierarchy on performance effectiveness in a supervisory task: Computerized flexible manufacturing system. Human Factors 27(6):713-722.

Sharit, J., and Salvendy, G.

1982 Occupational stress: Review and reappraisal. Human Factors 24(2):129-162.

Sheridan, T .B., and Simpson, R. W.

1979 Toward the Definition and Measurement of the Mental Workload of Transport Pilots. Flight Transportation Man-Machine Laboratory, Technical Report No. DOT-OS-70055. Cambridge, MA: MIT.

Simonson, E., and Weiser, P. C., eds.

1976 Psychological Aspects and Physiological Correlates of Work and Fatigue. Springfield, IL: Charles C. Thomas.

Singleton, W. T., Fox, J. G., and Whitfield, D., eds.

1971 Measurement of Man at Work. London: Taylor and Francis.

Small, D. W., and Weldon, L. J.

1983 An experimental comparison of natural and structured query languages. Human Factors 25(3):253-263.

Smith, S.

1969 Studies of small groups in confinement. In J. P. Zubek, ed., Sensory Deprivation: Fifteen Years of Research. New York: Appleton-Century-Crofts.

Smith, R. P.

1981 Boredom: A review. Human Factors 23(3):329-340.

Sutermeister, A.

1976 People and Productivity. 3rd. edition, New York: McGraw-Hill.

Swain, A. D.

1970 Development of a Human Error Rate Data Bank. Sandia Corporation Report SC-R-70-4286, Albuquerque, NM: Sandia Corporation,

1978 Estimating Human Error Rates and their Effects on System Reliability. Sandia Corporation Report SAND77-1240, Albuquerque, NM: Sandia Corporation.

Taylor, F. W.

1947 Scientific Management. New York: Harper Brothers.

Thiel, E. D., and Kurtzman, C. R.

1983 Space Applications of Automation, Robotics and Machine Intelligence Systems (ARAMIS). Phase II, Telepresence. [1-3], SSL Reports #30-83 - #32-83, Space Systems Laboratory and Artificial Intelligence Laboratory. Cambridge, MA: MIT

Tilley, A. J., Wilkinson, R. T., and Drud, M.

1981 Night and day shifts compared in terms of the quality and quantity of sleep recorded in the home and performance measured at work: A pilot study. In A. Reinberg, N. Vieux, and P. Andlauer, eds., Night and Shift Work: Biological and Social Aspects. Oxford: Pergamon Press.

Tilley, A. J., Wilkinson, R. T., Warren, P. S. G., Watson, B., and Drud, M.

1982 The sleep and performance of shift workers. Human Factors 24(6):629-641.

VanAllen, J. A.

1986a Myths and realities of space flight. Science 4754:1075-1076.

1986b Space science, space technology and space station. Scientific American 254(1):32-34.

VanCott, H. P., and Kincaid, R. G., eds.

1972 Human Engineering Guide to Equipment Design. Revised edition. Sponsored by the Joint Army-Navy-Air Force Steering Committee, U.S. Department of Defense. Washington, DC.: U.S. Government Printing Office.

Weiner, E. L.

1975 On simultaneous monitoring and tracking. Journal of Applied Psychology 60:100-105.

Weiner, E. L., Curry, R. E., and Faustian, M. L.

1984 Vigilance and task load: In search of the inverted U. Human Factors 26:215-222.

Welford, A. T., ed.

1974 Man Under Stress. New York: John Wiley and Sons.

Welford, A. T.

1973 Stress and performance. Ergonomics 16:567-580.

Weybrew, B. B.

1961 Human factors and work environment. II. The impact of isolation upon personnel. Journal of Occupational Medicine 3:290-294.

1963 Psychological problems of prolonged marine submergence. Pp. 87-125 in N. Burns, R. Chambers, and E. Hendler, eds., Unusual Environments and Human Behavior. New York: MacMillan.

Wickens, C. D.

1979 Measures of workload, stress, and secondary tasks. In N. Moray, ed., Mental Workload: Its Theory and Measurement. London: Plenum.

Wickens, C. D., and Kessel, C.

1979 The effects of participatory mode and task workload on the detection of dynamic system failures. IEEE Transactions on Systems Man and Cybernetics. SMC-13:24-31.

1980 Processing resource demands of failure detection in dynamic systems. Journal of Experimental Psychology: Human Perception and Performance 6(3):564-577.

Wierwille, W. W., and Connor, S. A.

1983 Evaluation of 20 workload measures using a psychomotor task in a moving-base aircraft simulator. Human Factors 25(1):1-16.

Wierwille, W. W., Rahimi, M., and Casali, J. G.

1985 Evaluation of sixteen measures of mental workload using a simulated flight task emphasizing mediational activity. Human Factors 27:489-502.

Williges, R. C., and Wierwille, W. W.

1979 Behavioral measures of aircrew mental workload. Human Factors 21:549-574.

Wise, J. A.

1985 The Quantitative Modeling of Human Spatial Habitability. NASA Grant NAG 2-346. Moffett Field, CA: NASA-Ames Research Center.

1986 The space station: Human factors and habitability. Human Factors Society Bulletin 29(5):1-3.

Woodson, W. E.

1981 Human Factors Design Handbook. New York: McGraw-Hill.

Young, L. R.

1969 On adaptive manual control. IEEE Transactions of Man-Machine Systems. MMS-10:292-331.

Zubek, J. P.

1973 Behavioral and physiological effects of prolonged sensory and perceptual deprivation -- a review. Pp. 8-83 in Man in Isolation and Confinement. Chicago: Aldine Publishing Co.

Zubek, J. P., and Bayer, S. M., and Milstein, S., and Shephard, J. M.
1969 Behavioral and physiological changes during prolonged immobilization plus perceptual deprivation. Journal of Abnormal Psychology 74:230-236.

DRAFT

NOT FOR DISTRIBUTION ATTRIBUTION OR QUOTATION

SYSTEMS PRODUCTIVITY: PEOPLE AND MACHINES

A COMMENTARY ON THE

NICKERSON PAPER

Robert C. Williges

Virginia Polytechnic Institute & State University

Blacksburg, Virginia

SYSTEMS PRODUCTIVITY: PEOPLE AND MACHINES

A COMMENTARY ON THE NICKERSON PAPER

Nickerson's paper provides an excellent review of human factors implications when considering productivity in the space station. In an attempt to amplify some of his points, I will restrict my comments to the ramifications of productivity as espoused in modern industrial engineering. As a point of departure, I will use the recent text by Sink (1985) on productivity management to discuss topics related to defining, measuring, and improving productivity.

WHAT IS PRODUCTIVITY?

In the most general form, productivity in industrial engineering is defined as a simple <u>ratio</u> of some quantity of output divided by some quantity of input. From a systems point-of-view, input quantities (e.g., labor, capital, energy, materials, etc.) go through some transformation (e.g., manufacturing, information processing, etc.) to yield an output (e.g., goods, services, waste, etc.) as shown in Figure 1. By comparing the output quantity to the input quantity, one can assess system productivity as a simple ratio.

Two implications are readily apparent from this operational definition of productivity. First, productivity is a metric that represents more than just output performance. It is a measure of output performance relative to input resources. Consequently, productivity is but one component of performance and should not be equated with overall performance. Other related system performance components might include efficiency, effectiveness, innovation, quality, profitability, etc. From a human factors point of view, productivity has the potential to serve as one metric for evaluating humans as components in complex space systems.

A second implication of the operational definition of productivity is that the ratio metric is based on some defined unit of analysis. Just as the Bureau of Labor statistics measure of overall national productivity (i.e., Gross National Product, GNP, divided by labor input) is of limited value, an overall measure of space station productivity is limited. Care must be taken to chose a meaningful level of analysis in assessing productivity in space systems. From a human productivity point-of-view, it may be difficult to distinguish productivity from human performance in cognitive tasks until better measures of input resources, cognitive processes, and output measures are available.

Productivity does, however, seem to be a viable metric to evaluate larger units of analysis of space-related missions in which the astronaut is considered one component of the unit of analysis. These larger units of analysis should be considered in terms of the human/machine interface level and above. For example, the human component could be considered in assessing the productivity of a space station or in assessing productivity of working environments such as intravehicular activities (IVA) at workstations, extravehicular activities (EVA) outside the space station, and combined IVA and EVA operations such as telerobotic activities (Gillan et al., 1986). In each case, the ratio metric of productivity includes human components along with hardware and software components, and these productivity assessments can be used to evaluate the relative contributions of various components.

HOW IS PRODUCTIVITY MEASURED?

Traditionally, both the time domain and the number of component factors measured are considered in calculating the productivity ratio. In the time domain, both static and dynamic measures of productivity are used. Static measures are used to calculate the productivity ratio for a particular point in time; whereas, dynamic measures are used to evaluate changes in productivity across a designated time unit. Both measures appear to be useful in evaluating the productivity of the human component in space. Static ratios can be used to assess the relative effect of the astronaut in terms of training investment and performance on a particular space mission. Dynamic productivity indices can be used to evaluate changes in team size, allocation of tasks/functions, and return on investments in automation for space missions.

Both static and dynamic measures of productivity can vary in their level of complexity depending upon the number of components measured. Sink (1985), for example, suggests three levels of complexity determined by the number of factors used to construct the productivity ratio. He refers to partial-factor, multifactor, and total-factor measures. Partial factor measures include only one component class (e.g., mission specialist); multifactor measures include several component classes (e.g., mission specialist and computer interface); and total-factor measures include all component classes (e.g., mission specialist, computer-interface, test equipment, documentations, etc.) included in any particular productivity unit of analysis. Obviously, the simple productivity ratio quickly explodes into a complex, multivariate measurement problem once the unit of analysis and number of factors of measurement increases. Research is needed to build and evaluate complex productivity measurement systems for assessing human components of productivity in space missions.

HOW CAN PRODUCTIVITY BE IMPROVED?

In that productivity is a ratio metric, increased productivity must be considered in terms of both input and output quantities and not merely in terms of improving output. Consequently, productivity improvement can be achieved in five ways, as shown in Table 1, depending upon the relationship of the input and output conditions. Although these conditions are somewhat restricted when considering the human component, all appear to be possible if the unit of productivity analysis includes human, hardware, and software components related to space missions. Mostly, one considers human productivity improvement in terms of human performance improvements as Nickerson suggests in his paper. But the implication of the conditions listed in Table 1 suggests that these potential human performance improvements (in output) must be evaluated relative to the input changes (e.g., increased training, cost of automation, etc.) in order to evaluate the real impact on productivity.

RESEARCH ISSUES

Productivity from an industrial engineering point-of-view provides an important metric for assessing human performance as a systems component in space missions. Human productivity per se needs to be considered in a systems context, and any evaluation of productivity must assess both input and output quantities in order to establish a ratio metric. Two general areas of productivity research in space-related missions appear to warrant increased attention.

Measuring Productivity

Several measurement issues must be addressed before human productivity assessments of space missions can be made. The appropriate units of analysis for productivity measurement must be specified. Criteria for partial-factor, multifactor, and total-factor measures need to be established and verified. Automated human performance assessment schemas (Williges, 1977) need to be constructed which could then be used for embedded performance measurement, evolutionary operation, empirical modeling, multivariate criteria, and realistic data bases from which theoretical extrapolations could be made to the design of a variety of future space-related tasks. Improved productivity measurement models with sophisticated human productivity parameters need to be developed and validated. Many of these measurement issues can be addressed by current multivariate measurement procedures, but each of them will require validation during actual space missions.

Improving Productivity

Most of the research issues presented in the Nickerson paper dealing with performance enhancements can relate to improving human productivity if the antecedent input quantities are evaluated in order to establish appropriate productivity indices. The unit of analysis at the human-machine interface level or above seems to provide the best opportunities for improved productivity given the characteristics of the productivity metric. Research issues raised by Nickerson dealing with workstation design, human input modes, decision aids, and automation are particularly relevant. In fact, many of the remaining topics to be discussed during this symposium are candidate issues that could be evaluated in terms of productivity improvement metrics.

CONCLUSION

Productivity is an often used and abused term. By accepting the rather straightforward operational definition of productivity as a ratio of output quantity divided by input quantity, I believe productivity holds promise as an important component metric of space station performance which include human, hardware, and software parameters. Before such a metric is useful, several productivity measurement and productivity enhancement research issues must be addressed.

REFERENCES

Gillan, D. J., Burns, M. J., Nicodemus, C. L., and Smith, R. L.

1986 The space station: human factors and productivity. Human Factors Society Bulletin 29(11):1-3.

Sink, D. S.

1985 Productivity Management: Planning, Measurement and Evaluation, Control and Improvement. New York: Wiley.

Williges, R. C.

1977 Automation of performance measurement. Pp. 153-168 in Symposium Proceedings of Productivity Enhancement: Personnel Performance Assessment in Navy Systems. San Diego: Navy Personnel Research and Development Center.

TABLE 1 Conditions for Improving Productivity (after Sink, 1985)

Increasing Output

1. Output increases; input decreases
2. Output increases; input remains constant
3. Output increases; input increases at a lower rate

Constant Output

4. Output constant; input decreases

Decreasing Output

5. Output decreases; input decreases at a more rapid rate

WILLIGES

Copy of ORIGINAL

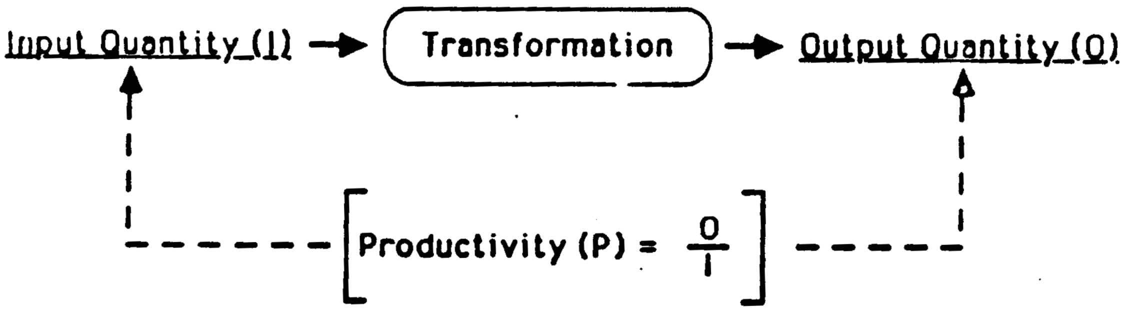

Figure 1. Basic configuration of the productivity metric.

DRAFT
NOT FOR DISTRIBUTION,
ATTRIBUTION
OR QUOTATION

SESSION 1:

SYSTEM PRODUCTIVITY: PEOPLE & MACHINES

SYNOPSIS OF GENERAL AUDIENCE DISCUSSION

Robert Williges

SYSTEM PRODUCTIVITY: PEOPLE & MACHINES

SYNOPSIS OF GENERAL AUDIENCE DISCUSSION

Areas of Concern

Two aspects of the space station provide different concerns for evaluating human productivity. Housekeeping activities may prove to be an important candidate for productivity improvement in terms of reducing the amount of time required to perform these functions. Another major component of the space station is the conduct of scientific activities. Improving productivity related to space research activities appear to be more difficult to measure. In addition to 'on orbit' space station concerns, the integration of ground-control and on-board activities is a prime candidate for productivity improvement studies.

Productivity Metrics

Several of the components related to human productivity in space will be difficult to quantify. Consequently, the accuracy and viability of these measures may be somewhat questionable at certain units of analysis. This underscores the appropriate choice of the unit of analysis. In addition, qualitative measures may need to be substituted for quantitative measures in certain instances.

Lessons Learned

Analysis of other isolated, long duration missions such as early warning systems and sea lab may be useful in making assumptions and generating initial models of key parameters related to productivity for space-related missions. For example, isolations may be a catalyst to trigger stress factors affecting productivity. Caution, needs to be exercised in extrapolating from these analogs, because clear differences exist. Nonetheless, evaluation of these related systems may be useful in isolating a common thread of critical variables affecting human productivity.

SESSION 2:

EXPERT SYSTEMS AND THEIR USE

Paper: Thomas Mitchell, Carnegie-Mellon University

Paper: Bruce Buchanan, Stanford

Discussant: Allen Newell, Carnegie-Mellon

DRAFT

NOT FOR DISTRIBUTIC...

ATTRIBUTION

OR QUOTATION

AI SYSTEMS IN THE SPACE STATION

Tom M. Mitchell

Computer Science Department

Carnegie-Mellon University

Pittsburgh, Pennsylvania

CONTENTS

DRAFT

NOT FOR DISTRIBUTION, ATTRIBUTION OR QUOTATION

[p. 156 DOES NOT EXIST]

PRECEDING PAGE BLANK NOT FILMED

INTRODUCTION

Among the technologies that will help shape life in the space station, Artificial Intelligence (AI) seems certain to play a major role. The striking complexity of the station, its life support systems, and the manufacturing and scientific apparatus that it will house require that a good share of its supervision, maintenance, and control be done by computer. At the same time, the need for intelligent communication and shared responsibility between such computer programs and space station residents poses a serious challenge to present interfaces between man and machine. Hence, the potential and need for contributions from AI to the space station effort is great.

The purpose of this paper is to suggest areas in which support for new AI research might be expected to produce a significant impact on future space station technology. Given the breadth of this task, the approach here will be to sample a few such areas and to rely on the other symposium participants and other sources (e.g., Technical Report NASA-ASEE, 1983; Technical Report NASA , 1985) to fill in the picture. More specifically, we will address here (1) the use of knowledge-based systems for monitoring and controlling the space station, and (2) issues related to sharing and transferring responsibility between computers and space station residents.

Before focussing on the specifics of these two problem areas, it is useful to understand their significance to the development of the space station (and to other advanced projects such as development of a lunar base and interplanetary probes).

In his keynote address to this symposium, Allen Newell provides an analysis of the general characteristics and constraints that define the space station effort. Those of particular relevance to this paper include the following:

- The station is an extraordinarily complex system with an extremely high premium to be placed on reliability, redundancy, and failsafe operation. In past space efforts, a large share of astronaut training has gone into acquiring the knowledge needed to supervise, control, and troubleshoot various spacecraft subsystems. The increased complexity of the space station argues for computer-based assistance in the supervision of many station subsystems, and it is no surprise that the history of the space program is a history of increasing automation and computer supervision. Furthermore, the high premium on failsafe operation places strong demands on the flexibility and adaptability of such computer-based supervisors. Such systems must be flexible enough to recognize and adapt to unanticipated events, and to communicate such unanticipated events clearly to the humans who help choose the response to these events. The flexibility demanded here goes well beyond that associated with present-day computer based supervisory systems.

- The space station is intended to be a highly evolutionary system, which will be continually reconfigured and upgraded over the course of its lifetime in space. The highly evolutionary nature of the station will make the task of crew training even more difficult than if the station were a static system. The problem of updating operating and troubleshooting procedures will be greatly exacerbated. In general, there will be greater demands on maintaining and updating the external documentation of the space station subsystems, and on prompt, thorough updating of procedures for monitoring, controlling, and troubleshooting the evolving space station. Computer-based methods for automatically updating such procedures, given updates to the description of the space station, would greatly enhance the ability to manage the evolving station.

- The crew of the space station will possess differing levels of expertise regarding different space station subsystems, and will live in the station long enough that their expertise will change over the course of their stay aboard the station. These differences in level of sophistication among various crew members (and between the same crew member at differing times) pose significant problems and opportunities for the computer systems with which they will interact. For naive users, computer systems that recommend given actions will have to provide a fairly detailed explanation of the reasoning behind the recommendation. For more expert users, less explanation may be needed. For advanced users, there will be an opportunity for the computer system to acquire new problem-solving tactics from the users. Furthermore, as a particular user becomes familiar with the competence and limitations of a particular computer-based supervisor, his willingness to allow the system to make various decisions without human approval may well change. The ability to interface effectively with a range of users, acting as a kind of tutor for some and acquiring new expertise from others, would allow the computer to act as the "corporate memory" for the particular aspect of the space station that is its domain and for which it will house a continually evolving set of expertise.

MONITORING, DIAGNOSING, AND CONTROLLING THE SPACE STATION

Given the above characteristics of the space station effort, it is clear that the use of computer-based assistants for supervising various space station subsystems could have a major impact on the overall reliability and cost of space station operations. In order to develop such computer-based supervisors, basic research is needed in a number of areas such as representing and reasoning about complex designed artifacts, inferring the behavior of such systems from schematics showing their structure, and automatic refinement of supervisory procedures based on empirical observation as well as the known system schematics.

Since the space station will itself be a large, well-documented artifact, it is reasonable to expect a significant number of opportunities for applying computers to the task of supervising, controlling and diagnosing the space station. For example, one might well expect that a computer could monitor various space station subsystems such as the parts of the navigation system, to detect behavior outside their expected operating ranges, take remedial actions to contain the effects of observed errors, diagnose the likely causes of the observed symptoms, and reconfigure the system to eliminate the error. Of course, limited applications of computers to this kind of problem are fairly common in current-day space systems. But present methods for automated monitoring, diagnosis and control are far from the levels of generality, robustness, maintainability, and competence that one would desire. AI offers a new approach to the problem of automated supervision. With appropriate research support, NASA might expect to significantly accelerate the development of AI methods for dealing with this class of problems, and thereby provide important new technology to support the space station.

A number of recent AI systems have addressed problems of monitoring, diagnosing, or controlling designed artifacts such as computer systems (Ennis et al., 1986), electro-mechanical systems (Pazzani, 1986), chemical processes (Scarl et al., 1985), and digital circuits (Davis, 1984; Genesereth, 1981). From this work, an initial set of techniques has emerged for building computer programs that embody a model (often in qualitative terms) of the behavior of the system under study, and which use this model to reason about the diagnosis, control, or reconfiguration of the system. While much remains to be understood, the initial approaches have shown clearly the potential for supervisory computer systems that combine judgemental heuristics with reasoning from a concrete model of the systems under study.

An Example

As an example of an AI system that deals with monitoring and troubleshooting a designed artifact, consider Davis' circuit troubleshooting system (Davis, 1984). This system troubleshoots digital circuits, given a schematic of the misbehaving circuit together with detected discrepancies between predicted and observed signal values. Its organization is typical of several troubleshooting systems that have been developed for electronic, mechanical, and other types of systems.

The basic idea behind this troubleshooting system is that it uses the schematic of the system, together with its knowledge of the expected behaviors of system components, in order to reason backward from observed incorrect output signals to those upstream circuit components that could have produced the observed error. This process is illustrated in Figure 1, taken from Davis (1984).

In this figure, if the circuit inputs are given as shown, the system will infer the expected outputs as shown in round parentheses, based on its knowledge of the behaviors of multipliers and adders. If the two observed outputs are as shown in square parentheses, then a discrepancy is found between the expected and observed values for signal F. The system will then enumerate candidate fault hypotheses by considering that the error may be due to a failure in Add-1, or to incorrect values for one of its inputs (either X or Y). Each of these last two hypotheses might be explained further in terms of possible failures of the components or signals on which it, in turn, depends. Thus, candidate fault hypotheses are enumerated by examining the structure of the circuit as well as the known behaviors of its components.

In addition to enumerating fault hypotheses in this fashion, the system can also prune these hypotheses by determining other anticipated consequences of presumed faults. For example, the hypothesis that the error in signal F is caused by an error in signal Y, carries with it certain implications about the value of signal G. The value of 10 for signal F can be explained by a value of 4 for signal Y, but this would in turn lead to an expected value of 10 for signal G (which is observed to hold the value 12). Hence, this hypothesis may be pruned, as long as one assumes that the circuit contains only a single fault.

The above example illustrates how a computer system can reason about possible causes of observed faults, by using knowledge of the schematic of the faulty system as well as a library describing the expected behaviors of its components. There are many subtleties that have been glossed over in this example, such as reasoning about the possibility of multiple system faults, interactions between faults, intermittent errors, utilizing statistical knowledge of likely faults and the resulting faulty behavior, scaling this approach to more complex systems, and the like. Basic research is still needed to develop more realistic diagnostic systems of this sort, and many of these issues are under study at this time. In addition, a good deal of research has been devoted to developing similar troubleshooting systems for artifacts other than digital circuits (e.g., mechanical electromechanical, and chemical processes). The topic of reasoning about the expected behavior of designed artifacts of many types is an active research area within AI (see, for example, the recent special volume of Artificial Intelligence on qualitative reasoning about physical systems (North-Holland, 1984).)

"Hands-On" Supervisory Systems

The above example is meant to suggest how a program can utilize an internal model of the system it is monitoring in order to localize the cause of anomalous behavior. Since the space station will be heavily instrumented with sensors and with computer-controlled effectors, the real opportunity here lies in developing a technology for "hands-on" AI supervisory systems: systems that have the means to directly observe and control the behavior of systems that they monitor, and that possess an explicit model of the system under supervision to guide their reasoning about monitoring, controlling, and troubleshooting this system. Figure 2 illustrates the general organization of such a hands-on supervisory system.[1]

One instantiation of the scenario characterized in the figure could be an electronically self-sensing, self-monitoring space station. Here the system under supervision is the space station, sensors may observe the temperatures, pressures, and electrical behavior of various subsystems of the space station, and effectors may correspond to electrically controlled devices such as signal generators, heaters, compressors, and alarm systems. The goal of such an intelligent, self-monitoring space station would be to observe its behavior through its sensors, comparing these observations to the behavior anticipated by its internal model, and utilizing its effectors to maintain stable operation, reconfigure subsystems, and control the trajectory of states of the system. A number of observations are apparent about such a system: To a limited degree it is already possible to build such partially self-monitoring systems. The theoretical possibilities for computer monitoring and control in such systems far exceed the capabilities of our present techniques. The effectiveness of such a system will depend on continuing fundamental research in AI, especially in areas such as qualitative reasoning, diagnosis, control, and learning. To allow for such a future, the initial design of the space station must allow for flexible introduction of new sensors and effectors in all subsystems of the space station, and over the entire life of the station.

A very different instantiation of the scenario of Figure 2 is obtained by introducing mobility in the sensors and effectors of the computer monitor. In this case, the supervisor could take the form of a collection of mobile platforms whose sensors include cameras, range finders, touch sensors, and oscilloscope probes, and whose effectors include wheels, rocket engines, manipulators, signal generators, and arc welders. Such a system might be expected to monitor the physical plant of the space station, checking for wear, and repairing the station as necessary, both interior and exterior. Several observations follow from considering this scenario: The leverage gained by adding mobility to sensors and effectors is large -- especially in situations such as troubleshooting where the system parameters in question might not be directly observable or controllable by statically positioned sensors and effectors. A number of difficult issues arise in representing and reasoning about three dimensional space, navigation, and the mechanics of physical systems. Given previous experience with robotics, it is clear that the difficulty of the technical problems can be considerably eased by designing a well-engineered work environment (e.g., by including easy grasping points on objects that are to be manipulated) in the space station.

In fact, we would like our supervisor to possess a combination of mobile and stationary sensors and effectors, including the union of those in the above scenarios. Thus, these two scenarios illustrate different aspects of the class of hands-on supervisor problems summarized in Figure 2. The two scenarios suggest a number of common technical problems, including problems of integrating human judgement with computer judgement, planning a sequence of control operations based on only an incomplete model of the system under supervision, and utilizing sensory input to refine the model of the system under supervision. At the same time, each scenario carries its own technical problems which overlay those generic issues. For example, a mobile supervisor for monitoring and repairing the exterior surface of the space station must face issues such as representing and reasoning about three dimensional space and navigation, interpreting a rich set of perceptual data taken from a changing (and incompletely known) vantage point, and using tools to manipulate the space station. Thus, NASA should consider supporting research on the generic problems of hands-on supervisory systems, as well as research on selected instances of the problem which it expects would yield significant practical gains.

Nature of the Problem

A fundamental defining characteristic of the system supervisor problem is uncertainty in the supervisor's knowledge of the system under study. A supervisor can almost never have complete and certain knowledge of the exact state of the system, of the rules that determine how one system state will give rise to the next, or of the exact effects of its control actions on the system. This characteristic alters dramatically the nature of diagnostic and control tasks. For example, given a perfect model of the system under study, a program might derive an open-loop control sequence to place the system in some desired state. However, in the absence of a perfect model, controlling the system requires interleaving effector actions with sensory observations to detect features of the system state.

The types and degrees of uncertainties faced in system supervision problems vary, of course, with the specific task. For instance, the task of monitoring a digital circuit might correspond to an extreme point in the spectrum of possibilities, since circuits schematics do, in fact, provide a very detailed model of the system, and since observing digital signal values is (by design) a relatively unambiguous task. It is probably no accident that several of the earliest attempts to construct AI troubleshooting aids were conducted in the domain of digital circuitry. However, that work showed that even in this domain it was very difficult to troubleshoot circuits based only on the knowledge available from the circuit schematic (Davis, 1984). The problem is that circuit behavior can depend on thermal effects, physical proximity of components, and other factors which are not typically reflected in a circuit schematic. Furthermore, it is precisely in troubleshooting situations that such effects become significant to determining the system's behavior. The problem of incomplete knowledge in modeling subsystem behaviors is even more difficult when one considers systems with combinations of electrical, mechanical, chemical, and biological subsystems.

In addition to uncertainty in modeling the expected behavior of the system under study, the difficulty of interpreting sensory input adds another kind of uncertainty in many domains. In the digital circuit world, it is fairly straightforward to observe the value of a desired signal, though it is rare that circuits are constructed so that every signal is brought outside the circuit for troubleshooting purposes. If the system under study is a chemical process rather than electrical, detecting relative concentrations of chemicals can often be a more complex task. In mechanical systems, detecting exact locations and forces is generally out of the question. If the system is the exterior of the space station and the sensors are video cameras, then the difficulty of sensing the exact location and physical condition of each subcomponent can itself become such an overwhelming task that the observations themselves must be treated as uncertain.

Yet another dimension of uncertainty arises from the effectors that are utilized by the supervisor to alter the system under study. Again, in the circuit domain effectors such as signal generators are relatively reliable. But in the robotics domain, in which the system being supervised is the physical world, effectors such as artificial limbs may be fairly unreliable in executing actions such as grasping. In such cases, the problem of planning a sequence of actions to bring the system to a desired state must take into account nondeterminism in the effect of actions it performs.

In a sense, the ability to observe and affect the system under study and the ability to predict its behavior provide redundant sources of knowledge so that one can be used to make up for uncertainty in the other. For instance, feedback control methods utilize sensory information to make up for an incomplete model of the next-state function. On the other hand, one can make due with observing only a small proportion of the signal values in a circuit and use the model of subcomponent behaviors to infer additional signal values upstream and downstream of observed signals.

Given the various uncertainties that must be faced by a supervisory system, it is unlikely that purely algorithmic methods can be mapped out for dealing with all eventualities (although the vast NASA troubleshooting manuals indicate the degree to which this might be possible). A supervisory system will do best if it possesses redundancy to make up for the uncertainties that it must face: redundancy in the sensors that give it information about the world, in the effectors with which it controls the world, and in the behavioral models that it uses for reasoning about the system under study. While such redundancy can help reduce uncertainty, it will not be eliminated, and the supervisor must therefore employ problem solving methods designed to operate under incomplete information. All of these needs suggest the importance of combining heuristic methods with deductive methods for reasoning about the system under study. Finally, these same problem characteristics that suggest the utility of employing AI methods (the need for flexibility in solving problems despite uncertainty) also suggest the importance of including humans in the problem-solving process. Even by optimistic estimates, it seems unlikely that AI systems will be able to completely replace human judgement in many supervisory tasks, though they may well augment it in many tasks. Thus, in many cases we envision cooperative problem solving involving computer systems and humans. Section "Sharing and Transferring Expertise in Man-Machine Problem Solving" discusses issues related to man-machine cooperation in this regard.

Research Recommendations

What research should be supported by NASA in order to maximize the future availability of hands-on supervisory systems of the kind described above? This section lists some areas that seem especially important, though the list is certainly not intended to be complete.[2]

- Modeling system behavior at multiple levels of abstraction. At the heart of the ability to supervise a system lies the ability to model its behavior. Systems theory provides one body of (primarily quantitative) techniques for describing and reasoning about systems. AI has developed more symbolic methods for describing and reasoning about systems, given a description of their parts structure. A good deal of research is needed to further develop appropriate behavior representations for a variety of systems at a variety of levels of abstraction, and for inferring behavioral descriptions from structural descriptions. In addition, work is needed on automatically selecting from among a set of alternative models the one most appropriate for the task at hand. For example, one useful research task might be to develop a program which can be given a detailed schematic of a large system (e.g., a computer) as well as a particular diagnostic problem (e.g., the printer is producing no output), and which returns an abstract description of the system which is appropriate for troubleshooting this problem (e.g., an abstracted block diagram of the computer focussing on details relevant to this diagnostic task).

- Planning with incomplete knowledge. The planning problem is the problem of determining a sequence of effector actions which will take the external system to a desired state. This problem has been studied intensely within AI, especially as it relates to planning robot actions in the physical world. However, current planning methods make unrealistic assumptions about the completeness of the robot's knowledge of its world, and of its knowledge of the effects of its own actions. New research is needed to develop planning methods that are robust with respect to uncertainties of the kinds discussed above. One useful research task here would be to develop methods that produce plans which include sensor operations to reduce anticipated uncertainties in the results of effector actions, and that include conditional branches in the plan to allow for "run-time" decisions based on sensory actions.

- Integrating methods from control theory with symbolic control methods. Problems of system control, diagnosis (identification), and monitoring have been studied for some time in fields such as system control theory. Such studies typically assume a quantitative, mathematical model of the system under supervision, whereas AI methods model the system in a symbolic, logical formalism. System theory has developed various methods for using sensory feedback to make up for uncertainty in the model of the system under supervision, but these methods are difficult to apply to complex planning problems such as determining a sequence of robot operations to repair a failed door latch. Still, both fields are addressing the same abstract problems. Very little attention has been paid to integrating these two bodies of work, and research on both vertical and horizontal integration of these techniques should be supported.

- Automatically refining the supervisor's theory of system behavior through experience. As discussed in the previous subsection, a major limitation on the effectiveness of a supervisor lies in its uncertain knowledge of the system under supervision. Therefore, methods for automatically refining the supervisor's knowledge of the system would be extremely useful. In AI, research on machine learning and automated theory formation should be supported as it applies to this problem. The integration of this work with work in systems theory on model identification should also be explored. Possible research tasks in this area include developing robot systems that build up maps of their physical environment, and systems that begin with a general competence in some area (e.g., general-purpose methods for grasping tools) and which acquire with experience more special purpose competence with experience (e.g., special methods for most effectively manipulating individual tools).

- Perception from multiple sensors. One method for reducing uncertainty in the supervisor's knowledge of the system's state is to allow it to use multiple, redundant sensors. Thus, a robot might use several video cameras with overlapping fields of view, placed at different vantage points, together with touch sensors, range finders, infrared sensors, etc. Or a supervisor for monitoring a power supply system might utilize a set of overlapping voltage and current sensors together with chemical sensors, heat sensors, etc. The benefits of using multiple sensors is clear -- they provide more information. However, in order to make use of the increasing amounts of data available from multiple sensors, research is needed to develop more effective sensory interpretation/perception methods for individual sensors, and for fusing data from several sensors. An example research task here might be to develop a system that employs a number of video cameras, and which determines the correspondence between image features of the various images. A more ambitious project might try to predict image features likely to be found by one camera, based on information from other touch, video, and heat sensors.

- Representing and reasoning about 3D geometric properties. For supervisors that possess mobile sensors or effectors, a variety of problems exist in reasoning about navigating through space, and in reasoning about 3D mechanical linkages such as those that couple a robot arm to a screw via a screw driver. Research is needed on representing 3D objects (including empty space) in ways that allow for efficient computation of relations among objects, such as intersections (collisions), unions, possible packings, etc. Furthermore, since manipulating the world involves constructing temporary mechanical linkages among objects (e.g., among a robot arm, screw driver, screw, and wall), research is needed on efficiently representing and reasoning about such linkages so that effector commands can be planned that will achieve desired effects. While special-purpose robots operating in special-purpose environments can sometimes avoid using general methods for reasoning about 3D geometry, general purpose systems expected to solve unanticipated problems will require this capability.

o Designing systems to minimize difficulty in observing and controlling them. Given the great difficulties in the supervisory task that are introduced by uncertainty, one obvious reaction is to try to design the space station to reduce the uncertainties that automated supervisors will face. In short, the station should be designed to maximize the observability and controllability of those features which the supervisor will need to sense and effect. In the case of a supervisor with immobile sensors and effectors, such as a system to monitor the power supply, this requires that a broad and redundant set of sensors and control points be built into the power supply at design time. In the case of mobile supervisors, the observability of the station can be engineered, for example, by painting identifying marks on objects which will ease problems of object identification and of registering images obtained from multiple viewpoints. Similarly, the controllability of the physical space station can be enhanced, for example, by designing all its parts to present the same simple grasping point. While a good deal of anecdotal experience has been obtained on designing robot workstations to maximize their controllability and observability, little exists in the way of a science for designing such easily-supervised systems. Research in this area, if successful, could significantly reduce the number of technical problems that automated supervisors in the space station will face.

- Feasibility of replacing hardware subsystems by software emulations. For immobile supervisors which monitor subsystems such as power supplies, navigation systems, etc., one intriguing possibility is that they might be able to substitute additional computation in place of failed hardware. For example, consider a subsystem, S, with a failed thermostat, T1. If S is being supervised by a computer system with a good model of the subcomponents of S, then this supervisor might be able to keep S working acceptably by substituting its own simulated output of T1 for the output of the failed thermostat. The degree to which this is possible will depend, of course, on (1) the veracity of the supervisor's model of S, (2) the access the supervisor has to other sensors in S (the more redundant, the better), and (3) the ability of the supervisor to control the point in S corresponding to the output of T1. While a software simulation might be slower and less accurate than a working thermostat, the advantage of substituting software for failed hardware is clear. Perhaps a small number of high-speed processors (such as parallel processors that have been developed for circuit simulations) could be included in the space station precisely for providing high-speed backup for a wide range of possible hardware failures. While the feasibility of adding robustness to the space station by adding such computational power is unproven, the potential impact warrants research in this direction.

SHARING AND TRANSFERRING EXPERTISE IN MAN-MACHINE PROBLEM SOLVING

As noted in the previous section, the same problem characteristics that argue for flexibility and adaptability in computer supervisory systems also argue for allowing humans to participate in problem solving and decision making processes. As the complexity of computer support for the space station grows, the need for communication and shared responsibility between the computer and space station residents will grow as well. If ever we reach the stage of a fully automated, self-supporting space station, we are likely to first spend a significant period of time in which computer assistants will provide certain fully-automated services (e.g., simply monitoring station subsystems to watch for unexpected behavior), but will require interaction with their human counterparts in responding to many novel events. Effective methods for such man-machine interaction will encourage the introduction of computer assistants for many more tasks than possible if totally automated operation were demanded. This section considers some of the research issues related to developing effective communication between AI systems and their users.

Since several other symposium participants will address the issue of man-machine communication in general, I will try to focus this section on issues specific to sharing problem solving responsibilities and to transferring expertise from humans to their computer assistants.

Shared responsibility is a desirable characteristic whenever one is faced with a multifaceted task for which humans are best suited to some facets and machines to others. Humans use mechanical tools (e.g., wrenches) and computational tools (e.g., pocket calculators) for exactly such reasons. In the space station, we may find it desirable to share responsibility in motor tasks, as in a human controlling the mechanical robot arm in the space shuttle, in cognitive tasks, as in a human and computer system working jointly to troubleshoot a failed power supply, or in perceptual tasks, in which a human may assist the computer in finding corresponding points in multiple camera images so that the computer can then apply image analysis and enhancement procedures to the images. In each case, shared responsibility makes sense because the machine has certain advantages for some aspects of the task (e.g., physical strength and the ability to operate in adverse environments) while the human possesses advantages for other aspects (e.g., motor skills and flexibility in dealing with the unanticipated).

Sharing in the process of problem solving also raises the prospects for transfer of expertise. In many fields, humans learn a great deal by acting as an apprentice to help a more advanced expert solve problems. As the medical intern assists in various hospital procedures, he acquires the expertise that eventually allows him to solve the same problems as the doctor to whom he has apprenticed. One recent development in AI is a growing interest in constructing interactive problem solving systems that assist in solving problems, and that attempt to acquire new expertise by observing and analyzing the steps contributed by their users. This section argues that research toward such learning apprentice systems is an important area for NASA support.

An Example

In order to ground the discussion of shared responsibility and learning apprentices, we briefly summarize a particular knowledge-based consultant system designed to interact with its users to solve problems in the design of digital circuits. This system, called LEAP (Mitchell et al., 1985), is a prototype system which illustrates a number of difficulties and opportunities associated with shared responsibility for problem solving.

LEAP helps to design digital circuits. Users begin a session by entering the definition of some input/output function that they would like a circuit to perform (e.g., multiply two numbers). LEAP provides assistance in designing the desired circuit, by utilizing a set of if-then rules which relate desired functional characteristics to classes of circuit implementations. For instance, one rule in this set dictates that "IF the desired function requires converting an input serial signal to an equivalent parallel signal, THEN one may use a shift register." LEAP utilizes these rules[3] to suggest plausible refinements to the abstract circuit modules that characterize the partial design at any given stage.

Figure 3 depicts the interface to LEAP as seen by the user. The large window on the right contains the circuit abstraction which is presently being designed by the user/system. As shown in the figure, the circuit consists at this point of two abstract circuit modules. For each of these circuit modules, LEAP possesses a description of the function to be implemented. At any point during the design, the user selects one of the unimplemented circuit modules to be considered, and LEAP examines its rule set to determine whether any rules apply to this module (i.e., rules whose preconditions match to the specifications of the circuit module). If LEAP determines that some of its rules apply to this situation, it presents the recommendations associated with these rules to the user. The user can then examine these options, select one if he wishes, and LEAP will refine the design accordingly. Figure 4 depicts the result of such an implementation step. Should the user decide that he does not want to follow the system's advice, but instead wishes to design this portion of the circuit manually, he can undo the rule-generated refinement and use LEAP as a simple, graphics-oriented, circuit editor.

LEAP provides a simple example of shared problem solving between man and machine. The user directs the focus of attention by selecting which circuit module to refine next. LEAP suggests possible implementations of this module, and the user either approves the recommendations or replaces them with his own. LEAP thus acts as an apprentice for design. For design problems to which its rule base is well-suited, it provides useful advice. For circuits completely outside the scope of its knowledge it reduces to a standard circuit editing package, leaving the bulk of the work to the human user. As the knowledge base of LEAP grows over time, one would expect it to gradually take on an increasing share of the responsibility for solving design problems.

LEAP also illustrates how such knowledge-based apprentices might learn from their users (Mitchell et al., 1985). In particular, LEAP has a primitive capability to infer new rules of design by observing and generalizing on the design steps contributed by its users. In those cases where the user rejects the system's advice and designs the circuit submodule himself, LEAP collects a training example of some new rule. That is, LEAP records the circuit function that was desired, along with the user-supplied circuit for implementing that function. LEAP can then analyze this circuit, verify that it correctly implements the desired function, and formulate a generalized rule that will allow it to recommend this circuit in similar subsequent situations. The key to LEAP's ability to learn general rules from specific examples lies in its starting knowledge of circuit operation. Although it may not initially have the expertise to generate a particular implementation of the desired function, it does have the ability to recognize, or verify, the correctness of many of its users' solutions. In general, it is easier to recognize a solution than to generate one. But once a solution can be recognized and explained, then LEAP can generalize on it by distinguishing that certain features of the example are critical (those mentioned in the verification), whereas others are not (those not mentioned in the verification).

LEAP is still a research prototype system, and has not yet been subjected to testing on a large user community. While there are no doubt many technical issues still to be solved, it serves as a suggestive example of how a knowledge-based consultant might be useful as an apprentice even before its knowledge base has been fully developed. It also suggests how its interaction with the user might lead it to extend its knowledge base automatically. The methods for collecting training examples and for formulating general rules appear generic enough that similar learning apprentice systems might be developed for many supervisory tasks of the kind discussed in the previous section. Other current research is exploring the feasibility of such learning apprentices in task domains such as signal interpretation (Smith et al. 1985), proving mathematical theorems (O'Rorke, 1984), and planning simple robot assembly steps (Segre and DeJong, 1985).

Nature of the Problem

The LEAP system suggests one kind of shared responsibility between computer and human, as well as a mechanism for the gradual accretion of knowledge by the system so that over time it can take on a progressively greater share of responsibility for problem solving. The ability to acquire new rules by generalizing from the users's actions follows from LEAP's starting knowledge of how circuits work. That is, it begins with enough knowledge of how circuits operate, that it is able to explain, or verify, the appropriateness of the users' actions once it observes them. Once it has verified that the user's circuit correctly implements the desired function, then it can generalize on this action by retaining only those features of the specific situation that are mentioned in this explanation. Similarly, if one tried to construct such a learning apprentice for troubleshooting power supply faults, one would want to include sufficient initial knowledge about the power supply (i.e., its schematic) that the system could verify (and thus generalize on) users's hypotheses about the causes of specific power supply malfunctions.

Thus, in order for a system to learn from observing its users, it must begin with sufficient knowledge that it can justify what it observes the user do. It seems that for supervisory tasks of the kind discussed above, the primary knowledge required[4] to construct such explanations is a description of the structure and operation of the system under supervision. Since AI has developed methods for representing such knowledge, supervisory tasks seem like good targets for further research on learning apprentices.

In addition to cognitive tasks such as monitoring, designing, and debugging, one might consider learning apprentices for robotics tasks such as using tools (see Segre and DeJong, 1985 for one example). Given a new tool for the robot to use, one way to train it might be to use a teleoperator to guide the robot through several uses of the tool. For example, given a new type of fastener, a user might guide the robot to grasp the fastener and use it to fasten two objects together. If the system could start with enough knowledge to explain which features of its trajectory and other motions were relevant to accomplishing the given task, then it might be able to generalize accordingly. Research on such robotic learning apprentices seems worthwhile and highly relevant to the goals of the space station program.

To understand the issues involved in sharing information and responsibility between human and machine, it is instructive to consider the issues involved in sharing responsibility strictly among humans. In both cases there are certain subproblems that are best dealt with by individual agents, and others where shared responsibility makes best sense. Successful interaction requires arriving at an agreement on which agent will perform which task. In LEAP, the user makes all such choices. But in more complex scenarios the user may not want to spend the time to approve every suggestion of the apprentice. In such cases, there must be ways to agree upon a policy to determine which decisions are worth having the human approve. Of course there are many other issues that follow from this analogy as well: the cooperating agents eventually need accurate models of their relative competence at various subtasks. And there will be questions of social and legal responsibilities for actions taken.

Here we have tried to suggest that one class of computer assistants on the space station be viewed as dynamic systems that interact with their users and work toward extending their knowledge and competence at the task they perform. Preliminary results from AI suggest that this is a worthwhile research task. The nature of the space station suggests that such self-refining systems are exactly what will be needed. The continually changing configuration of the station itself, the continually changing crews and types of operations that will be conducted aboard the space station, the evolving technology that will be present, all dictate that the computer assistants aboard must be able to adjust to new problems, new procedures and new problem solving strategies over the life of the space station.

Research Recommendations

Here we suggest several areas in which NASA might support research toward advanced interfaces for interaction between humans and intelligent consultant systems.

- Architectures that support graceful transfer of expertise and responsibility. Research toward developing learning apprentice systems for space station applications is warranted based on recent AI results and on the importance of such systems to the space station program. A prudent research strategy at this point would be to support development of a variety of learning apprentices in various task areas (e.g., for troubleshooting space station subsystems, for monitoring and controlling subsystems, for managing robot manipulation of its environment). Such a research strategy would lead to experimenting with alternative software architectures for learning apprentices, as well as an increased understanding of the feasibility of constructing learning apprentices for specific space station task areas.

- Evolution of grainsize and initiative of interaction. As the expertise of the apprentice grows, and as the human becomes more familiar with the competence and communication capabilities of the computer, one expects that the optimal style of communication should shift. Changes may occur, for example, in who takes the initiative in controlling the direction of problem solving, and in the grainsize of the tasks (e.g., initially small subtasks will be discussed, but later it may be sufficient to focus only on larger grain subtasks). Research on interfaces that support these kinds of changes over time in the nature of the interaction, and which support explicit communication about such issues, should be encouraged. Such flexible interfaces are important whether the apprentice learns or not, since the user will certainly go through a learning period during which his understanding of the system's competence and foibles, and his willingness to trust in the system will change.

- Task-oriented studies of cooperative problem solving. In order to understand the kinds of knowledge that must be communicated during shared problem solving, it may be worthwhile to conduct protocol studies in which a novice human apprentices with an expert to assist him and to acquire his expertise (e.g., at a task such as troubleshooting a piece of equipment). Data collected from such experiments should provide a more precise understanding of the types of knowledge communicated during shared problem solving, and of the knowledge acquisition process that the apprentice goes through.

- Transferring knowledge from machine to man. Given the plans for a frequently changing crew, together with the likely task specialization of computer consultants, it is reasonable to assume that in some cases the computer consultant will possess more knowledge about a particular problem class than the human that it serves. In such cases, we would like the system to communicate its understanding of the problem to the interested but novice user. Certain work in AI has focused on using large knowledge bases as a basis for teaching expertise to humans (e.g., Clancey and Letsinger, 1984). Research advances on this and other methods for communicating machine knowledge to humans would place NASA in a better position for crew training and for integrating intelligent machines into the human space station environment.

SUMMARY

This paper presents a sampling of recommended research directions which NASA may wish to support in order to accelerate the development of AI technology of particular relevance to the space station. We feel that recent AI research indicates the potential for a broad range of applications of AI to space station problems. In order for this potential to become reality, significant support for basic AI research is needed.

Research toward developing a wide range of "hands-on" supervisory systems for monitoring, controlling, troubleshooting and maintaining space station subsystems is strongly recommended. Such research is important both because of its potential impact on reliability and safety of the space station and because the technical development of the field of AI is at a point where a push in this area may yield significant technical advances. Such hands-on supervisory systems could include both physically stationary supervisory systems that monitor electronic subsystems, power supplies, navigation subsystems and the like, as well as physically mobile supervisors that monitor and repair the exterior and interior physical plant of the space station. Important technical challenges remain to be addressed in both areas.

In support of developing and deploying such knowledge-based supervisors, it is recommended that research be conducted leading toward interactive, self-extending knowledge-based systems. Such systems may initially serve as useful apprentices in monitoring and problem solving, but should have a capability to acquire additional knowledge through experience. The evolutionary nature of the space station together with the turnover of crew assure that a continually changing set of problems will confront onboard computer systems. This feature of the space station, together with the need to continually extend the knowledge of problem solvers onboard, argue for the importance of research toward interactive, self-extending knowledge based systems.

There are certainly additional areas of AI research which would also benefit the space station program. The goal of this paper is to point out a few such areas, in the hope of stimulating thought about these and other possible uses of AI in the space station.

ACKNOWLEDGEMENTS

My thanks go to Allen Newell and Oren Etzioni for providing useful comments on earlier drafts of this paper. This work was supported in part by NSF grant DCR-8351523.

NOTES

1. In fact, initial AI systems for troubleshooting and control have generally been restricted to dealing with typed-in observation inputs and to typing out their recommendations rather than exerting direct control over the system. However, there are exceptions to this, such as the YES/MVS system (Ennis et al., 1986) which directly monitors and controls operations of a large computer system.

2. The research recommendations listed here represent solely the opinion of the author, and should not necessarily be interpreted as recommendations from the symposium as a whole.

3. LEAP also utilizes knowledge about behaviors of individual circuit components, plus knowledge of how to symbolically simulate digital circuits.

4. Other relevent knowledge includes the goals of the user (e.g., a decision must be made to act within 15 seconds), and empirical data on the frequencies of various types of faults.

REFERENCES

1984 Artificial Intelligence, Special Volume on Qualitative Reasoning About Physical Systems. North-Holland.

Clancey, W. and Letsinger, R.

1984 NEOMYCIN: reconfiguring a rule-based expert system for application to teaching. Pp. 361-381 in Clancey and Shortliffe, eds., Readings in Medical Artificial Intelligence. Addison-Wesley.

Davis, R.

1984 Diagnostic reasoning based on structure and behavior. Artificial Intelligence 24:347-410.

Ennis, R. L., et al.

1986 A continuousus real-time expert system for computer operations. IBM Journal of Research and Development 30:1.

Genesereth, M.

1981 The Use of Hierarchical Models in the Automated Diagnosis of Computer Systems. Technical report Stanford HPP memo 81-20, Stanford University, Stanford, CA.

Mitchell, T. M., Mahadevan, S., and Steinberg, L.

1985 LEAP: a learning apprentice for VLSI design. Pp. 573-580 in Proceedings of the Ninth International Joint Conference on Artificial Intelligence. August.

National Aeronautics and Space Administration

1983 Autonomy and the Human Element in Space. Technical report NASA-ASEE 1983 summer faculty program final report. National Aeronautics and Space Administration, Stanford, CA.

National Aeronautics and Space Administration

1985 Advancing Automation and Robotics Technology for the SpaceStation and for the U.S. Economy. Technical report NASA technical memorandum 87566. National Aeronautics and Space Administration, Springfield, VA.

O'Rorke, P.

1984 Generalization for explanation-based schema acquisition. Pp. 260-263 in Proceedings of the AAAI. AAAI, Austin, TX.

Pazzani, M. J.

1986 Refining the knowledge base of a diagnostic expert system: An application of failure-driven learning. Pp. 1029-1035 in Proceedings of the Fifth National Conference on Artificial Intelligence. AAAI, August.

Segre, A. M., and DeJong, G. F.

1985 Explanation-based manipulator learning: acquisition of planning ability through observation. Pp. 555-556 in Proceedings of the IEEE Conference on Robotics and Automation. IEEE, St. Louis, MO.

Scarl, E. A., Jamieson, J. R. and Delaune, C. I.

1985 A fault-detection and isolation method applied to liquid oxygen loading for the space shuttle. Pp. 414-416 in Proceedings of 1985 International Joint Conference on Artificial Intelligence. International Joint Conference on Artificial Intelligence, Los Angeles.

Smith, R. G., Winston, H. A., Mitchell, T. M., and Buchanan, B. G.

1985 Representation and use of explicit justifications for knowledge base refinements, Pp. 673-680 in Proceedings of the Ninth International Joint Conference on Artificial Intelligence. August.

MITCHELL

ORIGINAL PAGE IS
OF POOR QUALITY

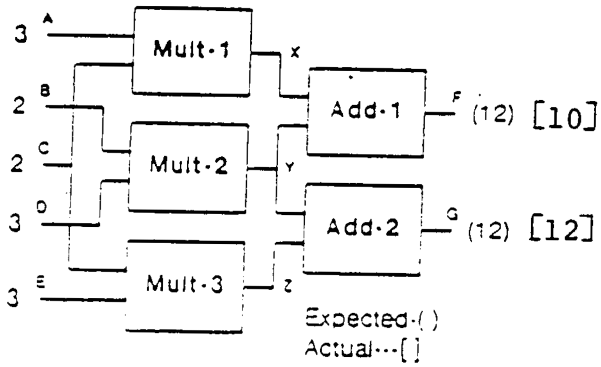

Figure 1: Troubleshooting example taken from (Davis, 1984)

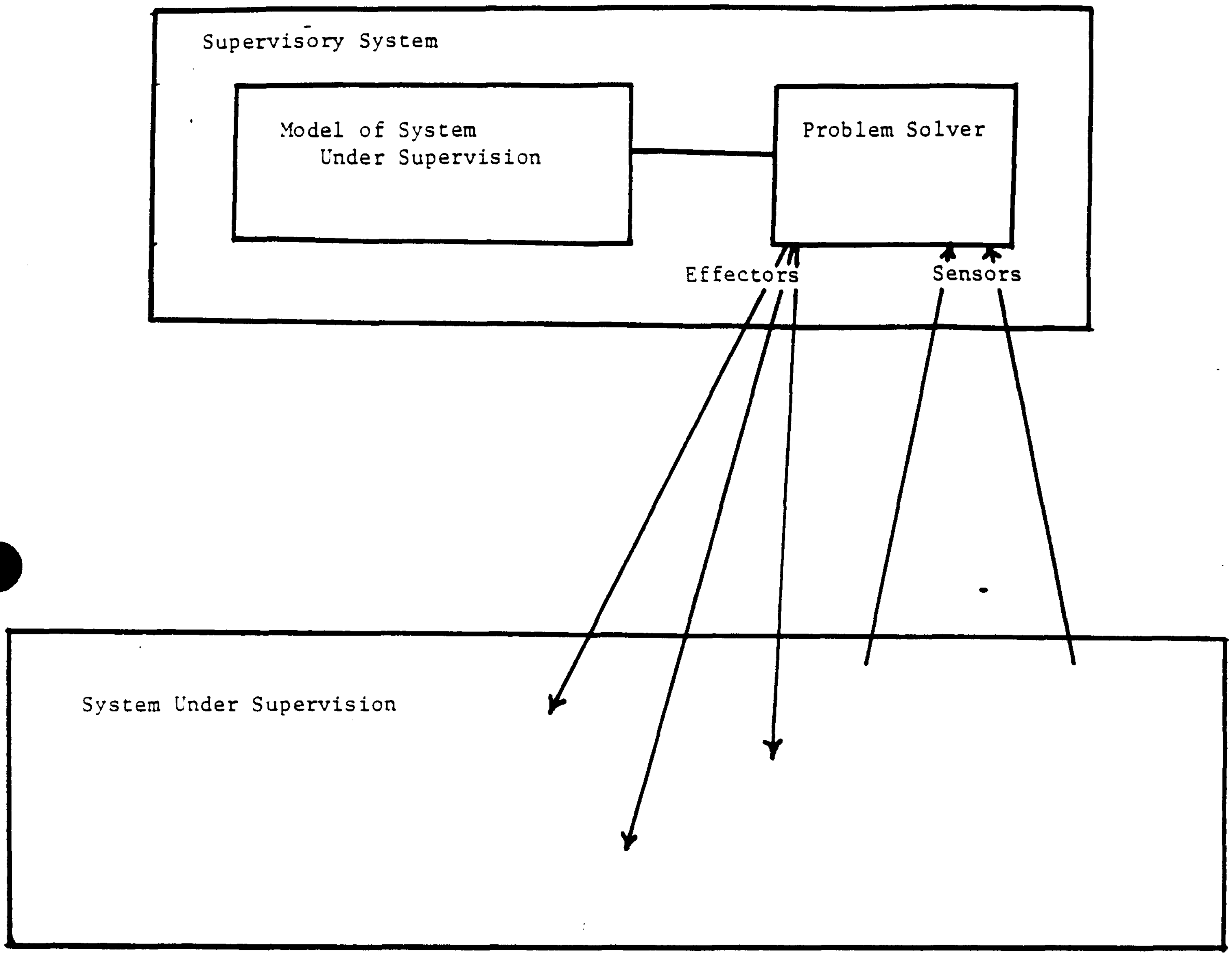

Figure 2: HANDS ON SUPERVISORY SYSTEM

MITCHELL

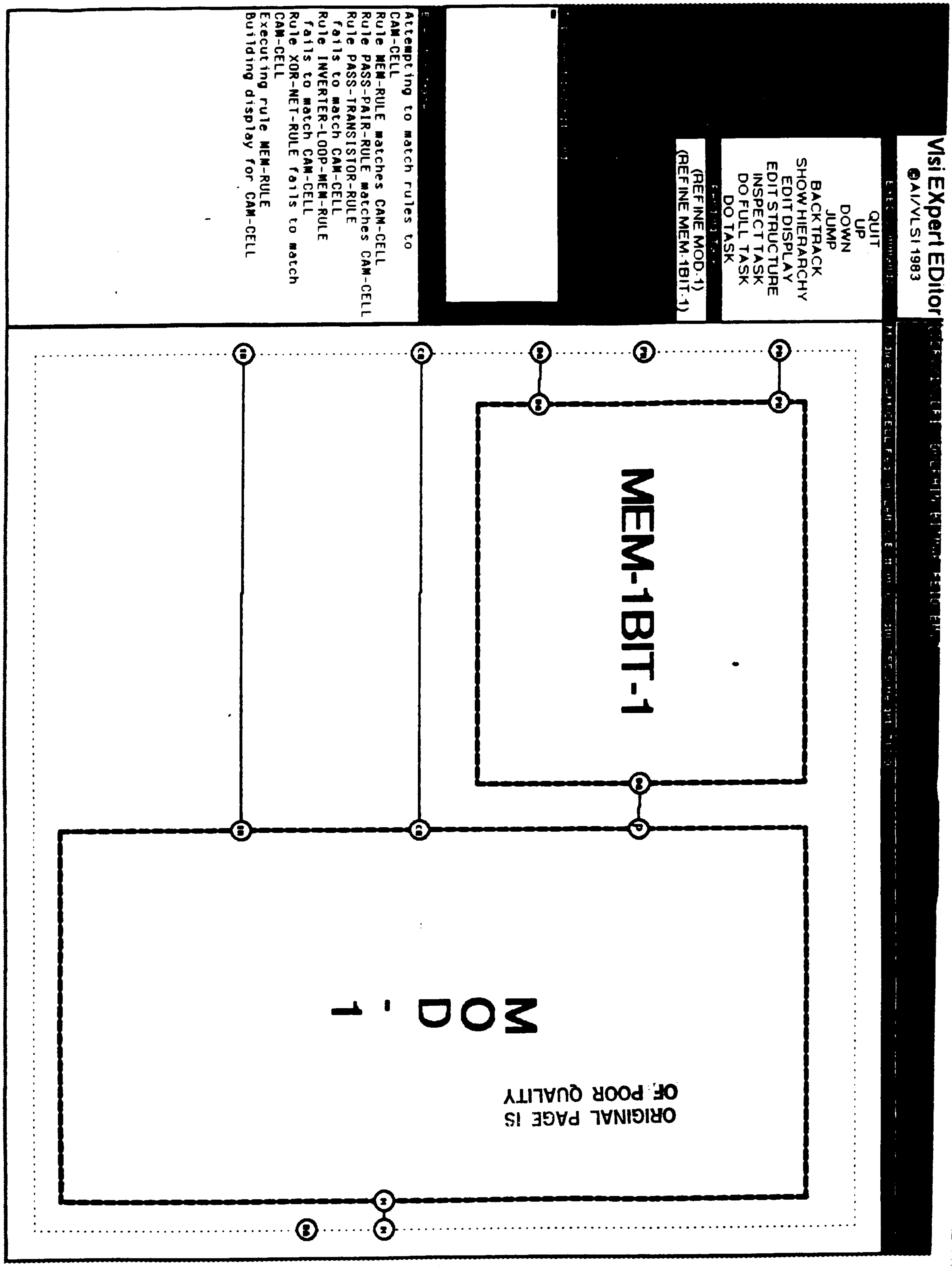

Figure 3 : Interface to the LEAP system

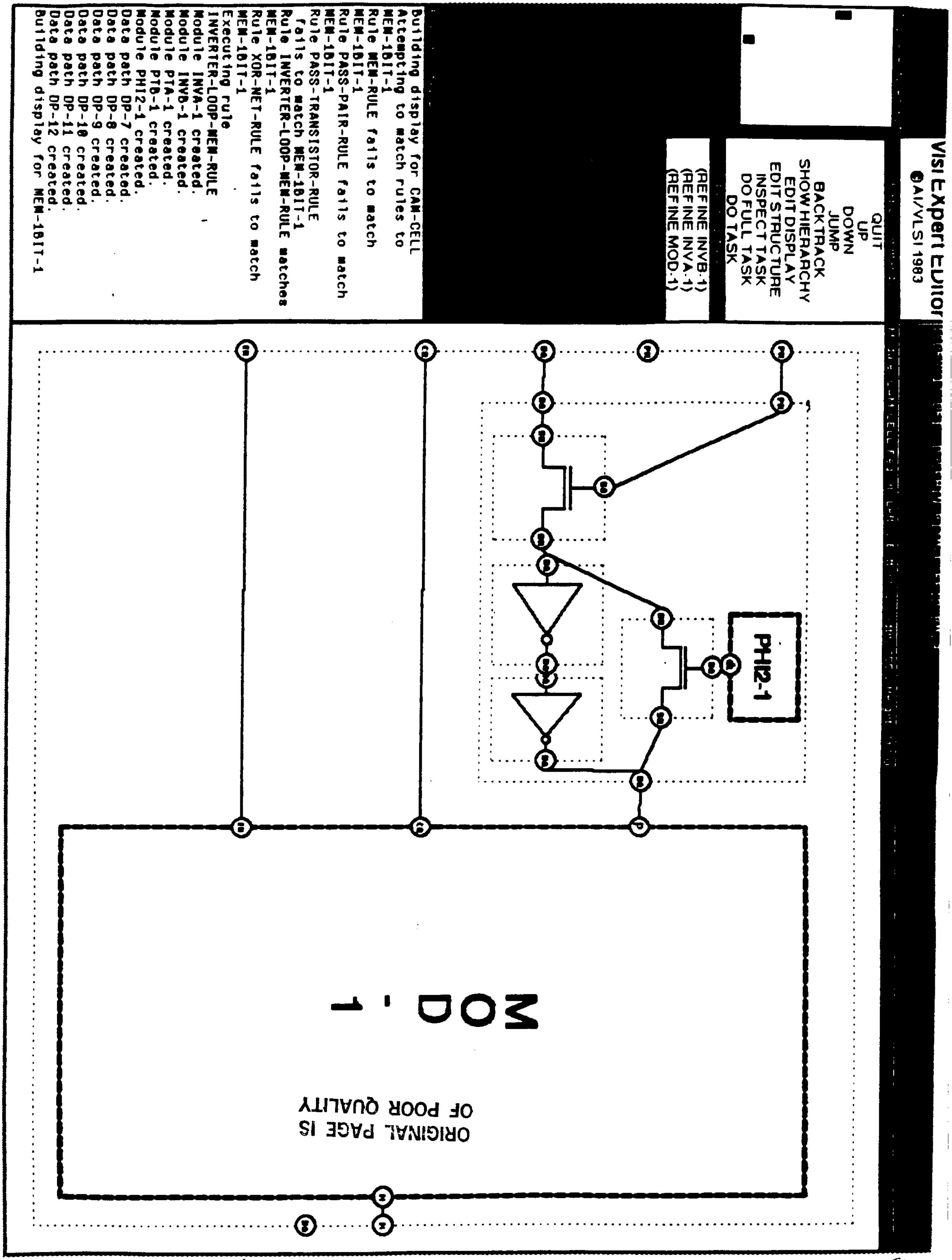

Figure 4 : Circuit refined by LEAP

EXPERT SYSTEMS:

APPLICATIONS IN SPACE

DRAFT
NOT FOR DISTRIBUTION,
ATTRIBUTION
OR QUOTATION

Bruce G. Buchanan

Knowledge Systems Laboratory
Stanford University
Stanford, California

198

CONTENTS

[p.199 DOES NOT EXIST]

INTRODUCTION

Artificial intelligence is one of the most important trends in computing because making computers behave intelligently is at least as important as manipulating data efficiently. Opportunities for using intelligent programs in NASA space station environments are numerous and obvious. But many of those opportunities require substantial research in artificial intelligence before they can be realized. This paper looks at the technology of artificial intelligence, especially expert systems, to define "from the inside out" what capabilities exist that are relevant for applications and environments in the space station, and what research needs to be promoted in order to achieve systems better able to interact symbiotically with a variety of persons for long times in space.

Anderson and Chambers (1985) mention a number of characteristics of systems in a human-centered space station. These include:

- symbiosis with humans: human and machine capabilities may complement one another
- autonomous,
- continuing operation for a period up to 20 years,
- operating in an information-rich environment,
- consequences of interactions with humans not entirely predictable,
- maturation of system implies flexibility to accommodate operational growth and minor upgrades,
- evolution of system implies flexibility to accommodate new and enhanced functionality,

- system may include electronic crew members (ECM's),
- humans may have to learn new skills to interact productively with computers,
- computers may learn from humans,
- autonomous agents may serve a variety of roles with varying degrees of decision making power and authority.

These are some of the relevant considerations in a top-down design of systems for the space station. Each of these points implies a research and development program of some intensity. This paper takes a bottom-up view of the same considerations -- i.e., starts with what exists today and asks how we can achieve these design goals. By doing so, I hope to introduce some relevant details into the design of systems and the planning of research.

Expert systems are now being used in many decision-making situations of direct relevance to NASA's mission, spanning manufacturing, engineering, medicine, and science. At present, they are used more as "intelligent assistants" than as replacements for technicians or experts. That is, they help people think through difficult problems and may provide suggestions about what to do, without taking over every aspect of the task.

Computer programs that reason autonomously are also of extreme importance in space, but they, too, must be integrated into an environment that is centered around people. They are extensions of present technology along several dimensions discussed here, that involve all of the same principles of design as the intelligent assistant programs.

One primary consideration is why intelligent systems are necessary in space. Although there are many reasons to build an expert system, they are all based on the premise: "Expertise is a scarce resource." The corollary (by Murphy's Law) is: "Even when there is enough expertise, it is never close enough to those who need it in a hurry." Because this is true -- almost by definition of the term 'expertise' -- constructing expert systems that reason at the level of NASA's, or their contractors', specialists may have several benefits. These are summarized in Table 1.

WHAT IS AN EXPERT SYSTEM?

The general nature of expert systems is familiar to everyone within NASA. A reiteration of the four major characteristics is provided below to help define the most important dimensions for research and development efforts.

An expert system is a computer program with expert-level problem solving abilities, which also fits some other criteria: it is a symbolic reasoning program that uses heuristics, its reasoning and knowledge base are understandable, and -- most importantly -- it is flexible. These characteristics are discussed below. All are important for applications in the space station, and all define research topics that will enhance current capabilities.

Example

One well-known expert system that has become a classic, although not actively used, is MYCIN. It was developed at Stanford by E. H. Shortliffe and others in the mid-1970's. Its task is two-fold: (a) diagnose the cause(s) of infection in a patient and (b) recommend appropriate drug therapy. From a medical perspective, MYCIN's knowledge base is now dated; from the perspective of expert systems it represents much of the kind of reasoning that is captured in today's systems. MYCIN's conclusions were demonstrated to be equal in quality to those of infectious disease specialists at Stanford Medical Center.

The sample typescript shown in Appendix A illustrates MYCIN's requesting information about a case and reasoning to conclusions about the best treatment.

Performance

Naturally we want computer programs to solve problems without error. But that is not always possible -- in fact, outside of mathematics and logic we don't have flawless methods we can put into programs. Specialists in engineering, science, education, the military -- and every area outside of pure logic -- must solve problems with less than perfect methods. How do they do it? Mostly by building up specialized knowledge through extra years of training and experience and by reasoning carefully with that knowledge in situations they have learned to recognize. They are not infallible, though. Specialists' decisions are challenged frequently -- most noticeably in the courts. So it is also unreasonable to expect computer programs to reason infallibly in all of these areas. Occasionally new methods are discovered that provide much better results than the established methods of the old practitioners. But these improvements can then be put into programs, thus raising the overall standard of performance while still keeping the same relative standard of comparison with the best specialists.

Reasoning

When we say that expert systems are reasoning -- and not just calculating with numbers -- we are saying that they belong to a class of programs using the methods of artificial intelligence (hereafter AI). In the 1940's, computers were used almost exclusively for large mathematical problems. At Los Alamos, for instance, scientists had to solve complex mathematical equations in order to calculate elements in the design of the atomic bomb. These applications are usually referred to as large-scale scientific computation, or "number crunching" for short. In the 1950's, IBM and other computer manufacturers, realized the enormous value in helping business solve problems of record keeping, payroll and the like. These applications extended the concept of computer-as-calculator to computer-as-data-manager.

In both of these classes of applications, the method of computation is error-free. There is no question that the result is correct, providing of course that the computer has been programmed correctly. A mathematical equation is solved correctly; an employee roster is sorted correctly -- if the methods are followed precisely. And computers are better able to follow complex instructions than people are. In computer science, logic and mathematics we call these procedures algorithms. They are procedures that can be guaranteed to provide a correct answer in a finite time, if there is one, and otherwise will provide a statement that the problem is not solvable.

Some algorithms are too expensive to use, however, even in computers. A classic example is finding the shortest route a travelling salesman can take to visit many cities once and end up at home. With more than a handful of cities, algorithmic methods will not finish in time to be useful. For this reason, alternative methods have been developed.

Around the mid 1950's and early 1960's an alternative style of computing came to be recognized as important. Instead of always using algorithms, a computer may use heuristics -- rules of thumb that aid in finding plausible answers quickly without guaranteeing the correctness of the results. Sometimes these rules of thumb are introduced into large numerical simulations in order to get the simulations to crank out answers more quickly. Or approximate methods may be substituted for more precise ones for the same reason. The assumptions may not all be correct; thus the results of the simulation may not be correct.

When heuristic (non-algroithmic) methods are combined with symbolic (non-numeric) data, we are dealing with that part of computer science known as artificial intelligence.

Understandability

When someone truly knows something, he or she can "give an account" of what he knows. In our terms, good performance is not enough to call a person (or program) an expert -- he/she (it) should also be able to explain why the solution is plausible, what features of the situation were noted to be important, what knowledge and problem solving methods were used. Otherwise we label a person as "consistently (but unaccountably) lucky", or maybe "psychic". Each field has its own standards of what a reasonable explanation is. A surgeon who recommends amputation of a leg generally talks about the process of disease or extent of injury and what will happen if it is not amputated. A broker who advised liquidation of one's stock portfolio may explain the advice with respect to technical charts, historical trends, or some economic principles that point to a stock market collapse. In their own communities, both the surgeon and the broker can usually justify -- in court if necessary -- the advice they give. And we regard them as experts partly because they have the knowledge that lets them do this.

Flexibility

We expect experts to be flexible in their thinking. And we regard persons as amateurs, not experts, when we encounter opinions that are rigid, locked-in ways of dealing with problems, or an inability to deal with new situations.

In particular, there are two situations in which we want expert systems to be flexible:

1. At advice-giving time we want the program (or a person) to provide good advice about situations that have never been encountered before. Novices with good memories may be able to provide the "textbook" answers for classic situations. Experts however, should, in addition, be able to reason about novel situations.

2. At the time a program is being constructed or modified (or a person is learning), we want it to be flexible enough to assimilate new bodies of information. There should be a capacity for growth of knowledge, not a rigidity that freezes either the depth or breadth of the program's knowledge.

SOME APPLICATIONS

Some of the types of problems for which expert systems have been constructed are shown in Table 2. Many of these, such as small troubleshooting assistance programs, are relatively straightforward. Although the state of the art is difficult to quantify, the programs in the table represent the kinds of commercially robust systems that can be built for NASA today, provided adequate resources and an appropriate problem. We don't have an adequate taxonomy of problem types. Many of these overlap, in being different forms of data interpretation, for example. Even this brief characterization, however, provides a reasonably good idea of what expert systems can do.

In general, expert systems can reduce costs or increase quality of goods and services -- in a single phrase, they can increase productivity in an organization. If you believe either that there is not enough expertise in the world, or that it is not well distributed, then you will be willing to entertain the idea that putting human expertise into an easily-replicated form may answer some productivity problems. Or, at least expert systems may provide a partial answer. Consider medical diagnosis. Specialists at university medical centers generally see more of the unusual disorders than a rural practitioner and thus stand a better chance of diagnosing them correctly. Putting some of that expertise more directly at the service of the rural practitioner could allow more effective treatment, and save patients the time and trouble of travel to the medical center.

Or consider troubleshooting a complex piece of equipment. Persons with the most field experience are often the ones promoted to desk jobs in the central office. When subtle combinations of causes keep a less experienced field service technician from fixing a mechanical failure, someone with more expertise is needed. On earth, depending on travel times and the criticality of the work flow in the central office, calling the experienced specialist out may be a very expensive repair procedure.

The following situations are all cases where it may make good sense to build an expert system:

- too few specialists for the number of problems;
- specialists not at the sites of problems when they occur;
- long training time for a specialist;
- high turnover among technicians;
- combination of complex equipment and poorly trained technicians;
- organization's best (or only) specialist in an area is nearing retirement;
- too many factors for a person to think through carefully in the time available.

KEY CONCEPTS

The four goals that characterize expert systems can be achieved with a few key methodological ideas. In this section, the key ideas will be introduced; in successive sections they will be elaborated on so as to explain a little how they work. The main organizational principle of expert systems is to keep specialized knowledge separate from the logical and heuristic inference methods that use it. This is easy to say but difficult to follow, for reasons that will be described later.

Keep Domain-Specific Knowledge Separate from General Reasoning Methods

-- KEY IDEA #1 --

Another key concept, which is imported from principled design of software generally, is modularity. (The first key idea is an instance of this, but that instance has taken on more importance than all the other instances of the general concept.) Modularity at the level of knowledge about the problem area implies conceptual separation of elements in the knowledge base. For example, medical knowledge about penicillin, although not totally independent, can often be separated from knowledge of other drugs. It can be modified in major ways, or deleted, without altering the program's knowledge of other drugs. So, this is to say that the concepts used to talk about objects in the domain should be chosen so as to allow talking separately about an individual object, a single property of an object, or a single relation of one type of object with another. Modularity at the level of programming constructs implies that the program's internal representation of knowledge elements (e.g., objects, properties, relations) is similarly "clean".

Keep independent pieces of knowledge independent.
Keep the rest as nearly-independent as possible.

--KEY IDEA #2 --

A third key concept is uniformity of conceptualization and representation of knowledge. The underlying intuition is that it is easier for a person or a program to build, understand, and modify a body of knowledge if it doesn't mix and merge a variety of different types of things. This is as true at the knowledge level as at the programming level. For instance, one of the most compelling aspects of Newton's Laws is that all physical bodies are treated as quantities with mass. He didn't need one set of laws for planets and another for apples. So it is desirable to build an expert system with a "conceptually clean", well-organized, simple collection of concepts. And it is important to use a simple, well-organized collection of programming constructs as well. Otherwise there are too many different kinds of things to keep track of and reason with.

There is more dispute among AI specialists about this principle. There are good reasons to violate it, as we shall see, in the interest of being able to say more about the objects and relations of interest than can conveniently be said in a single language. We are frequently told by bi-lingual friends, for instance, that there are some concepts that just can't be expressed fully in English. The same is true for programming constructs, but the basic principle for constructing expert systems is to try to maintain uniformity as much as possible.

Strive for uniformity of language
and programming constructs

-- KEY IDEA #3 --

A fourth principle is to design the expert system to mirror the ways experts think about problems in their domains. That means using the same terms and the same rules of reasoning as the experts use. One reason for this is that building and debugging a knowledge base depends necessarily on the expert, and using less familiar terminology or methods will introduce confusion and error before the knowledge base is completed. Also, after it is completed it needs to be comprehensible and unambiguous to the practitioners using the system or else confusion and error will result.

Note that we are assuming here that the expert designing the system knows how to make it understandable to users. Great care must be taken when building a system, however, to insure that this assumption is true.

There are times when this principle will be, or should be, violated. For example, when efficient computer algorithms can solve part of a problem, it doesn't often make good sense to use anything else for that part, even if the experts don't think about it in that way.

As much as possible, use the same vocabulary and methods in the program as the experts and practitioners use.

-- KEY IDEA #4 --

These key ideas help us achieve all of our four goals in the following ways.

- PERFORMANCE -- in problems whose solution methods are not already well formalized, which are considerable, much of the effort in building a knowledge base from an expert system lies in building the conceptual framework. Which properties and relations of objects to describe is often not well specified at the beginning. So the knowledge base is built incrementally, where experience with one knowledge base guides future modifications, extensions, or reformulations.

- o REASONING -- When the solution methods are not well characterized, it is important to encode heuristics that experts say they use. Storing these separately and in a simple form allows them to be changed easily. Since it is nearly impossible for an expert to articulate a complete and consistent set of heuristics at one sitting, it must be easy to add, remove, or modify the heuristics that determine the reasoning.

- o UNDERSTANDABILITY -- with modularity, individual elements of the knowledge base can be displayed meaningfully in isolation. Moreover, with the separation of knowledge base and inference procedures it is possible to peruse the knowledge base in order to find just those elements that were used to reason about a new case. And with uniformity of data structures, it is possible to build one set of procedures that produce explanations.

- o FLEXIBILITY -- when the elements of the knowledge base are in separate data structures, and not intertwined with code for inference procedures, we can add more knowledge with considerably more ease. When the individual items in the knowledge base are nearly separate, we have fewer interactions to worry about when we change an items. And when the representation is homogeneous, we can more easily write other programs that act as "editing assistants" or explainers that help us insure correctness of new items and help us understand what is in the knowledge base.

PERFORMANCE ISSUES

Expert systems constitute one class of computer programs. As such, they work the same way as every other program: they process input data to produce output data. But the nature of the processing is different from most conventional programs. The key ideas mentioned earlier are the key differences in the design and implementation of expert systems.

In order to design a reasoning program, we need to provide knowledge to reason with and reasoning methods to use. Both are needed. A powerful thinker needs something to think about, and a body of facts without methods for using them is sterile. Over the last few decades, research in AI has elucidated programming methods for making inferences and storing knowledge. We briefly characterize these topics below, although with some reservations about oversimplifying, in order to highlight research issues relevant to increasing the performance of expert systems. In addition to research on inference methods and representation of knowledge, several other issues are mentioned briefly as needing more research in order to improve the performance of expert systems.

Inference Methods

Aristotle's theory of the syllogism defined acceptable inference methods outside of mathematics for about 2000 years. His theory has been extended in this century by Russell & Whitehead, and others, in a formal theory that includes methods of reasoning with several statements and several variables in an argument.

Formal logic defines several inference rules which are guaranteed to create true conclusions if the premises of the argument are true. The chain rule (modus ponens) is the single most important inference rule in expert systems. It allows us to chain together a string of inferences:

If A then B

If B then C

If C then D

A

D

Many of the inferences we make in our lives are not guaranteed by the rules of logic, however, nor do we have certain knowledge about the truth of our premises. Whenever we argue that the future will be like the past, as in stock market predictions, we have to be prepared for exceptions. These inferences, labeled "plausible inferences" by George Polya, are the ones of most interest in AI.

One set of programming methods were in AI for making plausible inferences is to assert the facts categorically -- as if they were known to be true with certainty -- and then reason about exceptions that might force revisions to the conclusion.

Another set of methods deals explicitly with the degrees of uncertainty in the facts and in the associations. MYCIN (see Appendix A) uses this style of reasoning. Usually the degrees of uncertainty implied by words like "often" and "may" are expressed as numbers. And often these numbers are interpreted as probabilities.

A third, and most powerful, set of methods is to introduce heuristic rules, or rules of plausible inference, into the reasoning. These are facts or relationships that are not guaranteed to produce correct conclusions, but will often do so. Moreover, they often produce answers more quickly than their algorithmic counterparts. In the traveling salesman problem, for example, the problem is to plan a route for visiting each city in a set exactly once and end at the home city. This is an NP-complete problem, that is, the algorithm for solving it takes times that is exponential with the number of cities. One heuristic we may introduce is to go to the nearest city that has not yet been visited. This certainly speeds up the computation of the route, but may (and probably will) miss the route that is shortest overall. Some rules of plausible inference used, with caution, in some expert systems are shown below:

- Satisficing: If it will be expensive to find the very best solution to a problem, then stop with the first solution that satisfies easier criteria of being good enough.

- Inheritance: (Some specified) properties of a whole are shared by all its parts. E.g., An ice cube is cold and hard. Pieces of an ice cube are cold and hard. [But other properties, like "weight", do no behave the same.]

- Single Fault: If a piece of equipment (or any organized system) is malfunctioning, and one hypothesis explains the problem, then there probably is only a single cause of the problem.

- Compelling Evidence: If you have gathered a lot of evidence in favor of hypothesis H1, and very little evidence against it, and you have gathered little positive evidence for alternative hypotheses, then H1 is a plausible hypothesis.

- Decomposability: If there are many parts to a problem that are nearly independent, assume they can be solved independently. Then adjust the composed solution to take account of known interactions.

- Parsimony of Design: Designs or plans with fewer elements are preferred to those with more.

In principle, the rules of inference (both logical and plausible) may be applied again and again to a situation description, in any order, and the resulting conclusions will be the same. This is not always possible in practice, however. There may not be enough time to reason exhaustively about all possibilities and contingencies. For that reason AI researchers talk about controlling the inferences as being a more important, and more difficult, problem than making the inferences.

Controlling inferences breaks down into two subtasks: (a) deciding which rules to apply now, at this stage of the problem-solving process, and (b) deciding which part of the problem to work on now. Since we believe these subtasks require some intelligence, all of the principles for building knowledge-based systems also apply at this level of reasoning In particular, it is desirable to make this control knowledge explicit and separate from the inference methods

Representation of Knowledge

We have said that a key idea in building expert systems is storing knowledge separately from the inference methods. Another key idea was to avoid, as much as possible, representing it in a low-level computer language. But we have not said how to represent for the computer what an expert wants to tell it. English is too difficult for a computer to interpret unambiguously; FORTRAN and BASIC are too low-level for an expert to deal with efficiently. Clearly we need some stylized representations that are somewhere in between.

AI researchers have developed several different representation methods. There is no single one that is best in every case -- they each have strengths and weaknesses One of the fundamental trade-offs in thinking about the representation of knowledge is between simplicity and expressive power. We want a simple set of conventions for storing knowledge because that makes it easier -- for a person or a program -- to understand what is in the knowledge base at any moment. It is also easier to write simple statements without error. Aristotelian logic ("All A's are B's", etc.) and arithmetic are simple representations. The difficulty is they lack the expressive power to let us say everything we think is important about a problem. A hundred years ago DeMorgan noted the lack of expressive power is Aristotelian logic (and a weakness in its inference methods): if you know that all horses are animals, he said, you cannot prove that the head of a horse is the head of an animal. This sort of problem led Russell & Whitehead to develop a formalism with more expressive power.

There are two major classes of representation methods, reflecting two different ways of viewing the world: action-centered or object-centered. Different problem areas may focus on one or the other, or different experts in the same problem area may. For example, physicians talk about disease and classes of diseases as entities with expected properties and also talk about clinically relevant actions that determine what to do -- e.g., asking questions, measuring things, relating signs and symptoms to possible causes, matching likely causes to acceptable therapies. Neither point of view is wrong, but they focus on medical phenomena quite differently. And an expert system would similarly have one focus or the other.

Action-centered representations focus on conclusions that can be drawn from facts or, more generally, on relations between situations and actions. The formalism of mathematical logic is one popular choice. Another popular formalism is rules.

Object-centered representations focus on the organization of objects in the world, for instance into hierarchies. They still allow conclusions to be drawn when an object is found to have some properties, but those inferences are triggered from "within" an object rather than from outside. That means that objects and their properties -- and changes to any of them -- drive the inferences. But in an action-centered model, the inference rules drive the creation of new objects and properties. The net effect may be identical, as we said, but the way one thinks about the domain of discourse is distinctly different.

Also, in object-centered representations there is more machinery for saving storage space by using hierarchies. Properties of classes of object, for example, may be implicitly inherited by all of the instances without having to store it with each instance. The manager of a group is the manager of each person in the group, so the program only needs to store (once for each group) the name of the group manager and can use that, plus the class-instance hierarchy, to find the name of any individual's manager.

There are as many different conventions for representing knowledge as there are AI researchers working on this topic. This can be confusing when reading the literature. But they are basically all variations -- usually mixtures -- of the two different styles just discussed. There are many expert systems built out of these two sets of ideas, but considerably more experience -- and analysis -- is necessary to understand their strengths and limitations.

Validation and Robustness

It is impossible to prove logically that the contents of an expert system's knowledge base are correct or complete or that the inference procedures will always provide the best answers. Yet persons in a space station whose equipment and lives depend on the expertise of many systems need to know the scope and limits of each system. Or, alternatively, they need tools for determining the scope and limits of the programs they use. These range from better explanation systems to tools for checking knowledge bases.

Spatial and Temporal Reasoning

Many complex problems in a space station require autonomous computer programs that represent and reason about three-dimensional objects. Simpler representations do not allow programs to solve problems involving 3-d shapes and positions, such as problems of fitting parts or of maintaining some equipment. Building expert systems requires attention to making the systems' reasoning understandable to persons onboard the space station and changeable by them. That, in turn, requires a flexible, high-level description language as well as computationally efficient operations that implement the language.

Similarly, reasoning about sequences of inter-dependent actions and about situations that may change at arbitrary times are important aspects of problem solving in space.

Very Large Knowledge Bases

To date expert systems have used knowledge bases of modest size. With the complexity of operations in space, we need to design and maintain expert systems with very large knowledge bases. Although size is difficult to define, most knowledge based mention only a few thousand different facts and relations. Probably the largest today is the INTERNIST system in which about 250,000 facts are encoded (Miller et al., 1982). Some of this limit results from our own inability to keep in mind the interrelationships among more facts as much from the technology of storing and retrieving them. We must improve the technology to make it easier to build and maintain knowledge bases of much larger scale, which will be necessary in a system as large and complex as the space station.

Shared Knowledge Bases

Today's systems use single knowledge bases that have been built specially for them. As more and more systems are constructed, however, it will be important to use knowledge bases in different contexts and then reuse one system's knowledge base in another system. It is wasteful -- and should not be necessary -- to duplicate the contents of an old knowledge base in a new application. One should expect, for example, programs in the space station that reason about the function of life support equipment and others that reason about the mechanical structure of the same equipment, both of which must share considerable detail about the equipment itself.

Distributed Databases

Databases exist now on many machines. Yet it is nearly impossible to treat several of them as if they were one logical unit -- from any program. Expert systems also need this capability. Current research will allow much broader sharing of data among different databases than is currently available in commercial systems. There will be many computers in the space station. It is much sounder to think of separate specialized databases (with appropriate backup) that can accessed from various programs than to consider separate copies of every data base on every machine.

Parallel Computation

Computers are fast, but never fast enough. In addition to the immense speed-ups from improvements in the hardware, there are potential speed-ups from software. When a problem can be divided into nearly independent subproblems, it is conceptually easy to see that multiple computers could be used to solve the subproblems in parallel, thus saving considerable time. Work in the research laboratories indicates that this is feasible. Thus it will almost certainly become a commercial reality in the near future if it is cost-effective.

DEVELOPMENT AND MAINTENANCE ISSUES

Building an expert system requires finding out how an expert solves a problem and translating that expertise into a stylized form that can be read by computer. This is no different in principle from building a conventional program in which programmers find out what equations or algorithms experts use and then write FORTRAN or COBOL programs that embody those procedures. The main difference in practice is that expert systems must incorporate knowledge that is much more qualitative and judgmental. In fact, much of the time the expert's "know-how" is not yet written down and what he/she does is regarded as an art.

Because the expert's knowledge is often not already codified and because writing symbolic reasoning programs is itself often regarded as an art, building an expert system requires patience. It generally works best as a team effort involving one or more experts and one or more so-called knowledge engineers. A knowledge engineer is a programmer of knowledge-based systems who understands the conventions of the computing framework and who assists the expert in mapping judgmental knowledge into that framework. The dialogue between expert and knowledge engineer is often called "knowledge engineering".

One of the key ideas in knowledge engineering is to focus on case studies. It is much easier for any of us to tell someone how we would approach a specific situation than to say in general terms how we solve problems of a type. Of course, if we have a set method (sometimes called a "canned procedure") that we always use, we can say that. "Oh yes, I always use the French variation of the Alekhine-Gorbachev wave theory in situations like that", you might say. But then the knowledge engineer wants to know what do you do <u>next</u> and -- more interestingly -- when would you make exceptions to your set policy. And the best way for you to think about those things is to focus on cases. As long problem solving requires more than the application of set procedures, knowledge engineers will need to go through many cases, and variations on them, to help codify the expert's judgemental expertise.

Steps Involved in Knowledge Engineering

It may take months or years to build an expert systems, with the time depending largely on the complexity of the problem and the extent to which expertise is already codified. One reason it takes so long is that there are many steps involved. And at each step, the knowledge engineer or the expert may decide it is necessary to undo some results of previous steps. Very roughly, the steps are thought of as beginning, middle and end phases in which attention is focused on different aspects of the system, as shown below:

- o Beginning -- define the problem precisely; understand which concepts are used, what their definitions and inter-relationships are

- o Middle -- implement a substantial prototype after choosing a set of representation conventions and writing a small but substantive knowledge base.

- o End -- fill out the knowledge base to fix errors and extend the scope of the system's problem solving abilities, both of which are generally discovered by testing the systems on many test cases.

Tools to Aid in the Construction of Expert Systems

Just as carpenters can construct houses faster with the right tools, knowledge engineers can build expert systems faster with software tools that boost their productivity. These come in several forms. The main idea, however, is to provide programmers with mechanized intelligent assistants that know about programming conventions (including abbreviations and shortcuts), that can help locate and fix errors, that can display the contents and interrelationships in a program or knowledge base, and so forth. These are the kinds of extra capabilities that distinguish system-building environments from programming languages.

Some of the more powerful environments -- sometimes called shells -- are shown below. One characteristic of a shell is its commitment to a set of representation conventions of the sort outlined previously. See Table 3.

Learning

At present, expert systems do not learn from experience. This is a defect that many research groups are working to remedy. Early prototypes of learning systems promise some automated assistance in maintaining and extending a knowledge base through the experience of routine use, but these are not yet available.

It is possible, however, to learn an initial set of rules from a case library (collected past experience) and use it for classification problems. Induction programs are being used to build simple rule sets for expert systems in which there is little chaining of the rules and little use of uncertain inferences. These are largely marketed in Great Britain where it is better understood that even simple problems may carry great economic leverage. Current research is extending the scope of induction programs to more complex rule sets.

Resources Required

The major cost involved in building an expert system is in personnel time. Shell systems now run on most common computers, so it is not necessary to buy new equipment and, most importantly, it is not necessary to build the complete set of programming tools found in a shell.

Purchasing the shell and some training in how to use it are recommended. The amount of time needed from a team of experts and knowledge engineers is variable -- as are their salaries. Table 4 gives some estimates for a hypothetical small system constructed within an existing shell.

It is assumed here that a problem has been precisely defined before beginning, that a case library of at least a half dozen typical and hard cases has been assembled, that a commercial shell has been purchased and runs on an available computer, and that the senior knowledge engineer is very familiar with both the shell and the computer. It is also assumed that the team's primary responsibility is this activity, and that they have the blessing of their management.

In this simple model, the senior knowledge engineer also fills the role of project leader, with as much as half his/her time filled with reports, briefings, budgeting, and other managerial responsibilities. The junior knowledge engineer in this model is responsible for software engineering -- that is, integration of the expert system into the run-time environment -- as well as for help in building the knowledge base. And the expert, here, is (atypically)) also filling the role of "management champion" with some time devoted to securing resources to make the project happen.

One of the main factors that determines the length of time a project will take is, not surprisingly, the nature of the problem. This includes both the scope of the problem and the extent to which a commercially available shell is appropriate for the problem. Another main factor is the definition of the "deliverable", that is the terms of the contractual agreement specifying whether the product delivered is a prototype or is a smoothly polished software package.

There are added gains in building an expert system that offset some of the costs just mentioned. Besides the obvious gains showing up in work performed, there are very noticeable gains in the quality of information available.

Shortening the time required to build systems and increasing our ability to maintain them are thus two of the central issues for putting expert systems in the space station.

DRAFT

ENVIRONMENTAL ISSUES

The environments in which expert systems currently operate are closely constrained. While there is wide variation in the degree of autonomy exhibited across all working systems, most systems in place are interactive, requiring intelligent input from humans. The predominant model of interaction is a consultation model in which an expert systems asks a person for the facts (and interpretations of them) and then provides some advice. A consultation with MYCIN about a medical case is shown in the Appendix.

There are several reasons why the consultation model is appealing, each of which constitutes an opportunity for research. In the first place, a program that asks short-answer questions of a person can finesse the very large problem of understanding free-form English sentences and phrases. The program knows what answers are reasonable in the current context and can have advance expectations about the ways these answers may be framed.

Second, the consultation model provides a strong sense of context which not only helps the program understand a person's answers, but helps the person understand the sense of the questions. This is important because misinterpretation of the program's questions can have serious consequence.

Third, in a consultation it is reasonable to make strong assumptions about the users of an expert system -- what they know, what they don't know, what vocabulary they use, what environment they are working in, and so forth. This helps minimize problems in communication. This means also that so-called "common sense" knowledge may be supplied by users and need not all be supplied by the program.

Real Time Monitoring

As expert systems become faster, it will be easier to build systems that monitor other devices or processes with rapid changes. Conceptually a difficult problem is managing time-dependent relations efficiently, which is one of the necessary components of a monitoring system. The large amounts of data received and the speed with which they are received are also critical issues. Integrating AI methods of reasoning about the data with numerical methods for digitizing and filtering is essential.

Richer Input/Output

No one likes to interact with computers by typing. Considerable work on interactive graphics has reduced the need for typing. But it will be even easier when we can communicate with programs by giving voice commands and receiving spoken English output in return.

Models of Users and Situations

No single style of interaction is best for all users at all times. Specialists do not need explanations of the meanings of terms, for example, while less experienced users used considerable help understanding the context of the problem. Also, the criticality of the situation may demand taking shortcuts in data acquisition or reasoning to reduce the risk immediately before taking a more systematic, detailed look at the problem. Expert systems must be sensitive to models of both the user and the situation in order to request appropriate input, reason at an appropriate level of detail, and present conclusions and suggestions in an appropriate way.

CONCLUDING OBSERVATIONS

Expert systems already are saving organizations millions of dollars and performing tasks routinely that ordinarily require human expertise. The number of applications of today's technology is nearly boundless -- consider, for example, the number of pieces of equipment in a space station that we don't readily know how to fix. The first commercial shells on the market are robust enough to be used effectively. Integrating intelligent systems with conventional computer programs and with persons in the space station involves new research in many dimensions. The single biggest advantage of AI programs, amply demonstrated in expert systems, is their flexibility. This matches precisely the single biggest design requirement on software in the space station.

What we see now is just the beginning of a wave of intelligent software that can have as great an effect as business data processing software. It is impossible in any area of technology to make accurate predictions. However, there are many parallels between the growth of expert systems and of computing hardware, with about a 25-30 year lag. When electronic computers became available commercially, businessmen began to ask about applications that would make a difference to them. In 1955, several of these innovators assembled at Harvard to discuss their experiences. Some of the conclusions they drew from their early experience are summarized below (Sheehan, 1955):

1. "The initial overenthusiasm, which inevitably accompanies a project of this scope, can and does make the job harder. Too many people had the impression that this was the answer to all problems. Perhaps it is, but we haven't been smart enough to develop all of them...

2. "Some of our original thinking has been partly confirmed in that the greatest benefits to be derived from a computer will probably consist of information impossible to obtain previously...

3. "Our experience has shown that the computer is more adaptable to some projects than others...

4. "Programmers should be recruited within your own company...It is easier to teach men the required computer and program techniques

than to acquaint them properly with the complex procedures and routines of modern-day industry...

5. "I doubt if it is possible to overemphasize the desirability of providing for convenient corrections or deletion of errors in data...

6. "The maximum justifiable amount of flexibility for extending or integrating applications must be included in the initial programming..."

The parallels with expert systems are clear: we have just begun to explore the first set of applications and have only a limited view of possibilities in the future. Over the next 5-10 years, however, we do know what extensions in capabilities will be possible because they are the ones that are active research areas in the present.

Future developments will make the technology more broadly applicable and easier to use. Every professional activity -- from medicine to manufacturing -- contains many problem solving tasks that are not mathematical. The development of artificial intelligence methods that can solve many of these kinds of problems will change the nature of applications of computers in our society as much as the development of transistors changed the architecture and applicability of computers themselves.

NOTES

1. From B. G. Buchanan and E. H. Shortliffe, eds. Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project. Addison-Wesley, 1984, pp. 706-712.

INFECTIOUS DISEASE CONSULTATION SESSION in MYCIN[1]

(This is a typescript of a MYCIN consultation on an actual patient treated at Stanford, MYCIN was not used to suggest treatment. Spaces have been added for readability, and comments (in italics) have been inserted. Otherwise it is a verbatim transcript. User's responses follow double asterisks.)

Patient--538--

1) Patient's name:

**PT538 (Name, age, sex, and race are always asked because they are INITIALDATA.)

2) Age:

**34 YEARS

3) Sex:

**MAEL

=MALE [MYCIN corrects misspellings.]

4) Race:

**WHY [The user asks for an explanation. In this instance it is trivial.]

[i.e. WHY is it important to determine the race of PATIENT-538?]

[1.0] The race of a patient is one of the 5 standard parameters that identify a patient.

This may be useful later in the consultation. For example, it is relevant in determining the organisms (other than those seen on cultures or smears) which might be causing the infection.

[back to question 4...]

**CAUCASIAN   [This is the answer to the question about race.]

Please supply one answer under each heading. If your response is not understood, you will be prompted by the standard questioning method. (enter ? for normal prompting, * for questions that are not applicable)

Please enter the results of positive cultures using the following table.

site	culture number	date collected	date/ positive	method/ of collection [urine, sputrum only]

5) **DONE

(The next question asks for several items in tabular form since they are found together in the hospital record.)

Please enter the results of pending cultures in the following table.

site	culture number	date collected	stain/ examined	method of/ collection [urine,sputum only]

6)**CSF1234 30-Jan-77 YES*

stain of smear	morph/	[MYCIN follows up on the stain of the smear and the morphology.]

[type DONE if no organism was seen on the smear]

7)**DONE

[No information about stain or morphology is available. MYCIN now follows up question 6, asking about other positive cultures.]

site	cul #	date coll.	smear examined?/	coll. method/

8)**DONE

[There are no more positive cultures.]

Please enter negative cultures in the following table.

site	culture number	date/ collected	method of/ collection [urine,sputum only]

9) **DONE [There are no negative cultures.]

10) Do you suspect Pt538 may have an infection at a site from which you have not obtained culture specimens?

**NO

Please enter information on current antimicrobials.

drug name	route of administration	date/ started
11) **CEPHALOTHIN	IV	23-Jan-77

12) **DONE

Please enter information on prior antimicrobials.

drug name	route of administration	date/ started	date/ stopped

13) **DONE

[Considering organisms (other than those isolated from cultures) which might be present...]

14) HasPt538 recently had symptoms of persistent headache or other abnormal neurologic symptoms (dizziness, lethargy, etc.)?

**YES

15)Has Pt538 recently had objective evidence of abnormal neurologic signs (nuchal rigidity, coma, seizures, etc.) documented by physician observation or examiniation?

**YES

The CSF cultures will be considered to be associated with MENINGITIS.

16) Please give the date on which clinical evidence (symptoms, signs, or laboratory tests) of the meningitis first appeared.

**29-Jan-77 22:15

17) Does Pt538 have an abnormal chest X-ray?

**NO

18) Does Pt538 have one or more of the following risk factors or TB:a) positive intermediate strength PPD,b) history of close contact with a

person with active TB,c) a household member with a past history of active TB,d) apical scarring on chest X-ray, e) history of granulomas on biopsy of liver, lymph node, or other organs?

**NO

19) Does Pt538 have any cutaneous lesions or rash on physical examination?

**NO

20) Does Pt538 have evidence of ocular nerve dysfunction?

**NO

21) Is Pt538 a compromised host (e.g. alcoholic, sickle-cell-disease, immunosuppressed)?

**YES

22) Is Pt538 immunosuppressed (e.g. corticosteroid therapy, cytotoxic drug therapy, radiation therapy, leukemia, lymphoma)?

**NO

23) For how many days has Pt538 had abnormal neurologic signs?

**7.2 HOURS [Note that the answer was given in hours. MYCIN has to be flexible enough to convert units.]

24) For how many days has Pt538 had neurologic symptoms?

**1 DAYS

25) Has Pt538 had recent evidence of tender or enlarged salivary glands?

**NO

26) Has Pt538 been exposed to any contagious disease recently (e.g. meningococcal disease, mumps)?

**NO

Please enter CSF findings in the following table

CSF	Percent	Prote

**NO

(31) Has Pt538 ever undergone any type of surgery?

**YES

32) Does Pt538 have a history of NEUROSURGERY?

**NO

33) Does Pt538 live in a crowded environment?

**NO

34) Do you suspect recent alcoholic history in Pt538?

**NO

35) Is meningitis a hospital-acquired infection?

**WHY

This is a more interesting question, requiring presentation of the rule that caused question 35 to be asked. Note that the first five clauses of the rule have been investigated and question 35 is motivated by the last clause. MYCIN first rephrases the question to give the user its interpretation of the WHY.)

[i.e. WHY is it important to determine whether the infection was acquired while the patient was hospitalized?]

[2.0] This will aid in determining the organisms (other than those seen on cultures or smears) which might be causing the infection. It has already been established that

[2.1] the infection which requires therapy is meningitis, and

[2.2] organisms were not seen on the stain of this pending CSF culture, and

[2.3] the type of the infection is bacterial, and

[2.4] neurosurgery is not one of the relevant items from the history of Pt538, and

[2.5] the age of Pt538 is greater than 10 days

Therefore, if

[2.6] the infection was acquired while the patient was hospitalized

then

there is evidence that the organisms (other than those seen on cultures or smears) which might be causing the infection is e.coli (.75) staphylococcus-coag-pos(.3) pseudomonas-aeruginosa (3)klebsiella-pneumoniae (.5)

Author: Yu

[RULE545]

[back to question 35...]

[MYCIN now requests the answer to question 35 again. Instead of answering, the user asks how clause 2.3 was confirmed.]

**HOW 2.3

[i.e. HOW was it established that the type of the infection is bacterial?]

[Rather than printing the contents of the rules, MYCIN shows the rule numbers and the strength of positive and negative evidence from each rule.]

The following rules concluded about:
    the likelihood that the type of the infection is bacterial

	Rule	Yes	No
[2.7]	RULE148	(.19)	
[2.8]	RULE500	(.51)	
[2.9]	RULE501	(.95)	
[3.0]	RULE502	(.97)	
[3.1]	RULE526	(.98)	
[3.2]	RULE504	(.97)	
[3.3]	RULE524	(.98)	

Thus, it has been established that the type of the infection is bacterial.

[back to question 35...]

**HOW 2.5

[i.e.HOW was it established that the age of pt 538 is greater than 10 days?]

In answer to question 2 you said that the age of pt538 is 34 years

Thus, it has been established that the age of pt538 is greater than 10 days.

[back to question 35...]

**Y

36) Is Pt538's clinical history consistent with EPIGLOTTITIS?

**NO

37) Is Pt538's clinical history consistent with OTITIS-MEDIA?

**NO

38) Has Pt538 ever undergone splenectomy?

**NO

39) Is Pt538 a burn patient?

**WHY [Another explanation is requested.]

[i.e. WHY is it important to determine whether PATIENT-538 has been seriously burned?]

[4.0] This will aid in determining the organisms (other than those seen on cultures or smears) which might be causing the infection.

It has already been established that

[4.1] the infection which requires therapy is meningitis, and

[4.2] organisms were not seen on the stain of this pending CSF culture, and

[4.3] the type of the infection is bacterial

therefore, if

[4.4] pt538 has been seriously burned

then

there is suggestive evidence (.5) that pseudomonas-aeruginosa is one of the organisms (other than those seen on cultures or smears) which mightbe causing the infection

Author: Yu

[RULE578]

[back to question 39...]

**WHY [In this case, the user asks another WHY question, backing up the subgoal tree.]

[i.e. WHY is it important to determine the organisms (other than those seen on cultures or smears) which might be causing the infection?]

[5.0] in order to determine the therapeutic regimen of patient-538.

If

[5.1] information has been gathered about organisms isolated from the patient, organisms noted on smears of the patient, negative cultures of patient, suspected infections without microbiological evidence, current drugs of the patient and prior drugs of the patient, and

[5.2] an attempt has been made to deduce the organisms which require therapy, and

[5.3] You have given consideration to organisms (other than those seen in cultures or smears) which might be present, and found at least one organism which requires therapy.

then

determine the best therapy recommendation from among the drugs likely tobe effective against the significant organisms

[RULE092]

[back to question 39...]

**Y

[MYCIN now prints its intermediate conclusions about the likely causes of the infection.]

I have found evidence (RULE545 RULE557) that treatment should cover for E.coli causing meningitis.

I have found evidence (RULE545 RULE557) that treatment should cover for Pseudomonas-aeruginosa causing meningitis.

I have found evidence(RULE545 RULE557) that treatment should cover for Klebsiella-pneumoniae causing meningitis.

I have foundevidence (RULE507) that treatment souldcover for Diplococcus-pneumoniae causing meningitis.

[Now MYCIN begins to ask questions related to appropriate therapy.]

40) Does Pt538 have focal neurological signs?

**UNKNOWN

[Some questions are motivated by the organism under consideration. Thus the following lines sometimes provide useful contextual information to a user.]

[Determining which drugs are desirable for use against the E.colli...]

[Determining shich drugs are desirable for use against the Pseudomonas-aeruginsa...]

[Determining which drugs are desirable for use against the Klebsiella-pneumoniae...]

[Determining which drugs are desirable for use against the Diplococcus-pneumoniae...]

My therapy recommendations will be designed to treat for organisms that either are very likely or, although less likely, would have a significant effect on therapy selection if they were present. It is important to cover for the following probable infection(s) and associated organism(s):

INFECTION-1 is MENINGITIS

+ <ITEM-1> E.COLI [from clinical evidence only]
+ <ITEM-2> PSEUDOMONAS-AERUGINOSA [from clinical evidence only]
+ <ITEM-3> KLEBSIELLA-PNEUMONIAE {from clinical evidence only]
+ <ITEM-4> DIPLOCOCCUS-PNEUMONIAE [from clinical evidence only]

41) Does Pt538 have a clinically significant allergic reaction to any antimicrobial agent?

*NO

[Considering AMPICILLIN-AND-GENTAMICIN for use against INFECTION-1...]

42) Do you have reason to suspect that Pt538 may have impaired renal function?

** NO

43) Pt538's weight in kilograms (or <number>POUNDS):

**70

[Finally, MYCIN prints the best therapy it has found to maximize the chances of effective coverage of the causes(s) of the infection while minimizing the number of drugs, their toxicity, and side effects.]

[REC-1] My preferred therapy recommendation is as follows:

In order to cover for items <1 2 3 4>:

Give the following in combination:

1) AMPICILLIN

Dose: 3.5g (28.0 ml)q4h IV [calculatedon basis of 50 mg/kg]

2) GENTAMICIN

Dose: 119 mg (3.0 ml, 80mg/2ml ampule) q8h IV

[calculatedon basis of 1.7 mg/kg plus consider giving

5 mg q24h Intrathecal

Comments: Monitor serum concentrations

Since high concentrations of penicillins can inactivate aminoglycosides, do not mix these two antibiotics in the same IV bottle.

Do you wish to see the next choice therapy?

**YES [Other therapies can be examined.]

Which, if any, of the above drugs do you wish eliminated from consideration?

**AMPICILLIN

[Considering GENTAMICIN for use against INFECTION-1...]

[Considering PENICILLIN for use against INFECTION-1...]

[REC-2] Next best therapy recommendation:

In order to cover for items <1 2 3>:

Give: GENTAMICIN

Doses: 119 mg (3.0 ml, 80mg/2ml ampule) q8h IV [calculated on basis of 1.7mg/kg] plus consider giving 5 mg q24h Intrathecal

Comments: Monitor serum concentrations

In order to cover for item <4>:

Give: PENICILLIN

Dose: 3.5 million units (7.0 ml) q4h IV [calculated on basis of 50000 units/kg]

RECOMMENDATION-2 is equivalent to RECOMMENDATION-1.

Would you like to see details of the comparison?

**YES

Looking at the coverage of the most likely items, we see that:

RECOMMENDATION-1 consists of 1 first choice drug (AMPICILLIN for ITEM-land ITEM-4) and 1 third choice drug (GENTAMICIN for ITEM-1, ITEM-2 and ITEM-3)

RECOMMENDATION-2 consists of 1 first choice drug (PENICILLIN for ITEM-4) and 1 third choice drug (GENTAMICIN for ITEM-2, ITEM-1 and ITEM-3)

ORGANISIMS	RECOMMENDATION-1	RECOMMENDATION-2
'most likely'	Drug-Choice Ranking	Drug-Choice Ranking
ITEM-1	AMPICILLIN-AND-GENTAMICIN-1st	GENTAMICIN-3rd
ITEM-2	GENTAMICIN-2nd	GENTAMICIN-2nd
ITEM-3	GENTAMICIN-3rd	GENTAMICIN-3rd
ITEM-4	AMPICILLIN-1st	PENICILLIN-1st

(The desirability of a drug is defined to be its lowest ranking for the items it covers.)

Do you wish to see the next choice therapy?

**NO

REFERENCES

Anderson, J. L. and A. Chambers.

1985 Human-Centered Space Station Design. Paper presented at Conference on Computers in Aerospace, Long Beach, CA, October 21-23.

Bauman, B.

1984 MBA Thesis, University of Texas, December.

Buchanan, B. G.

1986 Expert systems: working systems and the research literature. Expert Systems 3(1):32-51. Also appeared as Stanford KSL Report 85-37 and STAN-CS-85-1075.

Harmon, Paul.

1986 Inventory and analysis of existing expert systems. Expert Systems Strategies 2(8):1-16.

Miller, R. A., Pople, H. E. and Myers, J. D.

1982 INTERNIST-1, An Experimental Computer-Based Diagnostic Consultant for General Internal Medicine. NEJM 307(8):468-476. Also appears in 'Readings in Medical Artificial Intelligence: The First Decade', Clancey. W. J. and Shortliffe, E. H., eds., Reading, MA, Addison-Wesley, Pp.190-209.

Sheehan, G. M

1955 An application to payroll. _Proceedings of the Automatic Data Processing Conference_. Harvard University Press, Cambridge, MA, September.

TABLE 1 Some Perceived Benefits of Expert Systems: Responses of 86 Users of Knowledge Engineering Tools

1. Replicate expertise
2. Preserve expertise
3. Increase productivity and cost savings
4. Free human experts for more demanding problems
5. Provide expert consultations to inexperience staff

SOURCE: Bauman (1984)

TABLE 2 Some Problems for which Expert Systems are Providing Solutions

RISK ASSESSMENT	St. Paul Insurance Co. -- assess risk of insuring large commercial clients
EQUIPMENT DIAGNOSIS	General Motors -- determine causes of vibration noises and recommend repairs
CONFIGURATION	Digital Equipment Corp. -- translate customers' orders for computer systems into shipping orders
LOADING	U. S. Army -- design loading plan of cargo and equipment into aircraft of different types
ROUTE PLANNING	U. S. Air Force -- plan an aircraft's route from base to target and back to avoid detection and threats when possible
DATA INTERPRETATION	Schlumberger -- interpret down-hole data from oil well bore holes to assist in prospecting
SCHEDULING	Westinghouse -- plan manufacturing steps in a plant so as to avoid bottlenecks and delays
THERAPY MANAGEMENT	Stanford Medical Center -- assist in managing multi-step chemotherapy for cancer patients

MONITORING	IBM -- monitor operations of MVS operating system
SCREENING	U. S. Environmental Protection Agency -- determine which requests for information fall under the exceptions to the Freedom of Information Act
PORTFOLIO MANAGEMENT	First Financial Planning Systems (Travelers Insurance) -- analyze an individual's financial situation and recommend types of investments
TROUBLESHOOTING MANUFACTURING STEPS	Hewlett Packard -- diagnose causes of problems in photolithography steps of wafer fabrication
CROP MANAGEMENT	Virginia Polytechnic Institute - <POMME> Assist in managing apple orchards
EUIPMENT DESIGN	Delco -- design special-purpose, low voltage electric motors
TRAINING	Elf Aquitaine Oil Company -- demonstrate reasoning to find cause of drill bit sticking in oil well and to correct the problem

SOFTWARE CONSULTANT	Shell Oil Corporation -- advise persons on which subroutines in large FORTRAN library to use for their problems and how to use them
EQUIPMENT TUNING	Lawrence Livermore National Laboratory -- specify parameter settings to bring a sensitive instrument (triple quadrupole mass spectrometer) into alignment

NOTE: Many more examples are listed in Buchanan (1986), and Harmon (1986).

TABLE 3 Some Commercially Available Shells for Building Expert Systems

S.1	Teknowledge
KEE	Intellicorp
Knowledge-Craft	Carnegie Group
ART	Inference Corp.
LOOPS	Xerox
Personal Consultant	Texas Instruments
M.1	Teknowledge
ESDE	IBM

TABLE 4 Percent Time Per Quarter

	Q1	Q2	Q3	Q4
expert	75	75	100	75
sr.KE	100	100	100	100
jr.KE	100	100	100	100

NOTE: Approximate percentage of time required from an expert, a senior knowledge engineer, and a junior knowledge engineer to build a hypothetical small system over four quarters of a year. The two main variables in determining the amount of time required are the nature of the problem and the definition of the deliverable.

EXPERT SYSTEMS AND THEIR USE

A COMMENTARY ON THE

MITCHELL AND BUCHANAN PAPERS

Allen Newell

Department of Computer Sciences
Carnegie-Mellon University
Pittsburgh, Pennsylvania

EXPERT SYSTEMS

A COMMENTARY ON THE MITCHELL AND BUCHANAN PAPERS

Bruce Buchanan gave us a broad view of expert systems and showed a rather large collection of aspects across the whole field that need to be worried about to make the advances NASA needs. This leads to a point I want to make, which concerns my own concern about whether research is really needed on some parts of expert systems.

As preparation, Figure 1 shows my current favorite diagram to explain AI. You need to understand about AI that there are two dimensions in terms of which to talk about the performance of systems. The first is the amount of immediate knowledge that they have stored up, that they can get access to. This can conveniently be measured by the number of rules. The second is the amount of knowledge that they obtain by exploring the problem. This can conveniently be measured by the number of situations examined before committing to a response. Thus, there are isobars of equal performance, with better performance increasing up towards the northeast. You can roughly locate different intelligent systems in this space. Expert systems are well up on the immediate-knowledge scale, without much search. The Hitech chess program, which has a little, but not very much knowledge, lies far out on the search dimension. The human being is substantially above the expert systems on the knowledge dimension. Also, most expert systems do less search than humans do. The whole point of this diagram is that, in the current era, expert systems are an attempt to explore what can be achieved without very much search and reasoning, but with a modest amount of immediately available knowledge.

If you accept the characterization of expert systems in the figure, then even without all the research that Bruce was talking about, there exists an interesting class of programs, even though it is very limited in capability. The expert systems of today constitute a class of programs that appears to be very useful if you limit the tasks to the right kinds. Bruce was helping to characterize that. We actually know a modest amount about this type of task. If you have the right knowledge assembled, then you know what to do and how to do it without very much involved reasoning. For such tasks and their expert systems, it is not clear that the big need is to do a lot more research. The big issue is to build lots of these systems for lots of these tasks. What is needed is more like a development effort, to find out which tasks can successfully be done with modest amounts of expertise. The need is not to build any more expert-system shells, or to build more tools. The need is to pour all of the effort into finding out, in the plethora of space-station tasks, which are the ones that the current level of technology really does provide interesting and useful solutions.

Tom Mitchell talked much more specifically than did Bruce about the fact that the space station is a physical system -- that if you want to use expert systems and AI systems, they had better interact directly with physical devices. I agree absolutely that this is a major issue and a very important one for NASA to research. In particular, bringing control theory and symbolic reasoning together so we understand those as a single field is important. What I would like to emphasize is how little we know about that. In some respects we do not even know the units to use to talk about it, or how such symbolic programs ought to interact with control systems.

To bring this point home, let me note that a lot of current effort in understanding the human motor system is directed toward exploring a kind of system which is not controlled in detail. A particular dynamic system that has the right properties is composed, and is sent off to do a motor action. A good example is Hollerbach's model of handwriting, in which the whole system is composed of simply-interacting dynamic subsystems, which continuously draw letter-like curves, which are then modulated for specific letters. These dynamic systems are not cast in concrete. They are created and torn down in seconds, in order to compose and recompose dynamically according to short-term task requirements. The motor units that the cognitive system interacts with are these composed dynamic systems. We know almost nothing about such systems. When we finally understand something about it, I suspect it will change our notion entirely of the interface between the symbolic system and the dynamic system. The point is that there is a lot of research before we even get a clear idea clear about how symbolic systems ought to interact with mechanical and dynamic systems.

Tom made a suggestion about emulating devices. If a device breaks, then the emulation can be plugged in. I think this is an intriguing idea and there may be a whole world of interesting research in it. You might counterargue that, if this is possible, then everything might as well be run in computer mode. But there is a real reason not to do that. Making the emulation work may take a lot of computing power. A principal reason for using real physical devices and not simulating everything is that your system runs faster if you do not simulate it. But that does not imply that, if one device breaks, you cannot bring to bear an overwhelming amount of computational capacity to try to compensate for it. Thus, the system is prepared to emulate everywhere, but only has to do it in one or two places on any occasion. Emulation provides a backup capability. In fact, it is never likely to be as good, but at least it will be better than having to shut down the whole system. I think this is an interesting path of research, which could be pursued a long way. In particular, the feature that Tom mentioned about thinking of ways to construct systems so that they are decomposable and emulatable might yield many interesting possibilities.

Tom also raised the issue of sharing responsibility. However, he did not in fact tell us much about how tasks should be shared. Rather he described a particular aspect of the issue, which suggests that the machine ought to learn from the human, and then, quite properly, that the human ought to learn from the machine. I approve of both of these activities, but they beg the whole question of sharing. They do not elaborate ways of sharing, but both spend a fair amount of their time simply learning to be like each other, and confusing who really has the knowledge and who really knows how to do what. In fact, if one has machines with this kind of capability, the entire question of what it means to share may get transformed. It will become extremely difficult to quantify or be precise about who knows what, who ought to do what, and even who is doing what in the space station. There exists a kind of complementarity, in which the more you spread capabilities around in the system, so that there is a lot of redundancy, the less possible will it be to characterize the role of system components effectively -- to say for instance what the separate contributions are to the productivity of the total station. All I want to observe is that such systems are not clean, and learning and performance get confused. However, even though they are not clean, they may turn out to be the kind of system one has to build in order to get the margins of safety that are needed in space.

Finally, I want to talk about the issue of robustness, although it was not a major focus of either speaker. It is a fact, I believe, that there has been essentially no work on making expert systems robust. There is much attention, of course, to their giving explanations. But fundamentally expert systems are collections of rules, which are ultimately brittle and unforgiving. The lack of attention to robustness arises, in part, because there is a market for programs that are not very flexible or very robust. They can nevertheless, be successful. They will be increasingly successful, especially if the problem is turned around by saying 'I've got this hammer; where are interesting things to hit with it?' As a result, the expert systems field is not focused on solving the problem that I think NASA has to get solved, which is that it cannot use expert systems in space unless we understand how to build robust expert systems.

A research program in robust expert systems could be fielded by NASA, and I would certainly recommend it. Given requirements on robustness, one could explore more redundant rule sets or the provision of greater backtracking and reasoning mechanisms. There are many approaches to robustness and reliability that have their analog in expert systems and could provide guidance.

However, I think something more basic is at stake. What is really wrong here is the whole notion of laying down code -- or rules, which play the role of code for existing expert systems. That is, as soon as you lay down code, it becomes an echo from the past, unadapted to the future. You have become subject to a mechanism. Code is blind mechanism, complex perhaps, but blind. The important thing about a blind mechanism is that it does not care. A bullet does not care who it kills. A broken beam does not care on whom it falls. The horror stories about non-robust software almost invariably reflect the fact that code was laid down in the past, in a fantasy land of what was going to be, and something different happened at run time, for which the code was not adapted.

The problem, I believe, is that the unit, the line of code, is wrong. A clue for what might be right comes from the database world, with its adoption of transaction processing. It was concluded that the wrong thing to do was to take a line of code to be the unit. What had to be done was to package the specification of behavior in a hardened form called the transaction, for which some guarantees could be made. This has the right flavor of having changed the nature of the unit to make real progress. It has the wrong flavor because the unit is still just a little mechanism.

Somehow, in the area of robustness, the smallest unit of action has got to be, if I can use a metaphor, a caring piece of action. It has to be an action, which has a big enough context, even in its smallest unit, to react in terms of the global goals of the system, so it can care about safety and can care about the consequences of what it is doing. Somehow we have to find out how to create units that have that property. The units cannot be rules or code and so forth, which are just mechanisms. I think NASA ought to go after that. It would be a great research project. It is my contribution to this symposium of a really basic research goal that has an exceedingly small chance of succeeding, but an immense payoff if it does.

NEWELL
DISCUSSANT

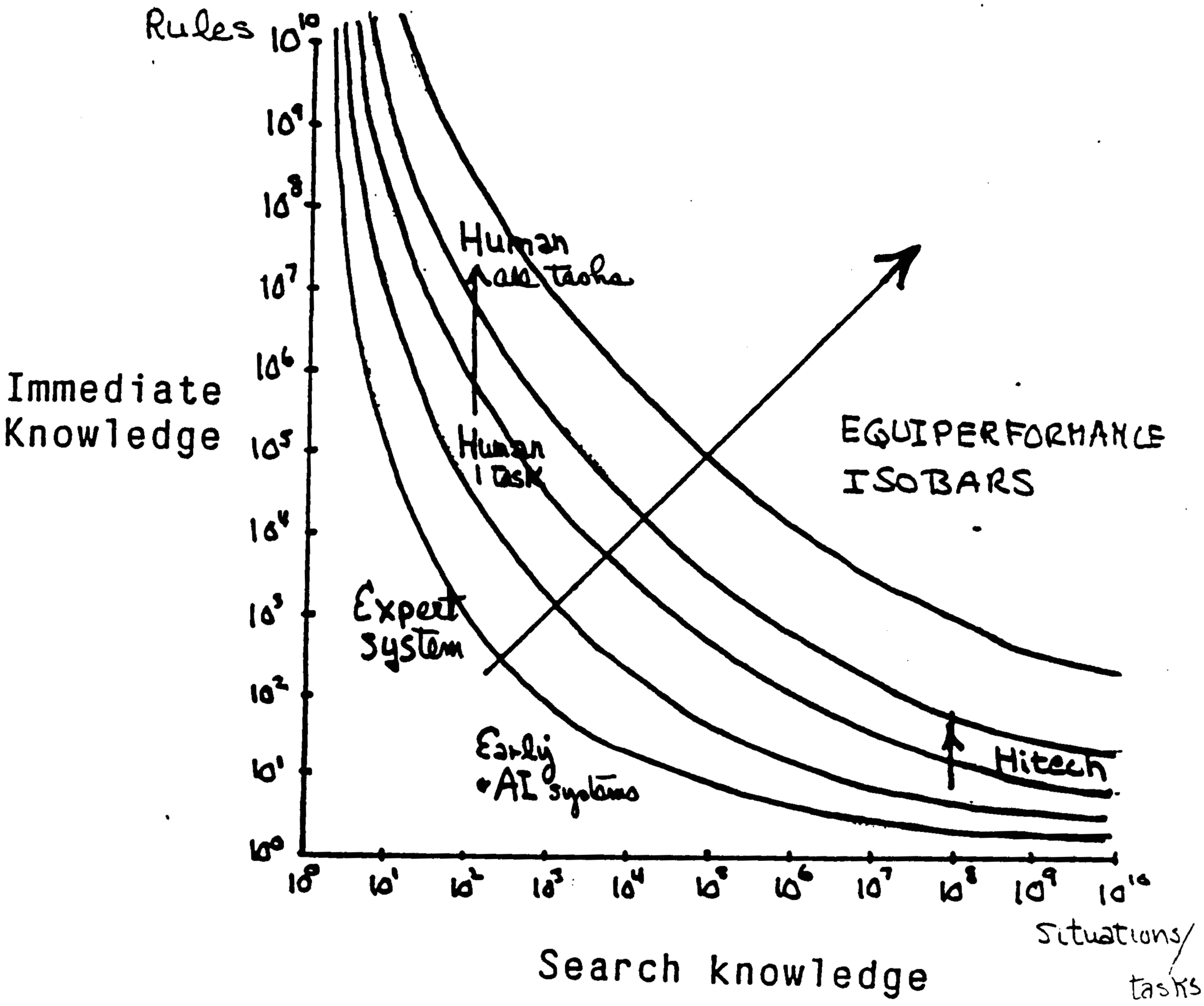

Figure 1.

DRAFT

NOT FOR DISTRIBUTION, ATTRIBUTION OR QUOTATION

SESSION 2:

EXPERT SYSTEMS AND THEIR USE

SYNOPSIS OF GENERAL AUDIENCE DISCUSSION

Allen Newell

EXPERT SYSTEMS AND THEIR USE

SYNOPSIS OF GENERAL AUDIENCE DISCUSSION

Concerns of several varieties were expressed about the knowledge engineering aspects of expert systems. Members of the audience with direct experience with developing expert systems gave these remarks special cogency. Expert systems seem to work better where good extensive formulations of the knowledge base already exist. Attempting to develop that knowledge base as part of the expert system effort often fails. The domains of expert systems are often exceedingly narrow, limited even to the particularity of the individual case. Given the dependence of the knowledge in expert systems upon the informants, there exists a real danger of poor systems if the human experts are full of erroneous and imperfect knowledge. There is no easy way to root out such bad knowledge.

On this last point it was noted that the learning apprentice systems discussed in Mitchell's paper provide some protection. The human experts give advice for the systems to construct explanations of the prior experience, and what the systems learn permanently is only what these explanations support. Thus the explanations operate as a filter on incorrect or incomplete knowledge from the human experts.

Concern was expressed about when one could put trust in expert systems and what was required to validate them. This was seen as a major issue, especially as the communication from the system moved towards a clipped "Yes sir, will do". It was pointed out that the issue has exactly the same complexity with humans and with machines, in terms of the need to accumulate broad-band experience with the system or human on which to finally build up a sense of trust.

Trust and validation are related to robustness in the sense used in Newell's discussion. It was pointed out that one path is to endow such machines with reasoning for validation at the moment of decision or action, when the context is available. This at least provides the right type of guarantee, namely that the system will consider some relevant issues before it acts. To make such an approach work requires providing additional global context to the machines, so the information is available on which to make appropriate checks.

Finally, there was a discussion to clarify the immediate-knowledge vs search diagram that Newell used to describe the nature of expert systems. One can move along an isobar, trading off less immediate-knowledge for more search (moving down and to the right) or, vice-versa, more immediate-knowledge for less search (moving up and to the left). Or one can move toward systems of increased power (moving up across the isobars) by pumping in sufficient additional knowledge and/or search in some combination. The actual shape of the equal-performance isobars depends on the task domain being covered. They can behave like hyperbolic asymptotes, where further tradeoff is always possible at the cost of more and more knowledge (say) to reduce search by less and less. But task domains can also be absolutely finite, such that systems with zero search are possible, with all correct response simply known. For these, there comes a point when all relevant knowledge is available, and no further addition of knowledge increases performance.

DRAFT
NOT FOR DISTRIBUTION,
ATTRIBUTION
OR QUOTATION

SESSION 3:

LANGUAGE AND DISPLAYS FOR HUMAN : COMPUTER COMMUNICATION

Paper: Philip Hayes, Carnegie-Mellon

Paper: Peter Polson, University of Colorado

Discussant: Judith Reitman Olson, University of Michigan

C-4

CHANGE IN HUMAN-COMPUTER INTERFACES ON THE SPACE STATION: WHY IT NEEDS TO HAPPEN AND HOW TO PLAN FOR IT

Philip J. Hayes

Carnegie Group Inc.

and

Computer Science Department

Carnegie-Mellon University

Pittsburg, Pennsylvania

and methods for dealing with the need for change in interfaces that will thus be established.

[p. 278 DOES NOT EXIST]

CONTENTS

OVERVIEW

The space station is unique in the history of manned space flight in its planned longevity. Never before have we had to deal with a manned space system that was expected to perform for twenty five years or longer. The implications of this requirement are far-reaching. This paper attempts to explore some of those implications in the area of human-computer interfaces.

The need for hooking (designing software for future extension and modification) is already well established[1] in the space station program as a whole. The paper explores in some detail why hooking is an important requirement for human-computer interfaces on the space station. The reasons are centered around the rapid rate of expansion in the kinds and combinations of modalities (typing, graphics, pointing, speech, etc.) available for human-computer interaction and in the interaction and implementation techniques available for them. Many of these modalities and associated interaction techniques are well-developed, others are in embryonic stages. Different modalities (or combinations of modalities) are appropriate to different situations. The paper therefore also looks at the appropriateness of the modalities according to task, user, and the space station environment. An appropriate matching of interface modalities, task, and user is essential to maximizing the potential of on-board computer systems in their primary goal of supporting and amplifying human abilities.

A second rationale for providing hooking in human-computer interfaces is related to the currently developing possibilities for intelligent interfaces. So the paper discusses methods of achieving intelligence in interfaces, and in what circumstances it is desirable. The issue of intelligence is also related to the distinction between conversational/agent type systems and machine/tool-like systems. The current culture at NASA is highly oriented towards the latter. The paper explores the tradeoffs between the two approaches and discusses the circumstances in which a more conversational/agent style system could fit space station goals and NASA culture.

After examining the need for hooking in human-computer interfaces, the paper turns to the question of how to achieve it. The discussion here centers around methods of achieving a clean separation between the interface and the underlying application (space station system) it interfaces to. The key advantage of this kind of separation is that it allows the interfaces to be changed independently of the applications, so that a new interface (possibly employing different modalities from the old one) can be rolled in without altering the application in any way. In an environment such as the space station where the underlying applications may be complicated, mission critical, and highly integrated with other applications, such separation becomes all the more important.

The feasibility of a completely clean separation between interface and application is unclear at the moment. The question is currently being addressed by the major subarea of human-computer interaction that deals with user interface management systems (UIMSs). Unfortunately, it is infeasible to wait for research on this topic to reach full maturity. Unless the original applications and interfaces are built with separation in mind, retrofitting separation is likely to be impossible. So the paper discusses what kind of interface/application separation is feasible for the space station initial operating capability (IOC), and looks at how this will constrain the overall possibilities for human-computer interaction.

Separation of interface from application has two other important advantages in addition to hooking. First, it promotes consistency between interfaces to different applications. Most of the work on UIMSs emphasizes a common set of tools for construction of the separated interfaces, and this inevitably leads to considerable consistency of (at least fine-grained) interface behavior between interfaces. The importance of consistency in interfaces has been appropriately emphasized by Polson in the preceding paper. Secondly, the hooking made possible through separation also makes it easier to alter interfaces during their initial development. The only effective way of developing excellent human-computer interfaces is to build interfaces, see how users perform, and then repeatedly alter them to deal with problems. This process is much more effective if the interfaces are easy to modify. The paper explores these two other aspects of interface/application separation further.

APPROPRIATE INTERFACE MODALITIES

The need for change in human-computer interfaces on the space station and the consequent need for hooking arises out of the rapid development that has occurred and continues to occur in interface modalities (typing, graphics, pointing, speech, etc.) and the interaction techniques used with them. This section discusses what interface modalities (or combinations of modalities) and techniques are appropriate for different kinds of interface tasks. An appropriate matching of interface modalities, task, and user is essential to maximizing the potential of on-board computer systems in their primary goal of supporting and amplifying human abilities.

Interface Requirements for the Space Station

The basic considerations in designing good human-computer interfaces for the space station are the same as for any human-computer interface on Earth. In particular, the interfaces should be:

- easy to learn
- easy to use
- efficient to use

Much has been written, e.g. (Hansen, 1971), about this and similar lists of attributes. For present purposes, we can treat them as self-evident, though of different relative importance in different interface situations. There are, however, some special characteristics of the space station environment that require further discussion before looking at the relative utility of the different available interface modalities. These characteristics include:

- Weightlessness: In addition to being the most obvious special characteristic of the space station environment, zero-g causes specific problems for human-computer interfaces. The problem is that movement by humans in a weightless environment induces other movement. This is particularly true if the movement involves pressure against another object, such as in typing or pointing on a touch sensitive screen, but it is also true for any kind of gesture, such as with a non-touch light pen. A person employing such interface modalities will tend to drift away from or change orientation with respect to the workstation he is using. The simplest solution to involuntary movement induced by human-computer interaction is simply to tether the user physically to the workstation. This, however, has the obvious disadvantage of inconvenience, especially if the interaction session will not last long. Also, the tethering would have to be relatively complex and therefore intrusive to solve completely the problem of changing orientation.

- Analogue/continuous interaction: Many interactions on the space station require (or could benefit from) command input which can be given rapidly and/or in an analogue/continuous manner. Obvious examples include any kind of docking or remote manipulation activity. Less obvious ones include manipulation of continuous variables in, for instance, systems controlling the life-support environment. Analogue/continuous interactions require different kinds of interaction modalities and techniques from those used in more traditional computer command languages.

- Varied groups of users: Although the most mission-critical systems will continue to be operated by highly trained personnel, the sheer number of systems likely to be available in the space station suggests that this will not be true for all systems. Some less mission-critical or time-critical systems in, for instance, the areas of personal comfort, provisioning, or inter-crew communication, are likely to have to interact with users of varying degrees of sophistication and experience with respect to those systems. To avoid negative transfer effects between different systems, interfaces need to be as consistent as possible across the various systems. To deal with users who are inexperienced (for that system), interfaces also need to be as self-evident, self-explanatory, and self-documenting as possible. The goal should be for experience with some subset of the non-mission critical systems and appropriate knowledge of the domain the system deals with to serve as sufficient experience for the accomplishment of straightforward tasks with any of the other non-mission critical systems.

- Hands-free operation: There are many situations in the space station environment in which hands-free interaction would be useful. An obvious example is extra-vehicular activity, but more frequent examples might arise when it was important to avoid the induced motion problems mentioned above (in the weightlessness bullet) or when it was useful to have an additional I/O channel in the context of a complex hands-on analogue activity such as remote manipulation. The most natural hands-free modality is speech, but other possibilities include control through eye-movement, or in specialized circumstances use of feet or other body parts.

Having looked at some of the space factors which might influence choice of interface style and modality, we now look at the appropriateness and range of applicability of the various modalities. Some of the discussion presupposes certain styles of interface for each type of modality. The presuppositions are not always necessarily valid, but are characteristic of the way the modalities have typically been used.

Character-Oriented Interfaces

The vast majority of human-computer interfaces currently in use are character-oriented. The users of these interfaces provide input by typing on a keyboard, and the systems provide output through a screen with a fixed number of character positions (typically 24 lines of 80 characters). Interfaces of this kind do not have a great deal to commend them for the space station environment. Reasons include:

- The physical pushing motion involved of typing leads to the induced motion problem mentioned above. Typing sessions of any length require some kind of tethering arrangement.

- Typed input is unsuitable for analogue/continuous interaction.

- In character-oriented interaction, the user typically issues commands through expressions in a line-oriented artificial command language). Such languages generally require significant learning effort, making them difficult to use for initial or casual users. Some command languages, such as the one for DEC's Tops-20 operating system, have shown that it is possible through uniformity and carefully thought out help facilities, to reduce the difficulty of use by non-expert users. However, command line interaction is inherently more limited in its perspicuity than the direct manipulation style described in the section titled "Graphically-Oriented Interaction".

- Although some of the learnability and ease of use problems with command-line interaction can be overcome through selection from menus from the keyboard, this can be seen as an attempt to overcome the limitations of the modality by use of a interaction technique borrowed from another modality, i.e. pointing input. It seems more appropriate to use the pointing modality directly.

- Character-oriented interaction is essentially an old, though very well worked out (see e.g. Martin, 1973), technology.

Graphically-Oriented Interaction

A recently developed and increasingly popular style of interaction is based on the use of a high-resolution graphical display and a pointing device such as a mouse or joystick. A well known system exemplifying this scheme is the Macintosh personal computer (Williams, 1984). Interaction in this style is based on techniques such as menu-selection, icon selection and movement, and other kinds of graphically-oriented operations. This style of interaction is also known as direct manipulation (Hutchins et al., 1986; Shneiderman, 1981), indicating ideally that the user should feel that he is directly manipulating the objects represented by the computer system. An example of this kind of direct manipulation analogy is deleting a file by using a mouse to "pick up" the icon representing the file and move it into an icon depicting a wastepaper basket.

There are many interfaces that are graphical in nature, but fall well short of the ideal of direct manipulation of providing the user with the illusion of operating directly on the "world" of the underlying application. Interfaces that rely on menus, for instance, often do not support such an illusion. Interaction will have more of the flavor of direct manipulation if the user can perform an operation by moving an icon, for instance, as in the file deletion example above, than by selecting the name of the operation from a list in a menu. To the extent that they can be maintained, the metaphors implicit in direct manipulation interfaces make the interfaces more easily learnable, and reduce the need for help systems. This is important for the varied groups of users that will be using non-mission-critical systems. The Xerox Star (Smith et al., 1982) and Macintosh (Williams, 1984) have given some idea of what is possible in this line in the office and personal computing arena. More research is needed to provide more interaction metaphors on which to build direct manipulation interfaces. The creation of such metaphors will be aided by the existence of new and innovative I/O devices (see section titled "Novel I/O Modalities").

Graphically-oriented or direct manipulation interfaces are in many ways superior to character-oriented interfaces for the space station environment, but there are still some deficiencies. In particular, some of the standard pointing devices used on earth are not well adapted to a weightless environment. This is particularly true of the mouse which is intended to be used on a flat surface under the influence of gravity. The lightpen and the tracker ball both require pressure against a surface and so have an induced motion problem. The joystick may be better adapted from the point of view of induced motion since it requires that the user grip it to manipulate it. This raises the possibility that correction of the motion induced might be possible through the user's grip. However, there are obvious problems with this approach for fine-grained movements, but there is a great deal of experience with the use of joysticks in weightless environment from such tasks as remote manipulation.

A better approach may be solved by further development of innovative pointing devices specifically aimed at use in a weightless environment. One possibility is a freely movable hand-held "mouse" which induces 2-D motion on a screen. Of course, the full six degrees of freedom of motion with such a device also open up the possibility of control of three-dimensional simulations or real actions. Devices of this kind are available and investigations into their use and refinement should be encouraged.

Another innovative kind of pointing technology even better adapted for space is eye tracking. Eye tracking has the dual advantages of no significant induced motion and hands-free operation. It has the disadvantage of intrusive apparatus. It may be particularly appropriate for activity in a space suit where the eye-tracking apparatus can be incorporated into the helmet with no increment in discomfort or inconvenience. Further work is needed both to develop less intrusive forms of eye tracking and on the use of eye tracking control in extra-vehicular activity.

Earth-based direct manipulation interfaces generally operate within the context of fixed workstations. While there are many space station tasks for which this is perfectly appropriate, there are others where a more portable arrangement is required or preferable. EVA is the most common, but other examples include inventory, inspection, and communication tasks. Work on in-helmet displays is needed for EVA to complement the work on eye-tracking. Other work on hand-held or otherwise portable display and pointing devices is needed for the on-board tasks requiring mobile interactive devices.

Natural Language Interaction Via Keyboard

Typed natural language input and output is not a modality in its own right, but a variation on character-oriented interaction. However, it is sufficiently different from typical command language interaction that it is worth considering separately.

A low-level, but nevertheless significant, artifact of the redundancy of human language is that natural language will usually require many more keystrokes than a command language designed for a specific interaction task. This means that the remarks above about the undesirability of the significant amounts of typing involved in command language interaction apply with greater strength to typed natural language interaction. Also for rapid interaction or interaction with an expert user, the amount of typing involved typically makes natural language interfaces unacceptably slow.

Natural language interaction, however, has the important advantage over command language interaction that it allows the user to express things in a way that is natural for him, rather than having to learn an artificial (and frequently arcane) command language. It is thus more suitable for casual users and could help to meet the goal of making a wide variety of space station systems accessible to many different users of varying skill levels.

This argument in favor of natural language interaction presupposes that the interfaces can handle any form of expression that a user cares to come up with and is relevant to the underlying application. At the current state-of-the-art, this is an invalid assumption. In practice, natural language interfaces fall well short of full coverage on syntactic, semantic, and pragmatic grounds, even for the restricted domain of discourse implied by a specific underlying application. This leads to the habitability problem (Watt, 1968) in which many of the advantages of naturalness and lack of learning disappear because the user has to learn what is still essentially a subset of English (or whatever natural language is being used) artificially restricted by the limitations of the natural language processing system. This problem can sometimes even make the language more difficult to learn than a simple command language because the limitations are less easy for the user to identify and remember. On the other hand, these problems can be minimized by appropriate human engineering for interfaces to appropriately limited applications. However, this is very time-consuming and expensive at the time the interface is developed since it involves detailed observations of many users interacting with the system and repeated extensions of the natural language coverage until all the commonly occurring syntax, semantics, and pragmatics are handled.

Perhaps the most important reason for not using natural language interaction is that most interaction can be handled more easily by direct manipulation or other graphically-oriented means. Moreover, as the section titled "Graphically-Oriented Interaction" points out, graphical interaction is likely to be more suitable for the space station environment than character-oriented interaction in general. Whenever the user is trying to select between a limited number of alternatives or is trying to manipulate objects or access information that can be presented to him in an intuitive spatially-distributed manner, then natural language interaction (or any other form of keyboard interaction) is likely to prove inferior to graphical interaction. There are, however, some circumstances in which natural language or command language interaction is preferable to graphical interaction, including:

- o When there is a large range of options to choose between, especially when the options can be composed in a combinatorially explosive kind of way;

- o When there is no convenient way to distribute the information in a two-dimensional space;

- o When a suitable spatial distribution exists, but the resulting space of information is so large that only a small fraction of it can be presented to the user at any one time;

- o When the user is looking for information that is distributed across several spatially-distinct items, so that retrieval of the information by direct manipulation would require iterative examination of each of the relevant interface components.

These conditions are not true for most interactive situations, but come up frequently enough for natural language to be considered as a secondary mode of interaction for many applications to supplement a largely direct manipulation interface. To be effective in this role the natural language interaction has to be suitably integrated with the direct manipulation interaction. Some work has been done in this area on how to use visual context to help interpret pronouns and other anaphoric and deictic references by the user and also to allow intermixing of pointing and natural language input (Bolt, 1980; Hayes, 1987a). However, integrated natural language and graphical interfaces could provide significant benefits given an appropriate research effort.

Speech Interaction

Although a combination of typed natural language and graphical interaction offers some attractive advantages, natural language interaction through speech offers many more. While the habitability problems mentioned in the section titled "Natural Language Interaction Via Keyboard" remain, spoken input is much more rapid and natural than typing the same words. Moreover, the voice and ears offer channels of communication quite separate from the hands and eyes. Speech input leaves the hands free and speech output leaves the eyes free for other tasks (either computer interaction or interaction with the physical world).

In terms of suitability for speech interaction, the space station environment has one specific advantage and one specific disadvantage. The advantage is the absence of any need for speaker-independent speech recognition. At the present state-of-the-art in speech processing, considerably better results can be obtained if the speech recognition system has been trained in advance on the specific characteristics of a speaker's voice (through recordings of the speaker saying a predetermined set of words several times). Given the relatively small number of people that will be on-board the space station at any given time, their relatively long training period, and their relatively long stay, such system training is unlikely to be a problem. The specific disadvantage of the space station environment is the relatively high level of ambient noise that can be expected inside it, at least if the experience of the Shuttle is a guide. Ambient noise is problematic for speech recognition. At the current state-of-the-art, resolving this problem would probably require the use of a close-speaking microphone of some kind. This itself has the disadvantage of being intrusive and inconvenient to take off and put back on.

The current state-of-the-art in speech processing is still fairly limited. In addition to the speaker-dependent and ambient noise limitations mentioned above, the better commercially available systems tend to be able to handle only small vocabularies (less than a thousand words is typical) and pauses between each word or group of words that the system recognizes as a lexical units (so-called connected speech recognition, as opposed to continuous speech recognition in which no pauses are needed). However, this is a field where rapid advances are occurring and new commercial developments plus a very active academic research program are pushing back all of these limitations. In fact, speaker-independent, large (10,000 word plus) vocabulary, continuous speech recognition in noisy environments is likely to be available within the lifetime of the space station, and systems in which a subset of these restrictions have been relaxed are likely in the early part of the space station's lifetime.

Given these prospects for advancement and the inherent advantages of speech interaction, it seems natural for NASA both to plan on a significant role for voice in space station human-computer interfaces and to keep track of or actively support research on speech processing. Nevertheless, even if the underlying speech technology advances as projected above, other problems remain that will require solution before speech can make its full contribution to human-computer interaction on the space station.

First, speech interaction on its own is quite unsuitable for some kinds of interaction, particularly analogue/continuous commands -- it would be very difficult to control a remote manipulation device through a series of "left a bit", "down a bit" kinds of commands. Moreover, even in situations where speech could be used, such as the specification of discrete commands in an inventory tracking system, it may not always be the preferred mode of interaction. For instance, if the arguments to a particular command all have relatively complex verbal descriptions, but there are only four of them, it is probably simpler, more mnemonic, and more reliable to let the user input the argument by pointing at a menu or set of icons representing them. Both of these situations indicate the need for techniques for integrating speech interaction with other modalities including pointing and 3-D manipulation. Speech can then be seen as a complementary channel for issuing discrete commands during continuous/analogue manipulations while both hands are occupied, such as releasing catches during a remote manipulation task. It can also be seen as a supplementary channel for issuing whatever commands or portion of commands are convenient during a discrete command interaction, and as a stand-alone interaction medium for discrete commands whenever hands-free operation is necessary or convenient. Many of the same research issues arise in integrating speech with other modalities as were described in the section titled "Natural Language Interaction Via Keyboard" for the integration of typed natural language and graphical interaction. These issues include resolution of deictic phrases ("this one", "that") and other pronouns, use of the user's visual context in interpreting what he says, and methods of combining input from pointing and speech to form a single command. Although interesting explorations have already been undertaken in this area (Bolt, 1980; Hayes, 1986), these issues all require further research.

In addition to problems of integration with other input modalities, speech interaction raises some interesting problems of its own related to managing the dialogue between human and computer. The first problem concerns when the computer should listen, i.e. when it should try to interpret the speech that its users are producing. The users will speak to other people (or sometimes to themselves) as well as to the machine and attempts by the machine to interpret speech not directed at it is only likely to cause trouble. Techniques that have been explored here include physical switches (typically foot switches on Earth) or switches based on key phrases (such as "listen to me" and "stop listening") that have to be uttered to start and stop the machine trying to interpret speech. These devices are clumsy and detract from the feeling of naturalness that spoken interaction should provide, but will probably be necessary until speech systems become sophisticated enough to make positive determinations that spoken input is not being directed at them. The prospect of such an ability is well beyond the horizon of current research.

Another dialogue issue with special implications for speech is that of ensuring reliable communication. An interactive speech interface must ensure that it understands the user accurately; that the user is confident of this; that the user becomes aware when the system has failed to understand correctly; and that the user is able to correct such errors when they arise. Humans have developed sophisticated conventions (Sacks et al., 1974; Schegloff et al., 1977) for ensuring that communication is indeed robust in this way. Unfortunately, many of these conventions rely on a level of understanding and intelligence that is unrealistic for machines. However, to have smooth conversations, ways must be found to perform the above functions that are both suitable for the limited intelligence of current machines and fit reasonably well with human conventions. A limited amount of work has been done in this area e.g., (Hayes and Reddy, 1983), but much more is needed.

Finally, there is the same problem of habitability that arises for typed natural language interfaces. For speech, however, the problem can be even worse since the user is less well able to be deliberate and precise in his choice of words and phrasings while speaking than while typing. Moreover, when speech is used as a stand-alone human-computer interaction modality, there is no possibility of reminding the user through a display about the limitations of the domain of discourse or the phrasings that can be used. Work is needed here to find better ways of developing a reasonably habitable subset of a natural language for a restricted domain, to develop ways for the system to encourage the user to stay within the bounds of the restricted language through appropriate output of its own, to devise methods for partial understanding when a user strays outside the bounds of the restricted language, and to develop interaction methods for steering the user back on track when he does stray as he inevitably will.

Novel I/O Modalities

The interaction modalities discussed so far are conventional in the sense that they have already been widely used (this is least true of speech) in earthbound interfaces and other space systems. However, the numerous challenges posed for human-computer interaction by the space station and the recent emergence of some novel and innovative interaction modalities suggest that it is worthwhile also to consider some of these less-developed modalities for use in the space station.

An innovative input modality of potentially considerable utility on the space station is the use of gesture. The conventional use of a mouse or other pointing device in conjunction with a display screen is a limited form of gesture, but it is possible to sense and interpret a much broader range of human gesture by machine. Large scale gestures involving whole limbs are not practical for the space station because of the constraints of a weightless environment, but smaller-scale gestures are quite suitable. The least problematic form of gesture from the point of view of the induced motion problem is eye motion. As already discussed in the section titled "Graphically-Oriented Interaction", eye tracking can be used as a substitute for pointing via a mouse or other conventional pointing device. It is particularly well suited for use with in-helmet displays.

A more radical departure from conventional technology is the interpretation of hand and finger gestures. Technology is emerging) that will allow a machine to recognize a full range of small manual gestures made in restricted spatial context. There is a large range of gestures that have associated conventional meanings (such as yes, no, get rid of it, move it from place to place, etc.). This suggests that interfaces that accepted such gestures as input could be very easy and intuitive to learn and natural to use. It might even be possible to resolve any motion problems induced by gesturing through the use of balanced symmetrical gestures which employ two equal and opposite motions.

We have discussed two ways in which gesture can be used in innovative ways for computer input. There may well be others. In general, there is a need for imaginative exploration of the whole range of ways in which human movement compatible with a weightless, noisy environment can most easily be sensed by machine.

Another potentially promising area for innovation in interaction techniques involves output by means other than fixed screens and simple audio feedback. In-helmet displays hold significant promise in this direction. Although such displays are most natural in circumstances in which the user has to wear a helmet anyway, such as EVA, they can also improve human-computer interaction in other circumstances. Current investigations, including some at NASA-Ames, have shown the utility of in-helmet displays for presenting a complex 3-D world view to the user. This work involves the use of direct-eye projection, rather than an actual display screen inside the helmet. It provides the illusion of a 3-D world by sensing the direction in which the user's head is pointing and adjusting the projection accordingly. This is a good example of the kind of innovative work in novel interaction modalities that needs to be undertaken to exploit fully the potential for human-computer interaction on the space station.

Other kinds of novel output modalities on which further research could bring useful results include force or tactile feedback on joystick-type direct manipulation or analogue tasks and acceptably unobtrusive speech output. Force and tactile feedback has been used regularly in flying and remote manipulation tasks, but has been little explored for use in human-computer interaction for more abstract tasks, such as manipulating a set of computer files. Force or tactile feedback through a joystick on such problems could enhance the directness of the "feel" of direct manipulation interfaces and also be useful as an indicator of urgency, importance, or difficulty. Speech output has also been used before, but a recurring difficulty is getting the speech output to fit naturally into the flow of an interaction. Speech output is by its nature transitory and must be given at just the right point in the interaction and be repeatable by the user if desired. Moreover, the speech output should not occur so frequently that it becomes distracting to the user. Just as in the case with input modalities, much work is needed in the form of imaginative explorations over a large range of untried and speculative output modalities.

Finally in this section, we turn to the idea of expert interfaces, i.e. interfaces that require considerable expertise and training to operate, but offer high rates of very efficient interaction in return. The high degree of training that will be undergone by many space station personnel provides good opportunity for use of innovative expert interfaces, involving coordinated use of multiple limbs, eyes, etc. in multiple modalities for high efficiency interaction. Flying is best explored example of such an activity, and many of the techniques developed with flying have been successfully transferred to docking and other such maneuvers in space. Another source of ideas for expert interfaces can come from musical performance (Buxton, 1986). Players of such instruments as the organ learn after a long period of training to use all four limbs in a coordinated fashion to produce an enormously high rate of command input to the instrument. For interaction tasks that are important enough to justify the large training periods involved and could benefit from a high data transfer rate, interfaces which draw on the experience of flying and musical interfaces are well worth investigation.

INTELLIGENT INTERFACES

The need to plan for change in interfaces comes not only from the possibility for advances in interface modalities and the techniques used with them, but also from the increasing possibility of the development of intelligent interfaces. Intelligent interfaces are still a research area, rather than a set of proven interface techniques, but the potential benefits of truly intelligent interfaces in terms of ease of use make them an area worthy of investigation for future space station interfaces. Intelligent interfaces also fit very well with the increasing development of intelligent, autonomous application systems for space use. If an application exhibits intelligent task behavior, then it should also behave intelligently in its interaction with its user.

An initial fundamental distinction to be made in considering the potential of intelligent interfaces is the distinction between conversational or agent-like systems and tool or machine-like systems. Almost all current interfaces are of the tool/machine-like kind. Users of such systems accomplish a task by controlling a (hopefully) responsive, but essentially unintelligent system. Direct manipulation interfaces (see section titled "Graphically-Oriented Interaction") are the archetype of this kind of interface since they encourage the user to feel that he is directly controlling the world that the underlying system deals with. However, command language interfaces can also be thought of as tool/machine-like since they respond in predictable ways to a fixed set of commands. The user is left feeling firmly in control.

Conversational/agent interfaces, on the other hand, are intended to give the user an entirely different feeling. Users of conversational/agent systems are intended to feel that they are negotiating with a subservient, but intelligent, system. They accomplish their tasks through negotiation with and through the agency of the system, rather than through direct actions of their own. Conversational systems thus have much greater possibilities for intelligent interaction than machine-like systems. Conversational systems also do not fit well with the direct manipulation or command language styles of interface, but fit much better with natural language or speech interfaces which naturally lend themselves to a dialogue style. Interfaces to intelligent, autonomous application systems can also make good use of a conversational style of interaction.

The user of a conversational equipment reservation system might, for instance, request (in natural language) the reservation of a certain piece of equipment and then be engaged by the system in a dialogue concerning the period of the reservation and if the equipment was unavailable the possibility of substitute equipment or substitute times. The user of a tool/machine-like interface to the same underlying functionality would, on the other hand, expect to be forced to specify the reservation times through constraints on the interaction enforced by the interface. If equipment was unavailable at the desired time, he would also expect to have to initiate a search himself through alternative times and substitute equipment.

It is clear that the culture within NASA is very much oriented to tool/machine-like interfaces and moreover to interfaces in which the degree of control exercised by the user is very high. There are historical reasons for this related to the importance placed from early on in the space program (Loftus, 1986) on having as much human control as possible available so that there would be the maximum chance of fixing any problems that arose. As systems increase in complexity, the tool/machine-like interfaces have tended to reduce the amount of complexity (and therefore fine control) available to the user without, however, crossing over the line that separates tools from agents. At the current state of the art, this approach is entirely as it should be. There are no successful operational interfaces anywhere that could fairly be described as true conversational/agent systems[2] However, the promise of intelligent conversational systems remains. If this promise is successfully realized, then it offers an attractive way of achieving the goal of having a large variety of non-mission-critical space station system easily available to a broad class of users.

The key to the development of conversational/agent interfaces lies in the development of detailed models of the task and the user. To produce intelligent agent behavior, it is necessary to use Artificial Intelligence techniques to model what tasks the user can accomplish through the interface, how he can achieve his goals, and what his current goals and state of knowledge are. Previous work that has tried to do this includes (Huff and Lesser, 1982; Mark, 1981; Card et al., 1983).

This detailed level of modelling is necessary for intelligent agent-like behavior because, without it, the interface can only respond to the user's individual actions and the very local context. Using our equipment reservation example, knowledge of what purpose the user might be trying to achieve through use of a particular piece of equipment could allow the system to suggest a suitable alternative. Without that knowledge, the system can only respond on the availability of a particular piece of equipment.

This kind of modelling becomes much harder when the user is pursuing a goal that involves several system actions. An agent system then has to determine the nature of the higher level goal from observation of the individual actions. An electronic mail system, for instance, might observe that the user is trying to write a message out to a file and then use the contents of the file as the body of a message to another system user. If it recognized from this that the user was simply trying to forward the message to the other user, it could suggest an abbreviated method of doing so. Since individual system actions can often fit into many plans and since system users often interleave plans to achieve several goals, the detection of such larger scale goals out of lower level actions is a very hard task. A system that has such an ability can, however, assist the user in a variety of ways including suggesting simpler ways of doing things (as in the example above), warning about pitfalls that it can foresee could lead to the user's current plan not achieving his overall goal, offering to take over and complete the plan it believes the user to be following, or offering to perform the next action or actions in the plan whenever it becomes clear what they are.

The kinds of task and user modelling abilities mentioned above could be used in conjunction with any kind of interface, not just one that uses natural language. However, agent-like interfaces fit particularly well with natural language for two reasons. First, natural language is a natural medium for the kinds of negotiation that arise when a system is trying to respond to the goals it believes its user to have rather than direct commands. Second, the goal and task models themselves can be very useful in natural language and speech understanding. The biggest single problem in natural language processing is handling ambiguity of various kinds (syntactic, semantic, referential, etc.) and if one version of the ambiguity makes sense in the context of the other user model and the other does not, then the one that does not fit can be eliminated.

The whole area of conversational modelling is still in its infancy. Much work remains to be done to produce usable systems. However, progress in this field is necessary for truly intelligent interfaces, whether or not they are based on natural language. Given the potential benefits of intelligent interfaces to the space station, it is an area of research well worth pursuing.

The same kind of techniques that go into pure conversational systems can also be used in conjunction with more conventional interaction techniques to produce a hybrid kind of interface that incorporates both conversational/agent and tool/machine-like components. The basic flavor of such an interface is essentially tool/machine-like. The conversational component serves as medium through which the system and user can exchange comments about what is going on in the central tool/machine-like component. The user can also use the conversational component to instruct the system indirectly to perform actions or present information that he could perform or request directly (though perhaps more tediously) through the tool/machine-like component.

A system of this kind has several advantages. First, pure conversational systems are unsuitable for any task that can be performed effectively through direct manipulation techniques, and particularly for tasks that involve continuous/analogue interaction. Adding a conversational/agent component to a tool/machine-like direct manipulation interface for performing such tasks allows the basic task to be performed in the most efficient manner, but also allows components of that task that could benefit from a conversational approach to do so. Examples of conversational interaction in such a situation include: the user requesting information that would require multiple actions to retrieve through the direct manipulation interface; the user asking questions about how to use the direct manipulation interface component; the system volunteering information about more efficient ways to use the direct manipulation component; the user requesting the system to achieve a higher level goal that would require extensive interaction with the direct manipulation component.

A second advantage of this kind of hybrid system is that the conversational component does not have to be used at all if the user does not so desire. This kind of arrangement may be the best way to introduce conversational systems into a culture like NASA's that has good reason to be cautious about such systems. The unproven nature of conversational/agent systems suggests that they be introduced in a way that gives their user alternative methods of accomplishing all their tasks.

This kind of hybrid agent/machine-like interface requires the same technological underpinnings as pure conversational systems and hence the same research program. However, it also requires additional work on how to integrate the two components in a smooth way. Some work (Negronponte, 1981; Bolt, 1980; Hayes, 1987b) has already been done in this area, but much more is required.

PLANNING FOR CHANGE IN INTERFACES

The previous two sections have discussed some of the potential developments in interface modalities and techniques that will generate the need for change in human-computer interfaces during the life of the space station. In this section, we turn to the issue of how to deal with such change.

User Interface Management Systems

The essence of the approach discussed here is based on hooking, i.e. designing software for future extension and modification. The kind of hooking envisaged is determined by the assumption that it is unnecessary and probably infeasible to rewrite the underlying application systems whenever interfaces change. This means that the application systems need to be hooked in such a way that new interface systems can be developed for them without changes to the applications themselves. This in turn means that applications and interfaces must be written in as separate a way as possible with communication between them as narrow and as tightly defined as possible.

There is already a substantial body of work in the human-computer interaction literature on this kind of separation between application and interface, e.g. (Tanner and Buxton, 1983; Hayes and Szekely, 1983; Hayes et al., 1985; Wasserman and Shewmake, 1984; Jacob, 1984; Yunten and Hartson, 1984). The systems developed to achieve this kind of separation are known as user interface management systems (UIMSs). However, work to date is far from achieving a consensus on the best way to achieve the desired separation or indeed the degree of separation that is desirable, appropriate, or possible. This is unfortunate from the point of view of building the software for the space station IOC, since to achieve any useful degree of separation both interface and application must be built using a strict model of the kinds of communication that can occur between application and interface. Decisions made now on this kind of communication will affect the possibilities for interface/application separation for the life of the space station. Since research work in this area is far from reaching a conclusion about what is the best model of communication, whatever model is adopted now is likely to be considerably less than optimal. However, adopting some model may be better than none at all, so the remainder of this section reviews current research and future directions in the area of UIMS work.

The basic model adopted by most work on user interface management systems is shown in Figure 1. The user communicates with the UIMS which in turn communicates with the application. Communication between the UIMS and the application is achieved through a carefully defined protocol which limits the kind of interaction that can occur. A typical repertoire of communication events might include:

o request from the UIMS to the application to perform a particular operation with a certain set of parameters

- o notification by the application of completion of an operation
- o update by the application of a variable indicating progress towards completion of an operation
- o error message from the application
- o request from the UIMS for a check on the semantic validity of a proposed parameter for an application operation
- o reply from the application to such a request

The precise content of the messages that flow between UIMS and application is defined by a declarative data base, the Application Specification Data Base of Figure 1, which specifies what actions and operations the application is capable of.

This model is not the one adopted by the most usual approach to interface standardization, that of providing a set of standard subroutines for high-level interface actions, such as getting the user to chose a value from a fixed set by presenting him with a pop-up menu. A typical interface subroutine for this task might take a set of choices as a parameter and return one of the choices. The subroutine would take care of the details of presenting the user with the menu and interpreting his mouse movements in making a choice from it. A disciplined use of a comprehensive package of such subroutines can thus provide a significant degree of low-level consistency across applications that use it. However, it cannot provide some of the other advantages of the kind of separation between interface and application described above, as we shall see.

The kind of separation between application and interface shown in Figure 1 can allow the interface to change without any alteration to the underlying application, whether or not the interface is provided by a UIMS. A UIMS goes further by defining the behavior of the interface itself through another declarative data base (possibly integrated with the application specification data base). This interface specification data base governs the details of the way the user is able to issue commands to the application. It would govern, for instance, whether commands were selected from menus, from an array of icons, through a command language line, etc., or whether a particular parameter to a specific command would be selected from a menu, from a row of "radio buttons", or typed into a field on a form, etc.. The UIMS provides a basic set of facilities to perform these various kinds of interaction, and the interface developer chooses the desired kind of interaction out of this cookbook by an appropriate interface specification. This arrangement has several advantages:

- Consistency: Since interfaces for different applications use the same basic set of UIMS-provided facilities, the interfaces will be consistent at the level of interaction details (how menus work, how icons are selected, etc.). Careful design of the UIMS interface specification formalism can also lead to consistency at a higher level. Consistency of this kind is very important in the space station, particularly for those less mission-critical interfaces where not all users may be fully expert. The transfer effects made possible through consistent interface behavior will greatly facilitate interaction with unfamiliar interfaces. Moreover, consistency avoids the negative transfer effects that can impair operation of even familiar interfaces.

- Ease of interface development: Specifying an interface through the interface specification formalism of a UIMS should be significantly less effort than programming one from scratch. The UIMS formalism should provide high-level abstractions that allow the interface developer to specify the interface in terms that relate to the functionality of the interface as perceived by the user, rather than having to program it in a conventional manner at a level of detail more closely related to the implementation of the interface. This remains true even if the conventional implementation uses a high-level subroutine package of interface operations - using a subroutine package still places the emphasis on implementation, rather than abstract interface operations.

- Easier convergence on good interfaces: Despite all the advances in human-computer interaction that have occurred and continue to occur, the only known way to produce an excellent interface that fully meets the needs of its users is to build (or realistically simulate) the interface, let users interact with it, and modify it to resolve the problems that are observed. It is generally necessary to go around this loop many times before the interface performs satisfactorily, so anything that makes the loop easier and cheaper to follow is likely to improve the quality of the resulting interface by allowing more iterations. The UIMS model can speed up the modification part of the loop since interface modification can be done through modification of the declarative interface specification, rather than reprogramming in a conventional sense. This leads to a speed up in the loop as a whole.

- Ease of involvement of human factors experts: Since the UIMS model does not require programming to specify interface behavior, the interface specification can be done directly by people who are specialists in human-computer interaction, rather than by programmers. This allows better division of labor during interface/application development. Also, since programmers often think in terms of implementation ease and efficiency, rather than thinking about the interface from the user's point of view, better initial interfaces are likely to result if they are produced mainly by human factors specialists.

Of this set of advantages, only the first, consistency, and that at a relatively low level, is shared by the alternative approach of using a set of standardized interface subroutines. The other advantages all rely on a level of separation between interface and application that the subroutine approach does not provide.

Given this significant set of advantages for the UIMS approach, the natural question is why are all interfaces not produced through UIMSs.

The answer is that current UIMS systems approach the ideal described above only imperfectly. There are several specific problems.

The primary problem is that the constraints imposed by the need for an interface specification make it hard to provide ways of specifying interfaces that are carefully tailored to the needs of an individual application. Solutions to this problem (Szekeley, 1987) have tended to introduce a procedural component into the interface specification formalism. The ability to program interaction allows the interface builder to tailor interface behavior to individual interface needs. The problem with this solution is that it tends to negate the benefits of the UIMS approach, such as consistency and ease of interface modification, that depend on the interface being specified declaratively. The way around this difficulty may be to include a procedural component in the interface specification formalism, but organize it at as high a level of abstraction as possible from the interface point of view. The procedural component could then be seen as a highly specialized programming language for interface specification. Such a language could conceivably maintain consistency by encouraging through its available constructs a particular style of interaction. Ease of use for rapid interface development and use by human-computer interaction specialists would be promoted by the high-level of the abstractions involved. A great deal more research would be needed to bring this idea to fruition, but the potential payoff could be great.

A second problem with current UIMS work is that the model of communication between application and interface is too limited. Many UIMS models allow only a subset of the list of message types listed above as flowing over the UIMS/application link. And even that list is insufficient for a sizable portion of applications, especially those involving graphical or analogue manipulation, which need a much closer coupling with their interfaces than that list of communication events allows. Again, the solutions that have been explored (Szekeley, 1987; Myers and Buxton, 1986) tend to change the model in the direction of tailoring the UIMS/application link to the needs of particular applications through use of a specialized programming language - a move away from the cleanest form of the UIMS model. A compromise here may be to develop several general UIMS/application communication protocols for large classes of applications with similar needs, while still leaving open the possibility of specialized communication protocols for particular applications.

A final problem with current UIMS work concerns the potential discussed earlier for interfaces employing multiple interaction modalities in effective coordination. The coordination of the different modalities increases the challenge for the UIMS model, and the use of a UIMS approach with multiple modalities has not been explored.

Work is needed to overcome all these problems if the UIMS approach is to be practical for the space station. Unfortunately, if the UIMS approach is to be used at all, a UIMS/application communication model must be adapted before the underlying applications are developed. Since meeting the needs of complex applications through a UIMS model is still a research problem with no clear solution, the only practical way a UIMS approach can be adopted for the space station IOC is to choose that (probably quite large) subset of simpler space station applications that can be adequately serviced by currently well-developed UIMS/application communication protocols. Research in extending the limits of applicability of these protocols could nevertheless be useful for new systems developed after IOC. If these practical difficulties of adopting a UIMS approach appear too formidable for IOC, the fall-back position would be disciplined use of a comprehensive package of interface subroutines. This fall-back approach would provide the major advantage of a significant level of consistency across applications.

Interface Development Environments for Rapid Prototyping

A topic highly related to the UIMS approach to interfaces is that of interface development environments. Since the only known way to generate excellent interfaces is through an iterative process of creation, testing with users, and modification, a rapid prototyping facility for interfaces can materially improve the quality of interfaces produced by making it easier and faster to go around this loop. The rapid prototyping facilities most useful from this point of view allow interfaces to be seen and interacted with as they are developed, rather than forcing the interface developer to create the interface through working in a programming language or other formalism distinct from the interface itself. Examples of this approach include (Gould and Finzer, 1984; Myers and Buxton, 1986). They can be thought of as interface editors analogous to a what-you-see-is-what-you-get (wysiwyg) text editors. Such interface editors are a relatively new arrival on the human-computer interaction scene; their utility means they deserve a great deal more research attention.

Although rapid prototyping facilities can exist independently of the UIMS approach to interface design, they fit well with it. The cleanness of the based separation between application and interface in the VIMS model makes an interface development environment particularly useful in conjunction with a VIMS approach. A VIMS interface can be developed before the real application is available (or without incurring the expense of running the real application) by creating a dummy application that operates according to the same UIMS/application protocol as the real application. Coupled with a rapid prototyping facility, this capability allows rapid development of interface mock-ups to provide cheap and fast initial "sanity checks" on interfaces as they are developed.

Another intriguing possibility with wysiwyg interface development environments is their use (probably in restricted mode) by end users to reconfigure interfaces to their personal needs or preferences. So long as the interface modification facilities are made as easy to operate as the interfaces themselves, and so long as they do not interfere with the normal operation of the interfaces, this kind of facility could serve to improve significantly the level of personal satisfaction that space station users find with their interfaces.

Work in the area of wysiwyg interface development facilities has been almost entirely concentrated on graphical direct manipulation interfaces. This is natural in that it is the visual aspect of the interfaces that is most natural to specify in this manner. However, additional work is needed both to develop techniques for this kind of interface further, and to extend the natural interface specification techniques to multi-mode interfaces as well.

CONCLUSIONS

This paper has focussed on change in space station interfaces - the reasons that it must be expected and ways to plan for it. We have identified several topic areas associated with these two aspects of change in space station interfaces in which further research effort would be beneficial. We conclude by listing several broad areas in which we particularly recommend the support of further work.

- investigation of speech recognition techniques and natural language processing techniques for use with spoken input, plus the integration of both of these modalities with direct manipulation interfaces;

- exploration of innovative I/O devices suitable for the space station environment;

- work on the user and task modelling needed to support conversational interfaces and the integration of such interfaces with machine-like direct manipulation interfaces;

- continued development of the UIMS concept, coupled with highly interactive interface development environments for all interface modalities.

NOTES

1. The complementary concept of scarring (designing hardware for future extension and modification) is also well established, but is not addressed in this paper.

2. Though see Mark (1981), Carbonell, et al., (1983), and Douglass and Hegner (1982), for examples of successful experimental agent systems.

REFERENCES

Bolt, R. A.

1980 Put-that-there: voice and gesture at the graphics interface. Computer Graphics 14(3):262-270.

Buxton, W.

1986 There's more to interaction than meets the eye: some issues in manual input. Pp. 319-337 in User Centered System Design. D. A. Norman and S. W. Draper, eds., Erlbaum, New Jersey:

Carbonell, J. G., Boggs, W. M., Mauldin, M. L., and Anick, P. G.

1983 The XCALIBUR project: a natural language interface to expert systems. Proceedings of the Eighth International Joint Conference on Artificial Intelligence. Karlsruhe, August:

Card, S. K., Moran, T. P., and Newell, A.

1983 The Psychology of Human-Computer Interaction. Hillsdale, N.J.: Erlbaum.

Douglass, R. J. and Hegner S. J.

1982 An Expert Consultant for the UNIX Operating System: Bridging the Gap Between the User an Command Language Semantics. Los Alamos National Laboratory.

Gould, L. and Finzer, W.

1984 Programming by rehearsal. Byte 9(6):187-210.

Hansen, W. J.

1971 User engineering principles for interactive systems. Pp. 523-532 in Proceedings of the AFIPS, Fall Joint Computer Conference.

Hayes, P. J.

1986 Steps towards Integrating Natural Language and Graphical Interaction for Knowledge-based Systems. Proceedings of the Seventh European Conference on Artificial Intelligence, Brighton, July, Pp. 456-465.

1987a Using a knowledge base to drive an expert system interface with a natural language component. In J. Hendler, ed., Expert Systems: The User Interface. New Jersey: Ablex.

1987b Intelligent interfaces to expert systems. In T. Bernold, ed., User Interfaces, Gateway or Bottleneck? North Holland.

Hayes, P. J. and Reddy, D. R.

1983 Steps toward graceful interaction in spoken and written man-machine communication. International Journal of Man-Machine Studies 19(3):211-284.

Hayes, P. J. and Szekely, P. A.

1983 Graceful interaction through the COUSIN command interface. International Journal of Man-Machine Studies 19(3):285-305.

Hayes, P. J., Szekely, P. A., and Lerner, R. A.

1985 Design Alternatives for User Interface Management Systems Based on Experience with Cousin. Proceedings of CHI85, San Francisco, April.

Huff, K. E. and Lesser, V. R.

1982 Knowledge-based command understanding: an example for the software development environment. Computer and Information Sciences. University of Amherst, Massachusetts.

Hutchins, E. L., Hollan, J. D., and Norman, D. A.

1986 Direct manipulation interfaces. Pp. 87-124 in D. A. Norman and S. W. Draper, eds., User Centered System Design. New Jersey: Erlbaum.

Jacob, R. J. K.

1984 An executable specification technique for describing human-computer interaction. In H. R. Hartson, ed., Advances in Human-Computer Interaction. New Jersey: Ablex.

Loftus, J. P.

1986 Space: Exploration-Exploitation and the Role of Man. Johnson Space Center: NASA

Mark, W.

1981 Representation and inference in the consul system. Pp. 375-381 in Proceedings of the Seventh International Joint Conference on Artificial Intelligence. August, Vancouver.

Martin, J.

1973 Design of Man-Computer Dialogues. New Jersey: Prentice-Hall.

Myers, B. A. and Buxton, W.

1986 Creating highly interactive and graphical user interfaces by demonstration. Pp. 249-258 in Computer Graphics: SIGGRAPH '86 Conference Proceedings, August, Dallas, Texas.

Negronponte, N.

1981 Media room. Proceedings of the Society for Information Display 22(2):109-113.

Sacks, H., Schegloff, E. A., and Jefferson, G.

1974 A simplest semantics for the organization of turn-taking for conversation. Language 50(4):696-735.

Schegloff, E. A., Jefferson, G., and Sacks, H.

1977 The preference for self-correction in the organization of repair in conversation. Language 53(2):361-382.

Shneiderman, B.

1981 Direct manipulation: A step beyond programming languages. Computer 16(8):57-69.

Smith, D. C., Irby, C., Kimball, R., Verplank, W., and Harslem, E.

1982 Designing the star user interface. Byte 7(4):242-282.

Szekeley, P. A.

1987 Separating User Interface and Application. Ph.D. Th., Carnegie-Mellon University Computer Science Department.

Tanner, P. and Buxton W.

1983 Some Issues in Future User Interface Management System (UIMS) Development. IFIP Working Group 5.2 Workshop on User Interface Management, Seeheim, West Germany, November.

Wasserman, A. I. and Shewmake, D. T.

1984 The role of prototypes in the user software Engineering (USE) methodology. In H. R. Hartson, ed., Advances in Human-Computer Interaction. New Jersey: Ablex

Watt, W. C.

1968 Habitability. American Documentation 19:(3):338-351.

Williams, G.

1984 The Apple Macintosh computer. Byte 9(2):30-54.

Yunten, T. and Hartson, H. R.

1984 Supervisory methodology and notation (SUPERMAN) for human-computer system development. In H. R. Hartson, ed., Advances in Human-Computer Interaction. New Jersey: Ablex.

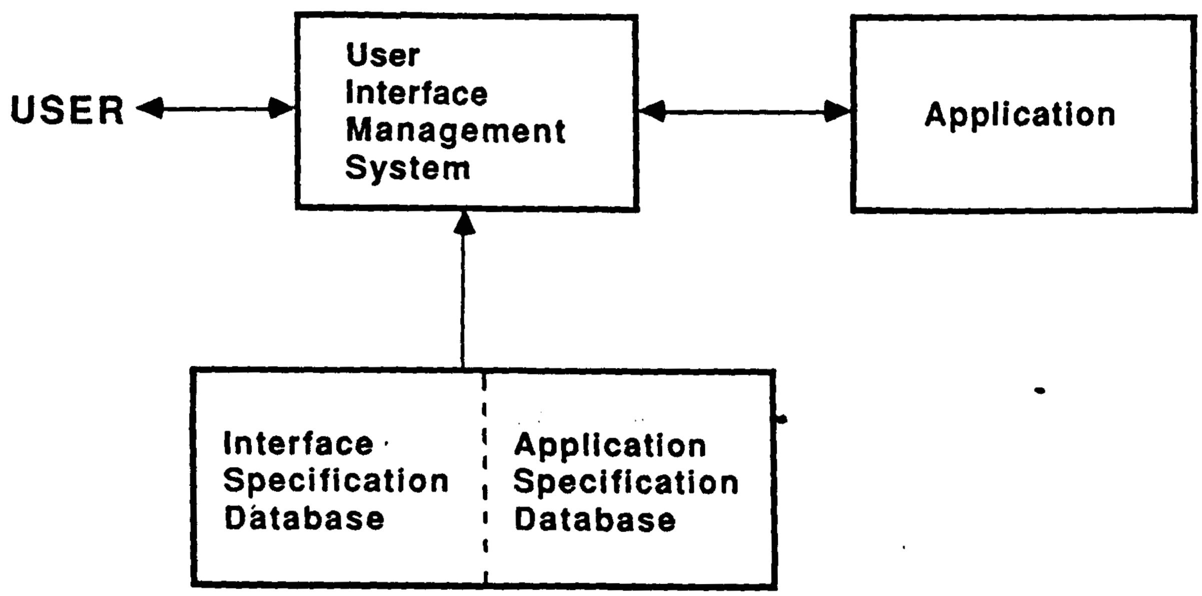

Figure 1: Model of communication in a UIMS

DRAFT
NOT FOR DISTRIBUTION, ATTRIBUTION OR QUOTATION

COGNITIVE FACTORS IN THE DESIGN AND DEVELOPMENT OF SOFTWARE IN THE SPACE STATION

Peter G. Polson

Institute of Cognitive Science,
University of Colorado
Boulder, Colorado

CONTENTS

[p. 331 DOES NOT EXIST]

Achievement of the operational and productivity goals for the Space Station will require extensive use of a wide variety of computer-based systems ranging from application programs that run on general purpose work stations to specialized embedded computer systems that monitor, operate, and trouble shoot critical subsystems, e.g., environmental and power control systems (Anderson and Chambers, 1985; Johnson et al., 1985). However, improperly designed user interfaces for these systems will compromise these goals.

The objectives of this chapter are to characterize major problems involved in the design of human-computer interfaces for systems on the Space Station and show how systematic application of empirical and theoretical results and methodologies from cognitive psychology and cognitive science can lead to the development of interfaces that reduce training cost and enhance space station crew productivity. This chapter focuses on four issues: 1) transfer of user skills, 2) comprehension of complex visual displays 3) human-computer problem solving, 4) management of the development of usable systems.

PROBLEMS

Transfer of User Skills

Inconsistent user interfaces in which the same basic function is performed by several methods in different contexts reduces transfer and interferes with retention (Polson, 1987; Postman, 1971). The Space Station's numerous computer-based systems and applications programs will be developed by different organizations over a period of many years. Inconsistency will be the rule rather than the exception unless extraordinary measures are taken in the design of user-interfaces for these systems. Popular and powerful applications programs developed for personal computers could be realistic models for software developed for the Space Station.

The typical popular applications program for a personal computer has been developed by an independent organization; the program has a great deal of functionality which is the reason for its commercial success. The user interface is unique to the application being embedded in the application's code. Effective use of the application requires specialized training and several weeks of experience. There is no consistency across different popular applications. For example, they can have very different methods for editing operations on a text string. Thus, editing an axis label on a graph, editing an operating system command, or modifying a line of text with an editor all require different sequences of user actions.

The Comprehension of Complex Visual Displays

Complex visual displays using graphics, color, and possibly motion will be used in the space station to present various kinds of information to crew members carrying out complex tasks. Poorly formatted, poorly organized, and difficult to comprehend displays will have negative impacts on the productivity. Such displays increase training costs, difficulty of complex tasks, and probability of serious operator errors.

There extensive knowledge of the processes involved in the perception of basic visual properties like color and form (Graham, 1965; Walraven, 1985), and there are numerous guidelines for display layouts and use of symbols and color (e.g. Smith and Moser, 1984; Kosslyn, 1985). However, there is no systematic knowledge of how people comprehend complex displays or use the information presented in such displays to perform complex tasks. There are no general principles for the development of effective complex displays.

Human-Computer Problem Solving

NASA has extremely ambitious plans for the use of artificial intelligence and robotics in the space station. The proposed application areas include information management, life support systems operations and monitoring, electrical power systems operations and monitoring, and guidance and navigation. Many of these tasks on the Space Station will be performed by systems with significant embedded intelligence in order to satisfy mission, technological, and economic constraints and to achieve productivity goals (Anderson and Chambers, 1985).

The use of artificial intelligence techniques can significantly increase the complexity of a system from the point of view of its human user. The crew member must now understand both the task performed by the system as well as the characteristics of the "intelligent" control program (Hayes, 1987). Waterman (1986) notes that expert systems are "brittle" when pushed beyond the very narrow domain of their real expertise can fail with little or no warning. Uncritical use of the current state-of-the-art in expert system's technology could decrease productivity of the crew and endanger their safety. Achievement of NASA's plans for the applications of artificial intelligence in the space station will require extensive basic research and rapid advances in the state-of-the-art.

SOLUTIONS

Four solutions are proposed for the problems outlined in the preceding sections: 1) Use of information processing models of tasks in the design process, 2) allocation of adequate resources to user-interface development, 3) use of user interface management systems, and 4) use of existing expertise in NASA.

Detailed Information-processing Models

The first, and most important, solution is that designs for applications programs, complex visual displays, and cooperative human-computer problem solving systems be based on detailed, information-processing models of cognitive processes involved in the performance of specific tasks. Information-processing models describe the knowledge, cognitive operations, and user actions required to perform a task. These models can also be used to generate predictions of usability parameters, e.g. training time, productivity, and mental work load, and they can be used to isolate design flaws in proposed versions of a computer-based system.

Information-processing models describe what transfers, the knowledge necessary to perform the task, and thus they can be used in the design of consistent user interfaces that facilitate transfer of user skills. Information-processing models can make important contributions to the development of effective complex visual displays. The models describe both the knowledge necessary to successfully complete a task, what is to be displayed, and the processes involved in extracting that knowledge from displays, how it is to be displayed.

Information-processing models are an important component in the successful development of effective human-computer problem solving systems. There is general agreement that successful human-computer problem solving systems will incorporate models of the task and the user (Hayes, 1987). Current theoretical methodologies in cognitive psychology and cognitive science can be used to develop both kinds of models.

Management of the Design Process

The second solution involves successful management of the development process for computer-based systems. The typical development process for complex computer-based systems in the military, NASA, and the civilian sector does not allocate enough resources to usability considerations. The primary focus of the process is on developing a system with specified functionality. Functionality is necessary but not sufficient for usability. Usability, training time and productivity, is typically evaluated late in the design cycle when it is far too late to make changes that improve usability. The design of highly productive complex computer-based systems requires solving simultaneously two interrelated sets of design problems involving functionality and usability.

What is proposed in this chapter is that usability and functionality considerations receive equal weight during all phases of the design cycle. The preliminary version of the system is evaluated for usability. If the system fails to meet usability goals, the design is revised. The revised design is then evaluated. This iterative process continues until the design meets both usability and functionality goals (Gould and Lewis, 1985; Hayes, 1987).

User Interface Management Systems

The third solution involves the use of appropriate technologies. Many of the problems involving transfer of user skills and consistency across applications can be solved using user interface management systems. The nature of these systems is discussed in Hayes (1987) and Hayes, Szekely, and Lerner (1985). They will not be discussed further here.

Existing Expertise in NASA

The fourth solution involves making effective use of the expertise already within NASA. What is being proposed here is similar to other modeling efforts currently underway in NASA dealing with problems of anthropometrics and habitability. OPSIM (Globus and Jacoby, 1986) is a computer model that simulates crew actions and interactions carrying out specific tasks under constraints imposed by different interior configurations, crew size and skills and other environmental factors. These simulated task scenarios are used to rapidly explore a large number of variables involving the environment and crew composition iteratively developing a more optimal design. Detailed models of the cognitive operations and physical actions required to carry out various types of tasks involving interaction between man and machine can be used in a similar fashion to optimize designs for user interfaces.

Alternative Solutions

Guidelines and Handbooks

Human factors guidelines (Smith and Mosier, 1986) and handbooks summarize information ranging from design goals and methodology to specific data on perceptual and motor processes. Guidelines and handbooks contain parametric information about basic perceptual and motor processes and information on limitations of classes of interaction techniques. However, they are of limited use in characterizing higher-level cognitive processes, e.g. comprehension, learning, and problem solving. Guidelines propose reasonable design goals for cognitive aspects of a system, but they contain little or no advice on how to achieve such goals. Examples of cognitive guidelines include "minimize working memory load" and "minimize the amount of information the user has to memorize".

Usability parameters characterize the use of a system to perform a task, e.g. training time, productivity, and user satisfaction. Developing a system that optimizes usability parameters requires understanding of the task and the cognitive processes involved in performing the task. Most features incorporated into user interfaces are not good or bad per se. Usability is determined by interactions of the specific features of a design with the structure of a task. Guidelines do not contain necessary information about task structure, the knowledge required to perform a task, or the dynamics of the cognitive processing required to perform the task. Our knowledge of cognitive processes is in the form of detailed information processing models of the performance of complex tasks.

Many writers (e.g. Gould and Lewis, 1985; Hayes, 1987) argue that successful interface design is an iterative process. This view is strongly championed in this chapter. It is not possible to derive an optimal interface from first principles. Accumulated experience, information in guidelines and handbooks, and careful theoretical analyses can lead to the development of a reasonable initial trial design. However, this design has to be evaluated, modified, and evaluated again. In other words, guidelines and handbooks are not enough.

Empirically Based Modeling Strategies

Gould and Lewis (1985) and Carroll and Campbell (in press) seriously question the theoretically driven design and evaluation processes championed in this chapter. They argue that there are serious limitations of current modeling techniques, e.g. the limitations on our knowledge of comprehension of complex visual displays. They champion empirically-based modelling and evaluations methodologies. Many successful, complex systems, e.g. today's generation of highly automated aircraft, evolved from a combination of increasing technical capabilities, e.g. highly reliable microprocessors, and extensive operational experience (Chambers and Nagel, 1985).

However, relying on empirical methods to evaluate trial designs has serious limitations. They include difficulties in extrapolating results, doing experiments to evaluate complex systems, and evaluating transfer of training. For example, in a very complicated system, it may not be feasible to do empirical studies to evaluate a large number of tasks or to evaluate transfer between many tasks. If the current version of a trial design has unacceptable usability parameters, a designer has the very difficult task of deciding what attributes of the current design should be changed in order to improve performance. A theoretical model provides an explicit decomposition of the complex underlying processes. This additional detail describing the underlying processes can be very valuable in making well motivated changes leading to the next iteration of the design process.

OUTLINE OF REMAINDER OF CHAPTER

The remainder of this chapter is organized into five sections. The first provides a general characterization of the kinds of theoretical models of cognitive processes that we argue should be the basis for the design of highly usable computing systems. The next section describes a detailed analysis of the process involved in the transfer of user skills and presents summaries of empirical results supporting these theoretical analyses. This section also provides a description of current theoretical models of human-computer interaction. Transfer is a well understood problem. The objective of this long section is to provide an illustration of a successful solution. The next section describes some of the difficult problems involved in the design of effective complex visual displays. The fourth section discusses the problems involved in the development of effective cooperative man-machine systems. The final section makes recommendations for further research.

MODELS OF COGNITIVE PROCESSES

The information processing framework (Newell and Simon, 1972; Gardner, 1985) provides the basis for the development of detailed process models of tasks performed on the Space Station. These theoretical analyses can be used as the basis for the design of human-computer interfaces that have minimal training costs and for the task and user models incorporated into human-computer problem solving systems.

The Information Processing Framework

An information processing model incorporates representations of the task, the knowledge required to perform the task, and the processes that operate on the representation to perform the task (Gardner, 1985). Such models are often formalized as computer simulation programs. The framework characterizes the general architecture of the human information processing system which in turn constrains the nature of the representations and the processes that operate on them, e.g., limited immediate memory. Newell and Simon (1972) and Anderson (1976, 1983) have proposed that the human information processing system can be described as a production system. The following section describes production system models of human-computer interaction.

Models of Human Computer Interaction

The GOMS model (Card et al., 1983) and Cognitive Complexity Theory (CCT) (Kieras and Polson, 1985) both characterize the knowledge necessary to make effective, routine use of software tools like an operating system, a text editor, or a data-base manager. The GOMS formalism describes the content and structure of the knowledge underlying these skills. CCT represents this knowledge as production rules which permits one to quantify amount. CCT incorporates all of the assumptions of the GOMS model. The production rule formalism enables one to derive quantitative predictions of training time, transfer of user skills, and performance. The next two sections describe each framework.

The GOMS Model

The GOMS model represents a user's knowledge of how to carry out routine skills in terms of goals, operations, methods, and selection rules.

Goals represent a user's intention to perform a task, a subtask, or a single cognitive or a physical operation. Goals are organized into structures of interrelated goals that sequence cognitive operations and user actions.

Operations characterize elementary physical actions (e.g., pressing a function key or typing a string of characters), and cognitive operations not analyzed by the theory (e.g., perceptual operations, retrieving an item from memory, or reading a parameter and storing it in working memory).

A user's knowledge is organized into methods which are subroutines. Methods generate sequences of operations that accomplish specific goals or subgoals. The goal structure of a method characterizes its internal organization and control structure.

Selection rules specify the conditions under which it is appropriate to execute a method to effectively accomplish a specific goal in a given context. They are compiled pieces of problem solving knowledge. They function by asserting the goal to execute a given method in the appropriate context.

Content and Structure of a User's Knowledge

The GOMS model assumes that execution of a task involves decomposition of the task into a series of subtasks. A skilled user has effective methods for each type of subtask. Accomplishing a task involves executing the series of specialized methods that perform each subtask. There are several kinds of methods. High-level methods decompose the initial task into a sequence of subtasks. Intermediate-level methods describe the sequence of functions necessary to complete a subtask. Low-level methods generate the actual sequence of user actions necessary to perform a function.

A user's knowledge is a mixture of task-specific information, the high-level methods, and system-specific knowledge, the low-level methods. The knowledge captured in the GOMS representation describes both general knowledge of how the task is to be decomposed as well as specific information on how to execute functions required to complete the task on a given system.

Cognitive Complexity Theory

Kieras and Polson (1985) propose that the knowledge represented in a GOMS model be formalized as a production system. Selection of production systems as a vehicle for formalizing this knowledge was theoretically motivated. Newell and Simon (1972) argue that the architecture of the human information processing system can be characterized as a production system. Since then, production system models have been developed for various cognitive processes (problem solving: Simon, 1975; Karat, 1983; text comprehension, Kieras, 1982; cognitive skills: Anderson, 1982).

An Overview of Production System Models

A production system represents the knowledge necessary to perform a task as a collection of rules. A rule is a condition-action pair of the form

IF (condition) THEN (action)

where the condition and action are both complex. The condition represents a pattern of information in working memory that specifies when a physical action or cognitive operation represented in the action should be executed. The condition includes a description of an explicit pattern of goals and subgoals, the state of the environment, (e.g., prompts and other information on a CRT display), and other needed information in working memory.

Production Rules and the GOMS Model

A production system model is derived by first performing a GOMS analyses and then writing a program implementing the methods and control structures described in the GOMS model. Although GOMS models are better structural and qualitative description of the knowledge necessary to perform tasks, expressing the knowledge and processes in the production system formalism permits the derivation of well motivated, quantitative predictions for training time, transfer, and execution time for various tasks.

Kieras and Bovair (1986), Polson and Kieras (1985) and Polson et al. (1986) among others have successfully tested assumptions underlying these predictions. These authors have shown that the amount of time required to learn a task is a linear function of the number of new rules that must be acquired in order to successfully execute the task and that execution time is the sum of the execution times for the rules that fire in order to complete the task. They have shown that transfer of training can be characterized in the terms of shared rules.

TRANSFER OF USER SKILLS

In a following section, research on transfer of user skills in human-computer interaction will be reviewed. This research shows that it is possible to give a very precise theoretical characterization to large transfer effects, reductions in training time on the order of three or four to one. These results strongly support the hypothesis that large transfer effects are due to explicit relationships between different tasks performed on the same system or related tasks performed on different systems. Existing models of the acquisition and transfer of cognitive skills enable us to provide precise theoretical descriptions of these transfer processes. These same models can in turn be used to design consistent user interfaces for a wide range of tasks and systems that will promote similar large reductions in training time and saving in training costs.

A Theoretical Model of Positive Transfer

The dominant theoretical approach for explaining specific transfer effects is due to Thorndike and Woodward (1901) and Thorndike (1914). Thorndike assumed that transfer between two tasks is mediated by common elements. Common elements acquired in a first task that successfully generalize to a second do not have to be relearned during the acquisition of the second task. If a large number amount of the knowledge required to successfully perform the second task transferred, there can be a dramatic reduction in training time.

Kieras and Bovair (1986) and Polson and Kieras (1985) proposed that a common elements theory of transfer could account for positive transfer effects during the acquisition of operating procedures. The common elements are the rules. Tasks can share methods and sequences of user actions and cognitive operations. These shared components are represented by common rules. It is assumed that these shared rules are always incorporated into the representation of a new task at little or no cost in training time. Thus, for a new task in the middle of a training sequence, the number of new unique rules may be a small fraction of the total set of rules necessary to execute this task.

Examples of Successful Transfer

This section briefly describes results from the human-computer interaction literature demonstrating the magnitudes of the transfer effects and showing how CCT (Kieras and Polson, 1985) can explain these results.

Polson et al. (1986) found very large transfer effects, on the order of four to one reductions in training time, for learning to perform a simple utility task on a menu-based, stand-alone, word processor. Their theoretical analysis showed that a significant portion of the knowledge, when quantified in terms of number of rules, required to perform these tasks were in consistent with low-level methods for making menu transitions, entering parameters, and the like.

Singley and Anderson (1985) found large transfer effects between different text editors, e.g., transfer from a line to a screen editor. Polson, Bovair, and Kieras (1987) found effects of similar magnitude for transfer between two different screen editors. Their theoretical analysis showed that editors share common top level methods that decompose the task of editing a manuscript into a series of subtasks involving individual changes in the manuscript. Furthermore, even very different editors share low-level methods, e.g., cursor positioning. Text editing is a task where transfer is mediated by knowledge of the general structure of the task as well as shared methods.

The Xerox STAR is a workstation that was explicitly designed to maximize the transfer of methods both within a given application as well as across different applications (Smith et al. 1983). All commands have a common format. The user first selects an object to be manipulated using specialized selection methods for different kinds of text or graphic objects. The operation is selected by pressing one of four command keys on the keyboard. For example, hitting the delete key causes the selected object to be deleted.

Ziegler et al. (1986) carried out transfer experiments with the STAR workstation. They studied transfer between text and graphics editors. They showed that common methods acquired in one context were successfully transferred to the other leading to very large transfer effects. Further, they were able to provide a quantitative analysis of the magnitude of these transfer effects using a production system model like those of Polson et al. (1987).

An Example of the Impact of Low Level Inconsistencies

Karat et al. (1986) examined transfer between three highly similar word processing systems that were intended by their designers to facilitate the transfer of user skills from one system to another. The first system was developed as a menu-based, stand alone word processor. A major goal in the design of the follow-on systems was to facilitate transfer from the dedicated, stand-alone, word processor to word processors hosted on a general purpose personal computer and a departmental computing system.

Karat et al. evaluated the magnitude of transfer effects from the dedicated version of the system to the other two system environments. The transfer effects were disappointingly small. Karat et al. found users' difficulties transferring their skill were due almost entirely to subtle differences in low level-methods. For example, many problems were caused by the fact that the dedicated version of the system has specialized, labeled function keys. On the general purpose personal computer and the departmental computer system versions, the user had to learn and retain the locations of the corresponding functions on an unlabeled, generic keyboard. Inconsistencies in key assignments for activating known functions disrupted performance when users attempted to transfer their skills from one version of the system to another.

Implications for the Design of Systems in the Space Station

The research reviewed in preceding sections shows that common methods are transferred across tasks and application leading to large reductions in training time, on the order of 100% to 300%. However, the Karat et. al. results show that these transfer effects are fragile and can be reduced by minor but arbitrary differences in low-level methods let alone more extensive inconsistencies. For example, the method for centering text is identical on both the dedicated and personal computer versions of the systems <u>except that</u> the centering function is activated on the dedicated version by Control-Shift C and by Control-Shift X on the personal computer version. This small inconsistency disrupted the performance of skilled users of the dedicated version forcing them to stop and refer to documentation to find the correct function key. This inconsistency was caused by the fact that Control-Shift C already used by many applications programs to abort and return to the top level of operating system.

The potential for serious inconsistencies in common methods across different systems and application in the Space Station is much greater than the example of the three word processing system studied by Karat et. al. They were all developed by a single manufacturer with the explicit goal of permitting transfer of skills developed on the dedicated version of the system.

COMPREHENSION OF COMPLEX VISUAL DISPLAYS

Rapid developments in hardware and software technology permit the generation and presentation of very complex displays combining text, color, motion, and complex visual representations. There is limited understanding of how to effectively utilize these new capabilities. There is extensive knowledge of the basic visual processes underlying color and form perception (Graham, 1965; Walraven, 1985). Detailed models of the comprehension of complex visual displays do not exist. There is some systematic work on the effective graphical presentation of quantitative information (e.g., Kosslyn, 1985; Tufte; 1983). The widely acclaimed book The Visual Display of Quantitative Information by Tufte is a collection of design guidelines.

Today, development of effective complex displays relies almost entirely on empirically-based, iterative design methods (Gould and Lewis, 1985). A good illustration of how effective these methods can be is shown in an experiment reported by Burns et al. (1986). These investigators were concerned with the problem of display format optimization. They designed a set of alternative displays to be used in orbital maneuvering tasks onboard the Space Shuttle. The new displays grouped information by function and include more meaningful abbreviations and labels. Burns et al. (1986) had both non-experts and Space Shuttle crew members retrieve specified items of information from the current, operational displays and the reformatted experimental displays.

Reformatted displays improved both speed and accuracy for the non-expert subjects. The changes in format had no effects on Space Shuttle crew member performance, and the reformatted displays improved their accuracy. These results are surprising. Extensive training and experience should have enabled the crew members to develop specialized skills to deal with even non-optimal displays. Any changes in display format should have disrupted these skills leading to reductions in performance for highly trained crew members. One possible conclusion is that the current displays are so far from optimal that even brief experience with the reformatted displays enabled trained crew members to perform at a level equal to their performance with actual displays.

The Burns et al. (1986) experiment shows that application of our current knowledge of visual perception and guidelines for formatting and labeling can lead to significant improvements of performance in an empirically-based iterative design process. However, the situation in the Space Station is more complex. The display technology for the systems onboard the Space Shuttle used small, alpha-numeric CRTs. Displays onboard the Space Station will make extensive use of graphics and color. In other words, increase capabilities provided by new display technology will enable developers to generate truly incomprehensible displays.

Furthermore, there are important transfer and consistency issues. Conflicting uses of symbols, color and motion cues, and inconsistent formats across applications will have the same impact on users as inconsistent methods for entering text, increased training time and probabilities of user errors. Dealing with issues involving more complex displays, consistency, and the use of displays as interfaces to systems with significant embedded intelligence are more complex design problems. The design problems will have to be solved using the combination of empirically-based evaluation methods combined with detailed models of the task and a theory of the comprehension of visual displays.

Consider the design problems involved in developing the displays for systems with significant embedded intelligence like the Space Station's environmental controls and power systems. Effective displays should be based on 1) an understanding of the knowledge required to successfully perform critical tasks, e.g., trouble shoot a malfunction, 2) a characterization of the cognitive processes involved in extracting the necessary information from the display, 3) and a description of how the information is utilized to complete the task. In other words, what is required is a complete theory of the comprehension of complex visual displays.

Ellis and his colleagues (Ellis et al., 1985; Kim, Won Soo et al., 1985) have proposed a methodology for the development of effective specialized displays for spatial tasks involving control of objects in three dimensional space with a full six degrees of freedom, e.g. the JPL Telerobot demonstrator, and Space Station Proximity Operations Displays. Ellis and his colleagues propose a design methodology that creates a very tight link between the characteristics of the task, a theoretical understanding of the perceptual processes, and empirical demonstrations that the displays actually facilitate performance of the task. This design strategy can be generalized in all various types of displays and tasks.

HUMAN-COMPUTER PROBLEM SOLVING

NASA has articulated a very ambitious design philosophy for expert systems to be used on the Space Station calling for the development of cooperative human-computer problem solving systems. Many issues concerning the design of such systems can be understood from experience with highly automated commercial aircraft (Chambers and Nagel, 1985), automatic test equipment (Richardson et. al., 1985), and automated control systems for nuclear power plants. Some of the issues are: 1) vigilance of the human operator, 2) safe transition from automatic to manual modes of operation, 3) maintenance of skills necessary to perform tasks manually, 4) successful completion of a task after the automatic system has failed, 5) allocations of functions between man and machine, 6) and the development of truly symbiotic human-computer problem solving systems. Although the basic issue have been identified, there are no well worked out general solutions nor are there any operational examples of symbiotic human-computer problem solving systems.

Autonomous Vs Cooperative Systems

Hayes (1987) distinguishes between conversational/agent and machine/tool-like systems. In a conversational/agent system, the user interacts with an intelligent agent to accomplish a task. Robots that carry out complex EVA tasks under human supervision and systems with sophisticated natural language interfaces are examples. Machine/tool-like systems are directly controlled by their users although they can be highly automated carrying out a whole sequence of low level steps without direct intervention. Examples include auto-pilots, automatic test equipment (ATE) and application programs like text editors and spreadsheets.

There also is a second important dimension, autonomy. Some systems, once initialized by their users, carry out their task completely autonomously or only make use of the human user as a low level sensor and manipulator. Examples include auto-pilots, ATE systems, and most expert systems. Auto-pilots and ATE systems are not normally considered intelligent. However, they carry out extremely complex tasks autonomously. They may not be classified as intelligent systems in that they carry out their tasks using well understood algorithms. Many expert systems imply the human user as a low-level sensor and manipulator. The task is carried out autonomously. The user can ask for explanations of the final results or why the system requested a given piece of data in the process of completing the task (e.g., Shortcliffe, 1976).

Limitations of Current Expert Systems

Intelligent systems can actually complicate the task of human user, e.g., telerobots and applications with natural language interfaces. Bejczy (1986) shows that intelligent agents can impose additional difficulties for users because they have to understand both the control program and the task. For example, no natural language interface is capable of responding correctly to unrestricted input. Such interfaces understand a limited subset of natural language and may have no or limited capabilities for reasoning about the task. Thus, even if the user's request is parsed correctly, resulting commands may be an incomplete and/or incorrect sequence of operations necessary to complete the task.

Consider the problem of effective handoff from automatic to manual operation in a troubleshooting task, e.g., finding a serious fault in the power distribution system. Current expert systems do not make the transition from automatic to manual operation gracefully. Waterman (1986) observes that expert systems have narrow domains of expertise and they have on capability to reason about their limitations. Because they can't reason about their limits, such systems are little use in assisting a human problem solver once they have failed to find the cause of a serious fault. Thus, the system can fail catastrophically leaving its user with a task of manually diagnosing a serious fault.

Building a system capable of reasoning about its limits and providing the user with a useful explanations regarding failure is beyond the current state-of-the art. However, it's exactly this kind of capability that is required in a truly cooperative system. In summary, current expert systems are not cooperative problem solving systems. In the process of performing their task, humans serve in a very low level subserviant role and when systems fail, they fail catastrophically providing their users with little or no information for the reason of the failure and no assistance in continued efforts to solve the problem.

Being able to reason about its own limitation is difficult because constrains embedded in the fundamental properties of current knowledge representation schemes (Jackson, 1986). The rules in current expert systems contain a complex mixture of control knowledge and domain specific and general problem solving knowledge. Such systems have no explicit model of domain principles or any specific knowledge of their strategies. Exactly this kind of knowledge is required to produce coherent explanations (Clancy, 1983). This type of knowledge is also required to reason about limitations.

Cooperative Human-computer Problem Solvers

NASA's goals are far more ambitious than the development of autonomous intelligent problem solvers with explanation capabilities. It is repeatedly proposed in various NASA documents to develop cooperative or symbiotic human-computer problem solvers (Johnson et al. 1985; Anderson and Chambers, 1985).

Discussions about the possibility of developing such systems have a surprising uniformity. The authors observe that powerful problem solvers can be developed if systems exploit the complimentary strengths of human and machine permitting one to compensate for the weaknesses of the other. The next issue is function allocation. The discussion of function allocation begins with a general assessment of the strengths and weaknesses of human and computers as problem solvers. This assessment is in the form of a characterizations human and machine components listing the strengths and weaknesses of each. Typical listings are in Johnson et al., 1985, pp. 27-28; Richardson et al., 1985, pp. 47-49; Anderson and Chambers, 1985. What is striking about these lists is their consistency. The following is taken from Richardson et al. (1985, pp. 47-49).

The strengths of the human component of the system are:

1. Processing of sensory data.
2. Pattern recognition.
3. Skilled physical manipulation but limited physical strength.
4. Limited metacognitive skills, e.g. ability to reason about limits of knowledge and skill.
5. Slow but powerful general learning mechanisms.
6. A large, content-addressabale permanent memory.

The weaknesses of the human problem solver are:

1. Limited working memory.
2. Limited capacity to integrate a large number of separate facts.
3. Tendency to perseverate on favorite strategies and malfunctions; set effects and functional fixity.
4. Limited induction capabilities.
5. Lack of consistency; limitations on the ability to effectively use new information.
6. Emotional and motivational problems.

7. Limitations on the availability of individuals with the necessary abilities and skills.

8. Limited endurance.

The current generation of expert systems and highly autonomous automatic systems, e.g. ATE's make use of human sensory processing, pattern-recognition, and manipulative skills. Most authors recognize this and point out that their objective in developing cooperative problem solving systems is to exploit human's cognitive capabilities as well as these lower level skills. Continuing to quote Richardson,et al., the strength of the computer component of the system are:

1. Large processing capacity.
2. Large working memory.
3. Capabilities of making consistent mechanical inferences taking into account all relevant facts.
4. Processing and utilizing large amounts of actuarial information.
5. Capabilities to store and retrieve training and reference material.
6. Availability of system is limited only by reliability of basic computer technology.
7. No motivational or other related problems.

The weaknesses of the machine component of the system are

1. No or very limited capacity to adapt to novel situations.
2. No or very limited learning abilities.

3. No or very limited metacognitive abilities, e.g., understanding of own limitations.

4. Very difficult to program particularly the current generation of expert systems.

Examples of Cooperative Systems

The best examples of cooperative systems are intelligent training systems (ITS) (Sleeman and Brown, 1983; Polson and Richardson, 1987). The main components of an ITS are: 1) the expert module or task model, 2) the student module or user model, and 3) the tutor module or explanation subsystem. A cooperative, intelligent problem solving aid has to have real expertise about the task, an accurate model of the other intelligent agent that it is interacting with (the human user), and the capability of conducting sophisticated dialogues with the user. Richardson et al. (1985) argue that the machine component should attempt to compensate for known limitations and failure modes that are characteristics of all forms of human problem solving: They are working memory failures, set and functional fixity, inference failures, and attentional limitations.

One important role for a cooperative intelligent system would be to reduce information overload by selectively displaying information relevant to the highest priority subcomponent of a task. Chambers and Nagel (1985) describe the cockpit of a Boeing 747 with its several hundred instruments, indicators, and warning lights as an example of where skilled pilots can be simply overwhelmed by the amount of available information. Plans for highly automated aircraft of the 1990's incorporate selective displays on color CRTs of a small subset of the total information about the state of the aircraft that is relevant to the current task. The ability to display relevant information would prevent information overload and augment human working memory by providing an external representation relevant information about the system's state.

Other proposals for the role of the computer in a cooperative system focus on its computational capabilities. Memory limitations prevent human users from adequately integrating information about the current state of the system and archival information concerning likelihoods of component failures. Thus, the machine takes on the role of filter, memory aid, and inference engine compensating for known general weaknesses in the human information processing system.

Possible Senarios - Serious Problems

These proposals are consistent with the large body of data about the strength and weaknesses of human diagnostic reasoning and problem solving. However, implementing these proposals into a functioning system can cause serious difficulties. Consider a situation involving the power distribution system of the Space Station where several interacting failures have occurred. The system makes a series of incorrect inferences about the cause of the faults and displays to the human partner information irrelevant to successful solution of the problem. Such misinformation could effectively block successful solution by the human user. It's essentially a set manipulation. The misinformation would be especially damaging if the system were normally successful.

Other problems could result if the system makes incorrect inferences from its model of the human user. Assume the system has concluded, correctly, that is is incapable of independently diagnosing the faults in the power distribution system. Using its advanced explanation capabilities, it explains to its human partner its understanding of the current state of the power distribution system and various partial results obtained in attempting to diagnose failures. In the process, system presents a series of complex displays showing the current state of the power distribution system. The expert human user recognizes a complex pattern of interrelated events and informs the computer of the correct solution to the problem. The system responds by attempting to evaluate the human partner's input using information contained in its user model. This model has a very detailed description of the limits of the human information processing system, and the system incorrectly concludes that the human partner is incapable of making the correct diagnosis on the basis of such complex input and the solution is rejected.

Conclusions

Many readers may think that the scenario presented in the preceding section is overdrawn. Of course, NASA would never tolerate the fielding of a system that was capable of effectively overruling a Space Station crew member. However, a system in which human users can override the machine partner compromises the goal of developing truly cooperative human-computer problem solving systems. Information overload, working memory failures, and failures to integrate historical data in making diagnoses are highly probable failure modes of human users. The incorrect inference made by the machine described in the preceding scenario is not unreasonable and would probably be correct in most situations. Experience with intelligence tutoring systems (Polson and Richardson, in press) shows that such cooperative systems are exceedingly difficult to construct.

RECOMMENDATIONS FOR FURTHER RESEARCH AND CONCLUSIONS

This section contains information on recommendations for further research and concludes that the difficulties in developing truly productive computer-based systems are primarily management problems.

Information Processing Models

Recommendation 1. Support the development of the software tools required to rapidly develop information processing models of tasks performed on the Space Station.

This chapter has recommended that information processing models of cognitive processes be the basis for the design of applications programs, complex visual displays and cooperative human-computer problem solving systems. A theoretical technology should be applied on a large scale to solve interface design problems on the Space Station. Unfortunately, the development of information processing models is currently an art and not a robust design technology. Furthermore, these models can be extremely complex simulating basic psychological process in detail (Anderson, 1983). What is required are engineering models (Newell and Card, 1986; Kieras & Polson, 1985).

Development of an effective modeling facility is an engineering problem, abeit a difficult one. There are no advances required in the theoretical state of the art in cognitive psychology. Models of various cognitive processes have to be integrated into a single simulation facility, e.g., models of perceptual, cognitive, and motor processes. Higher level languages should be developed that automate the generation of the simulation code and the detail derivation of models. A simulation development system will be required for designers to rapidly develop models of adequate precision for use in a timely fashion in the design process.

The Comprehension of Complex Displays

Recommendation 2. Support an aggressive research program on the processes involved in the comprehension of complex, symbolic displays.

Many tasks on the Space Station will require that crew members interact with complicated displays. Examples include monitoring and trouble shooting of complex subsystems, manipulation and presentation of scientific data, and interacting with expert systems to carry out trouble shooting and maintenance tasks. Rapid advances in computer and display technology will enable designers to develop complex displays making using of symbolic, color, and motion cues. Effective displays that facilitate performance on these complex tasks can have large positive effects on crew productivity. The complexity of the tasks and the freedom given to the designer by the display technology require that successful designs be based on explicit models of how information in such displays is used to perform these tasks.

Development of models of the comprehension of complex displays requires important contributions to cognitive theory. Current research in cognition and perception provides a solid foundation on which to build such models. It is possible that models of comprehension of complex displays can be based on the extensive body of theoretical results obtained on the processes involved in text comprehension (e.g., van Dijk and Kintsch, 1983). Excellent work on related problems is already going on within NASA; research programs in this area could be modeled in the work of Ellis and his colleagues briefly described in a preceding section.

Human-Computer Problem Solving

<u>Recommendation 3.</u> Design and support an aggressive research program leading to the eventual development of cooperative, human-computer problem solving systems.

Although the many analyses characterizating cooperative human-computer problem solving are correct, development of a useful cooperative system requires solutions to unsolved problems in expert system design, artificial intelligence, and cognitive science. A well structured research program would generate many intermediate results, components of the eventual cooperative system, that are useful in themselves on the Space Station. These include robust, high performance expert systems, advanced explanation subsystems, and various problem solving tools to assist the crew in management of the Space Station systems.

Consider utilities of an inspectable expert system and of an inference engine tool. By an inspectable expert system, we mean a system that displays intermediate states of its diagnostic processes during trouble shooting. The expert systems tool presents to the trained user intermediate results of the trouble shooting process using of complex, symbolic displays. Properly designed, such information gives the human expert the information necessary to confirm a diagnosis or take over effectively if the expert system fails. Most current automatic test equipment simply reports success or failure, e.g., a red light or a green light. An inspectable expert system would be a dramatic improvement over diagnostic systems with such limited feedback.

Another useful subsystem would be a inference engine, a tool that combines information about system state with actuarial data on the likelihoods of different failure modes. This system would be designed to enable a skilled human user to do what if calculations and serve as a memory aid reminding the crew member of infrequently occurring faults that are likely to be overlooked.

Inspectable expert systems are within the state-of-the-art and would serve as a very useful test bed for research on comprehension of complex symbolic displays and on the design of such displays. An interactive inference engine could be seen as a primitive prototype of a cooperative problem solving system. Both tools can be very useful in an operational environment and both are important intermediate steps in the eventual development of high performance cooperative systems.

There are important areas of research in cognitive science that will have to be better developed before it will be possible to build successful cooperative human-computer problem solving systems. These include models of human diagnostic reasoning, cooperative problem solving, and models of the processes involved in generating and comprehending useful explanations. A cooperative system must incorporate an extremely sophisticated model of its human partner which in turn requires a detailed understanding of how humans carry out the specific task performed by the system as well as the general characteristics of the human information processing system and its failure modes. User models are related to the problem of developing student models in intelligent training systems. Although progress is being made in the area of student modeling, there is still numerous important unsolved problems (Polson and Richarson, 1987).

In summary, the design and development of cooperative, human-computer problem solving is the most difficult of the technological goals related to cognitive science associated with the Space Station. This goal will only be achieved by a long term, well managed research program.

In Reality, It's a Management Problem

It is widely recognized that the ambitious productivity goals for the Space Station can only be achieved with extensive use of automated systems that have effective user interfaces. However, there is a broad gap between good intentions and actual development practice. It is widely recognized today that complex systems developed for civilian, NASA, and military use are far from the current state-of-the-art in human factors presenting serious problems for their users. Often, design errors are so obvious that applications of simple common sense could lead to the development of more usable interfaces.

In the final analysis, development of usable systems is a management problem. Consistent application of the current state-of-the-art in human factors and knowledge of cognitive processes during all phases of the development process would have dramatic and positive effects on the productivity of the Space Station crew.

REFERENCES

Anderson, J. R.

1976 Language, Memory, and Thought. Hillsdale, N.J.: Lawrence Erlbaum Associates.

1982 Acquisition of cognitive skill. Psychological Review 89:369-406.

1983 The Architecture of Cognition. Cambridge, MS: Harvard University Press.

Anderson, J. R., and Chambers, A. B.

1985 Human Centered Space Station Design. (Polson can't find his copy of this paper. It was included in one of the early mailings to the symposium panel providing background information and logistics for the originally scheduled (Feb. 12, 1986) briefing at JSC, Houston.)

Bejczy, A. K.

1986 Human factors in space teleoperation. Transactions of the 2nd International Symposium on Next Generation Transportation Vehicles. Analfi, Italy, June 20-24.

Burns, M. J., Warren, D. L., and Rudisill, M.

1986 Formatting space-related displays to optimize expert and nonexpert user performance. Pp. 274-280 in M. Mantei, and P. Orbeton, eds., Proceedings CHI'86 Human Factors in Computer Systems. New York: Association for Computing Machinery.

Card, S. K., Moran, T. P., and Newell, A.

1983 The Psychology of Human-Computer Interaction. Hillsdale, N.J.: Erlbaum.

Carroll, J. M.

1987 Interfacing Thought. Cambridge, MS: In press. Bradford Books/MIT Press.

Carroll, J. M., and Campbell, R. L.

1987 Softening up hard science: a reply to Newell and Card. Human-Computer Interaction. In press.

Chambers, A. B., and Nagel, D. C.

1985 Pilots of the future: Human or computer? Computer November:74-87.

Clancy, W. J.

1983 The epistemology of a rule-based expert system: A framework for explanation. Artificial Intelligence 20:215-251.

Ellis, S. R., Kim, W. S., Tyler, M., and McGreevy, M. W.

1985 Visual enhancements for perspective displays: perspective parameters. *Proceeding of the International Conference on Systems, Man, and Cybernetics*. IEEE Catalog No. 85CH2253-3. November, 815-818.

Engle, S. E. and Granda, R. E.

1975 *Guidelines for Man/Display Interfaces*. Technical Report TR 00.2720. Poughkeepsie, NY: IBM.

Garnder, H.

1985 *The Mind's New Science: A History of the Cognitive Revolution*. New York: Basic Books.

Globus, A. and Jacoby, R.

1986 *Space Station Operational Simulation (OPSIM)*. Moffett Field, CA: NASA-Ames Research Center.

Gould, J. D. and Lewis, C.

1985 Designing for usability: key principles and what designers think. *Communications of the ACM* 28:300=311.

Graham, C. H.

1965 *Vision and Visual Perception*. New York: John Wiley and Sons.

Hayes, P.

1987 Changes in human-computer interfaces on the space station: why it needs to happen and how to plan for it. Humans in Automated and Robotic Space Systems. National Academy of Sciences. Washington, D. C.

Hayes, P. J., Szekely, P. A., and Lerner, R. A.

1985 Design alternatives for user interface management systems based on experience with cousin. Pp. 169-175 in L. Borman, and B. Curtis, eds., Proceedings of the CHI 1985 Conference on Human Factors in Computing. New York: Association for Computing Machinery.

Jackson, P.

1986 Introduction to Expert Systems. Workingham, England: Addison-Westley Publishing Company.

Johnson, R. D., Bershader, D. and Leifer, L.

1985 Autonomy and the Human Element in Space: Final Report of the 1983 NASA/ASEE Summer Faculty Work Shop. Moffett Field: NASA-Ames Research Center.

Karat, J.

1983 A model of problem solving with incomplete constraint knowledge. Cognitive Psychology 14:538-559.

Karat, J. Boyes, L. Weisgerber, S., and Schafer, C.

1986 Transfer between word processing systems. Pp. 67-71 in M. Mantei and P. Orbeton, eds., Proceedings CHI'86 Human Factors in Computer Systems. New York: Association for Computing Machinery.

Kieras, D. E.

1982 A model of reader strategy for abstracting main ideas from simple technical prose. Text 2:47-82.

Kieras, D. E. and Bovair, S.

1986 The acquisition of procedures from text: A production-system analysis of transfer of training. Journal of Memory and Language 25:507-524.

Kieras, D. E., and Polson, P. G.

1985 An approach to the formal analysis of user complexity. International Journal of Man-Machine Studies 22:365-394.

Kim, W. S., Ellis, S. R., Tyler, M., and Stark, L.

1985 Visual enhancements for telerobotics: perspective parameters. Pp. 807-812 in Proceedings of the International Conference on Systems, Man, and Cybernetics. IEEE Catalog No. 85CH2253-3, November.

Kosslyn, S. M.

1985 Graphics and human information processing: A review of five books. Journal of the American Statistical Association 80:499-512.

Newell, A., and Card, S. K.

1986 The prospects for psychological science in human computer interaction. Human-Computer Interaction 1:209-242.

Newell, A., and Simon, H. A.

1972 Human Problem Solving. Englewood Cliffs, New Jersey: Prentice-Hall.

Polson, M. C., and Richardson, J.

1987 Foundations of Intellligent Tutoring Systems. In press. Hillsdale, NJ: Lawrence Erlbaum Associates.

Polson, P. G.

1987 A quantitative theory of human-computer interaction. In J. M. Carroll, ed., Interfacing Thought. In press. Cambridge, MS: Bradford Books/MIT Press.

Polson, P. G., Bovair, S., and Kieras, D. E.

1987 Transfer between text editors. Pp. 27-32 in P. Tanner, and J. M. Carroll, eds., Proceedings CHI'87 Human Factors in Comoputer Systems. New York: Association for Computing Machinery.

Polson, P. G., and Kieras, D. E.

1985 A quantitative model of the learning and performance of text editing knowledge. Pp. 207-212 in L. Borman, and B. Curtis, eds. in, Proceedings of the CHI 1985 Conference on Human Factors in Computing. New York: Association for Computing Machinery

Polson, P. G., Muncher, E., and Engelbeck, G.

1986 A test of a common elements theory of transfer. Pp. 89-83 in M. Mantei and P. Orbeton, eds., Proceedings CHI'86 Human Factors in Computer Systems. New York: Association for Computing Machinery.

Postman, L.

1971 Transfer, interference, and forgetting. In J. W. King and L. A. Riggs, eds., Woodworth and Scholsberg's Experimental Psychology. New York: Holt, Rienhart, and Winston.

Richardson, J. J. Keller, R. A., Maxion, R. A, Polson, P. G., and DeJong, K. A.

1985 Artificial Intelligence in Maintenance: Synthesis of Technical Issues. (AFHRL-TR-85-7) Brooks Air Force Base, TX: Air Force Human Resources Laboratory.

Shortcliffe, E. H.

1976 Computer Based Medical Consultations: MYCIN. New York: American Elsevier.

Simon, H. A.

1975 Functional equivalence of problem solving skills. Cognitive Psychology 7:268-286.

Singley, K., and Anderson, J.

1985 Transfer of text-editing skills. International Journal of Man-Machine Studies 22:403-423.

Sleeman, D., and Brown, J. S.

1983 Intelligent Tutoring Systems. London: Academic Press.

Smith, D. C., et al.

1983 Designing the STAR user interface. In P. Degano and E. Sandewall, eds., Interactive Computer Systems. Amsterdam: North-Holland.

Smith, S. L. and Moser, J. N.

1984 Guidelines for Designing User Interface Software. EDS-TR-86-278, MTR-10090. Bedford, MS: The Mitre Corp.

Thorndike, E. L.

1914 The Psychology of Learning. New York: Teachers College.

Thorndike, E. L. and Woodward, R. S.

1901 The influence of improvement in one mental function upon the efficiency of other functions. Psychological Review 8:247-261.

Tufte, E. R.

1983 The Visual Display of Quantitative Information. Cheshire, CN: Graphics Press.

van Kijk, T. A., and Kintsch, W.

1983 Strategies of Discourse Comprehension. New York: Academic Press.

Walraven, J.

1985 The colours are not on the display: a survey of non-veridical perceptions that may turn up on a colourdisplay. Pp. 35-42 in Displays.

Waterman, D. A.

1986 A Guide to Expert Systems. Reading, MS: Addison-Wesley Publishing Company.

Ziegler, J. E., Vossen, P. H., and Hoppe, H. U.

1986 Assessment of Learning Behavior in Direct Manipulation Dialogues: Cognitive Complexity Analysis. ESPRIT Project 385 HUFIT, Working Paper B.3.3, Stuttgart, West Germany: Fraunhofer-Institut fur Areitswirtschaft und Organisation.

DRAFT

NOT FOR DISTRIBUTION, ATTRIBUTION OR QUOTATION

DESIGNING FOR THE FACE OF THE FUTURE: RESEARCH ISSUES IN HUMAN - COMPUTER INTERACTION

A COMMENTARY ON THE

HAYES AND POLSON PAPERS

Judith Reitman Olson

The University of Michigan

Ann Arbor, Michigan

DESIGNING FOR THE FACE OF THE FUTURE:

RESEARCH ISSUES IN HUMAN - COMPUTER INTERACTION

DRAFT

NOT FOR DISTRIBUTION, ATTRIBUTION OR QUOTATION

A COMMENTARY ON THE

HAYES AND POLSON PAPERS

The hardest part of generating a research agenda now for issues in human-computer interaction for the Space Station is not in finding important issues and unanswered questions that are in need of careful research. It is selecting those research issues and the approaches to them that will answer the questions we have in the year 2000. In the year 2000, we will have devices that we can only dream of today; the Space Station environment will have a mission, size, and complexity that today we can only begin to sketch out. Our job, therefore, is not to recommend a research program that will answer specific questions that we know will arise in the design of the future Space Station. Rather, it is to prepare for that future with a research plan that lays the foundation, a sound theoretical base, that will make specific results both easy to predict and simple to confirm empirically. Additionally, the research has to produce a development environment, a flexible hardware platform and programming environment, that allows rapid prototyping for empirical testing and easy final implementation. These bases will serve us well when we have to make specific designs for the year 2000.

INTERFACES OF THE SYSTEMS OF THE FUTURE

It is important to begin by noting those things that are likely to be different in the Space Station environment than they are in the environments we focus our research on now. The most obvious differences, well discussed by both Polson and Hayes, are that the Space Station environment is weightless (with concomitant difficulties in forceful action and countermotion), perhaps noisy (with difficulties for the implementation of speech recognition and sound production), and complex (with a small number of people doing many, varied tasks with the help of computers, some of which they will be expert in, some of which they will not).

In addition, the tasks performed in the Space Station differ in other, more fundamental ways from the tasks we use today in our laboratory research on human-computer interaction. By far the largest amount of current research focuses on the behavior of people doing operational tasks: wordprocessing, spreadsheet formulation and analysis, database search in support of constructing a report. Our current research focuses on office tasks.

The Space Station, in contrast, is likely to have very little need for operational tasks; standard everyday tasks are more likely to be accomplished by ground personnel. Space Station personnel are more likely to be involved in:

- the monitoring and control of onboard systems (e.g., life support, experiment/manufacturing control),

- the occasional use of planning and decision systems (e.g., expert systems for medical diagnosis or for planning for changes in the mission), and

- The nearly constant use of communication systems (i.e., for both mission related information and for personal contact with friends and family), for both synchronous conversation and asynchronous messages.

ISSUES

There are important research issues that are common among these systems and the operational systems that we focus on today, but there are other, additional issues that are unique, requiring particular emphasis. The common issues, important to all future human-computer interaction, include:

1. How to design a system that is easy to learn and easy to use. One core feature of such a system is "consistency". Polson's paper makes the case for consistency -- a detailed argument for the importance of specifically modeling the user's goals and the methods necessary to accomplish the goals with a particular system. This is a very important research approach that promises

to give the right level of answers to questions about consistency that will arise in future designs.

2. A second core feature in making a system easy to learn and use involves a straightforward "mapping" between the way the user thinks about the task objects and actions and the way the system requires the user to specify them. For example, the mapping between the objects of wordprocessing, such as letters, words, and sentences, correspond much more closely to the objects in a visual editor than they do to the strings and line objects of a line editor. Moran (1983) has made a beginning in delineating this type of analysis; more theoretical work and empirical verification is necessary.

3. How to make decisions about what modes of input/output (and their combinations) are appropriate for a given environment and task. Hayes' paper discusses a number of considerations that must be taken into account when deciding among speech/visual/keyboard input and output modalities, as well as the use of appropriate combinations of these modalities.

4. What characteristics of the human information processor are primary determinants of the range of acceptable interface designs. One way of evaluating a design of an interface is to analyze it on the basis of the major processing that a user engages in in order to understand the output and generate the next input. For example, we can analyze an interface for its

perceptual clarity (e.g. adherence to Gestalt principles of grouping for meaning), its load on working memory (e.g., how many sub-goals or variables must be retained for short periods in order for users to accomplish their goals), and its requirements for recall from long-term memory (e.g., how many specific rules must be learned and how similar they are to each other). This approach, the cognitive science of human-computer interaction, by its generality across all application interfaces, promises to provide a theoretical thread through a number of empirical investigations. With a body of empirical tests of its predictions, this approach can both provide a robust base for future design situations and grow in sophistication and precision as a base for understanding complex cognition, even outside the domain of human-computer interaction.

Progress on these topics will make substantial contributions to our understanding of how to design human-computer interfaces for the Space Station in the year 2000, just as they will for those interfaces in offices and on the factory floor.

As discussed above, however, the systems on the Space Station are less likely to include operational systems, like those used in research on the above "common" topics, and more likely to include planning and decision, monitoring and control, and communication systems. Additional, important research issues arise in considering these latter three types of systems:

1. What characteristics of an interface appropriately alert users to abnormal situations in systems that must be monitored. What advice, information, or immediate training can be given users of a monitoring system that will guide them to behave in a creative but appropriate manner.

2. How are voice, video, keyboard, pointing devices, etc. to be used singly and in combination in each of these three types of systems? Certainly voice and video have begun to be explored in synchronous communication systems (e.g., picturephone and slow-scan video teleconferencing). How can these modalities be used to best advantage to support the need for long-term contact with friends and family when individuals are separated for a long time? How are privacy issues accommodated in such systems, both for personal communication and operational communication?

3. If users have to consult an expert system or if some intelligence is incorporated into a system, how is information conveyed to the user about whether the system is to be believed? Since current intelligent systems are "fragile," that is, easily put in situations for which their advice is not appropriate, we need to convey to the user information about the system's boundaries of capabilities. Or, better yet, we need to build intelligent facilities that allow the user to query or access the stored knowledge in ways that can make the advice fit new situations more flexibly.

4. Since the systems that Space Station users must deal with will be varied and the users will have varying expertise in either the task at hand or the particular system to be used, it is important to have the system provide requisite context or training. Training need not be a formal module that one accesses explicitly, as software training modules are designed today. The systems could be initially designed to be transparent (i.e, with objects and actions that fit the way the user thinks about the task), not requiring training. Or, they could be built to include a "do what I mean" facility or embedded "help" or "training" facilities, accessible either when the user requests it or when the system detects that the user is confused or doing things inefficiently.

5. Most of the current theoretical bases for the design of human-computer interfaces consider tasks that are well-known to the user. The GOMS analysis of Card et al. (1983), for example, is for skilled cognition. Kieras and Polson's (1985) production system formalism similarly considers only skilled performance of cognitive tasks. However, in the Space Station environment, users will be doing few routine tasks. They will be doing tasks that involve novel situations, situations that invoke creative problem solving, not routine cognitive skill. Space Station personnel, for example, may try to alter a system that their monitor has shown is malfunctioning; they may use the advice of a medical expert system to attend to a colleague who has an undiagnosed illness; they may use communication channels to acquire additional expertise from the ground crews to solve onboard problems or plan new missions.

In order to understand how these interfaces should be designed, more emphasis should be made in research in the area of human problem solving. The focus should be, for example, on how to build systems that, minimally, do not interfere with the information the person needs to keep track of during complex problem solving. Ideally, we want to be able to build systems that augment a person's abilities to explore and evaluate new actions in novel situations.

6. Furthermore, as Hayes' paper points out, most of our current research on human-computer interaction focuses on the use of a system as a "tool" not as an "agent." Our understanding of cooperative human behavior is woefully thin. Theoretical bases need to be established so that we can build systems that cooperate well with the human problem solver, so that systems can augment the intelligent human to produce an even greater level of understanding and action.

APPROACHES FOR THE UNDEFINED FUTURE

As stated at the beginning of this discussion, the most difficult aspect of the task of listing research issues that the Space Station of year 2000 will benefit from concerns predicting the Space Station environment and the technology that will be available at the time. We just don't know what the alternative design elements will look like. The best we can do at this time, therefore, is to recommend a research agenda whose results promise to be useful no matter what the environment and technology will be. At the core of these recommendations is research that centers on the capabilities of the human information processor, both as an individual and in a cooperative environment. The human will not have changed substantially by the year 2000.

Consequently, our understanding of human-computer interaction will benefit from research that accumulates results from a common theoretical core that:

1. delineates in detail the functioning of the human information processor, with particular emphasis on the interaction among cognitive resources and those resources involved in attention (for monitoring systems), problem solving (for expert systems and decision support systems), and communication,

2. within the domain of expert systems, explores the information a user needs and determines how it should be presented so that the user can assess the believability of the advice given, and

3. determines ways to help casual users of a variety of systems to use them without a great deal of "start up" effort, either through transparent design; effective, easy training; or embedded intelligent aids.

A salient aspect of this type of research is that it is based on cognitive models, not on design principles. Cognitive models allow the examination of the interaction of features of the task or interface, which principles cannot do. These cognitive models characterize details of what the task requires and details of the human information processor. By running these models, the designer or researcher can determine in detail areas of difficulty in the interaction (e.g., where the working memory is overloaded with subgoals and parameters to be retained). Certain changes to the interface design could be tested by running these models without having to invest in the expense of a full-fledged usability study. The number of researchers approaching issues in human-computer interaction with cognitive models is currently very small; their numbers should be encouraged to grow.

Furthermore, research should have as one of its goals the transfer of the knowledge developed in the laboratory to the design and development process. This calls for development of:

1. analytic tools for assessing consistency in a particular design.

2. analytic tools for assessing the amount of effort required in mapping the users' natural way of thinking about the task (i.e., an object/action language) into that required by the system, and

3. guidelines that will assist the designer in decisions about which modality or combinations of modalities are appropriate for a particular task and situation.

And, if systems are to be built for an evolving future, they must be built with scars and hooks, as Hayes notes. Software should be designed so that it has places that will allow easy growth in capabilities or input/output devices. Furthermore, research is needed to develop:

1. a method and language that allows the system designer to incorporate good human factors into the target system (e.g., a "toolkit" with components that have been designed with consideration for research on their human factors), and

2. a method that allows system developers to rapidly implement trial interfaces, so that they can be tested with real endusers, and then turned quickly into production code.

It is clear from the papers in this session that funds devoted only to simple empirical studies of users' behavior with new, increasingly complex technology will not be sufficient for answering the questions of the year 2000 and beyond. In contrast, research that focuses on:

1. the abilities of the human information processor with concommitant widespread, specific, robust cognitive modeling, and

2. additions to the development life cycle to make the production of good software rapid

can produce research that can make the human-computer interfaces on the Space Station of the highest possible quality for their time.

REFERENCES

Card, S. K., Moran, T. P., and Newell, A.

1983 The Psychology of Human - Computer Interaction, Hillsdale, N. J.: Erlbaum.

Kieras, D. E., and Polson, P. G.

1985 An approach to the formal analysis of user complexity. International Journal of Man - Machine Studies 22:365-394.

Moran, T.

1983 Getting into a system: external-internal task mapping analysis. Pp. 45-49 in Proceedings of the 1983 CHI Conference on Human Factors in Computing. New York: ACM.

SESSION 3:

LANGUAGE AND DISPLAYS FOR HUMAN : COMPUTER COMMUNICATION

SYNOPSIS OF GENERAL AUDIENCE DISCUSSION

Judith Reitman Olson

LANGUAGE AND DISPLAYS FOR HUMAN : COMPUTER COMMUNICATION

SYNOPSIS OF GENERAL AUDIENCE DISCUSSION

Two general points were raised from the floor.

1. When discussing natural language interfaces for human-computer interaction, one should make a clear separation between those requiring auditory input and those accepting natural language. Although these two features are highly correlated, they need not be. One could consider a speech input that would restrict language to a subset, such as single word commands or even special codes. Similarly, there could be natural language input that was entered via keyboard. Although there is an additional memory load imposed on the user if speech input accepted only a subset of natural language, there may be some applications that could effectively use this mode.

2. Allen Newell wished to emphasize the importance of having specific, detailed cognitive models as the basis for designing human-computer interfaces. The current researchers who are using this approach is very small, and though growing exponentially, the growth rate is very "leisurely." The approach has the advantage of not only specifying details of the processing mechanisms of cognition and their interaction, but also of specifying the details of the task the user is engaged in. Having the details of the task can provide benefits beyond redesign of the interface. They could serve as the basis from which the task itself could be

redesigned, affording productivity enhancements from a straightforward efficiency analysis. Newell recommended a strong incentive be established for researchers to conduct their work in the context of cumulative, model-based theories of cognition, and let the design principles fall from them.

SESSION 4:

COMPUTER AIDED MONITORING & DECISION MAKING

Paper: Randall Davis, MIT

Paper: Baruch Fischhoff, Eugene Research Institute

Discussant: William Howell, Rice University

DRAFT

NOT FOR DISTRIBUTION, ATTRIBUTION OR QUOTATION

ROBUSTNESS AND TRANSPARENCY IN INTELLIGENT SYSTEMS

Randall Davis

Massachusetts Institute of Technology

Cambridge, Massachusetts

CONTENTS

[pp. 398-399 DO NOT EXIST]

SUMMARY

ACKNOWLEDGMENTS

NOTES

REFERENCES

FIGURES

INTRODUCTION

Developing and building a space station will confront problems of significant complexity in an extraordinarily demanding environment. The station's size and complexity will make necessary the extensive use of automation for monitoring and control of critical subsystems, such as life support. The station complexity, along with the novelty of space as an environment, means that all contingencies cannot be anticipated. Yet the hostility of the environment means the consequences of failure can be substantial.

In such situations, robustness and transparency become essential properties of the systems we develop. A system is robust to the degree that it has the ability to deal with unanticipated events. A system is transparent to the degree that its operation can be made comprehensible to an observer.

This paper is concerned with these two properties -- robustness and transparency -- from a number of perspectives. We claim that they are crucial to the space station undertaking (and indeed to any situation with similar levels of complexity and similar consequences of failure). We argue that they are fundamental properties of models and system designs based on those models. As a result, robustness and transparency cannot easily be grafted on afterward; they must be considered at the outset and designed in. We explore how this might happen, i.e., how these two properties translate into constraints on system design and describe a number of research efforts that may lead to better understanding of how such design might be accomplished.

It is useful at this point to establish some simple vocabulary. By "system" or "device" we mean the hardware whose behavior we wish to understand and control. The power distribution system, for example, would include all the cables, batteries, fuel cells, solar arrays, switches, etc., that supply power to the station. By "model" we mean a description of that hardware that will allow us to analyze, interpret, diagnose, and guide its behavior. The model may be implicit in a program designed to monitor the hardware or it may exist in the mind of the human doing the same job. When expressed explicitly, it is typically written in terms of schematics, performance curves, engineering drawings, etc. The model also may be implicit in a program designed to monitor the hardware or it may exist in the mind of the human doing the same job. In any case it provides the basic framework used to understand the device.

While we speak broadly of systems and models, our concern here is for the most part with systems of physical devices and the associated engineering models of them; much of what we say is likely to carry over to software as well. Models of human behavior and social systems are largely beyond what we attempt to do here.

Unanticipated Events: Motivation

Because much of what we discuss is motivated by the difficulties of dealing with unanticipated events, it is worth taking a moment to consider what they are and why they are important. By unanticipated events we mean any occurrence requiring a response that has not been previously planned for, analyzed, and the appropriate response determined.

One compelling example might occur if the life support system monitors present a collection of readings that indicate a malfunction but do not match any known pattern of misbehavior. The readings need to be analyzed and an appropriate response initiated, yet this cannot be done "by the book;" it requires that we reason through what could have happened to produce such readings.

The importance of such events arises from their inevitability, due to both the complexity of the space station and the novelty of the environment. Unanticipated events and interactions are a fact of life for complex, large scale systems because the number of different kinds of things that can go wrong is so vast, and our ability to do exhaustive formal analyses of fault events has rather modest limits. Space is a sufficiently novel environment that we have no comprehensive catalog of standard fault models that can be checked ahead of time.

Unanticipated Events: Example

During STS-2, the second space shuttle mission, an interesting sequence of events lead at one point to the recognition that a fuel cell was failing and later to the realization that in its degraded state it could conceivably explode. This sequence of events helps to illustrate both the inevitability of unanticipated events and the kinds of knowledge and reasoning needed to deal with them.

Some brief background will help make the events comprehensible. The basic function of the 3 fuel cells (Figure 1) is to produce electricity by combining hydrogen and oxygen in a carefully controlled reaction using potassium hydroxide as a catalyst. The combustion product is water, removed from the cell by the water removal system (Figure 2): damp hydrogen enters the condenser at the right, pulled along by the flow produced by the motor and pump at left. The motor is also turning a separator that pushes condensed water droplets toward the walls of the chamber where they accumulate due to surface tension (recall this is a 0g environment). The now drier hydrogen returns to the fuel cell, while the annulus of water continually being formed at the separator is picked up and guided to the water storage area. A meter at the outlet monitors water pH, checking for contamination (e.g., potassium hydroxide from the fuel cell), since the water is intended for consumption.

In very much abbreviated form, the sequence of events leading to early mission termination of STS-2 proceeded as follows (Eichoefer, 1985):

Pre-Launch: During pre-launch activities, the fuel cell pH meters register high. Interpretation: Familiar, unexplained anomaly.

Pre-Launch: At various times oxygen and hydrogen flow meters read high; at one point oxygen flow goes off-scale. Interpretation: Sensors malfunctioning.

+ 3:00 Fuel cell 1 (FC1) begins to shed load; the other two assume more load.

Interpretation: Cell may be failing.

Controllers consider purging FC1. Degraded performance suggests possible flooding; pH high also suggests flooding; purging will remove water. Purging FC1 rejected -- purged KOH might solidify, blocking purge line that is common to all 3 cells.

+ 3:25 Crew asked to test pH manually. If sensor is correct, potable water may be getting contaminated by KOH.

+ 4:25 Crew too busy with other duties to perform test.

+ 4:40 FC1 off loads significantly

Interpretation: Clear failure.

+ 4:51 FC1 isolated from remainder of electrical system and shut down.

+ 5:48 Mission evaluation room recognizes new failure mode for the cell in the current situation. Once it is shut down pressure slowly drops, but can drop at different rates on each side. If pressure differential becomes large enough, gas bubbles from one side can cross to the other, possibly combining explosively.

+ 7:52 FC1 restarted with reactant valves closed; reactants consumed and voltage in cell drops to 0.

Post-mission analysis of the fuel cell and water separator revealed that the pH meter had been working correctly and that a small particle blocked the nozzle in the water separator of cell 1, preventing water removal to the storage area. The water backed up first in the separator and later in the cell, flooding the cell (hence the high pH), leading to performance degradation, consequent load shedding, and eventual failure.

Lessons From The Example

This example is useful for a number of reasons. It illustrates, first, robustness and transparency in the face of unanticipated events. The reasoning was robust in the sense that the blockage had not previously been anticipated, yet engineers were able to reason through how the device worked, and were able to recognize and predict a novel sequence of potentially serious consequences. The reasoning was transparent in the sense that the story above is comprehensible. Even given the very small amount of information in Figures 1 and 2 and the short description above, the description of the events "makes sense."

Second, it suggests the difficulty of a prior identification and analysis of all failure modes and all the ways those failures may combine. Even with all the careful design, testing, and previously experience with fuel cell technology, a new mode of cell failure was encountered.

Third, it illustrates the kind of knowledge and reasoning that was required to understand, diagnose, and repair the problem. The knowledge involved information about structure (interconnection of parts) and behavior (the function of a component labeled "motor" or "pump"), supplied by the diagrams in Figures 1 and 2. Knowledge of basic chemistry and physics was also involved, used to understand the behavior potassium hydroxide in solution and the notion of surface tension. Importantly, the reasoning relies on causal models, descriptions of devices and processes that capture our ordinary notion of what it means for one event to cause another (e.g., the motor causes the pump to turn which causes the hydrogen and water to move through the condenser, etc.).

The reasoning involved was of several varieties., The fourth event above, for instance, illustrates reasoning about behavior to predict consequences: if the cell is flooded, potassium hydroxide can get in the water, meaning it can get to the water separator and then into the water storage. Another form of reasoning involved working from observed symptoms to diagnoses and then to repair actions: If FC1 is shedding load, it's an indication of degraded performance, which suggests flooding. Flooding in turn suggests purging as a repair. Simple knowledge of connectivity and chemistry ruled out that action in the event above at + 3:00: it might have blocked the common purge line.

Finally, it offers a simple way summarizing much of what this paper is about: while all of the reasoning above was done by people using their models of the devices in question, we suggest giving computers exactly the same sort of knowledge and reasoning abilities. They could, as a result, perform as far more effective assistants.

We believe this can be done by supplying them with something like the diagrams of Figures 1 and 2, with knowledge about structure, behavior, an understanding of causality, chemistry, physics, electronics, and more. In short, we need to give them the same understanding of "how things work" that we use in everyday engineering reasoning.

The aspiration, of course, is easy, execution is considerably more difficult; this is clearly no small undertaking. In the remainder of this paper, we examine some of the research issues that arise in attempting to make this happen.

- How can we provide descriptions usable by a machine that are equally as rich as those in Figures 1 and 2? Consider, for example, how much knowledge is captured by the simple labels motor, pump, and condenser.

- How can we provide the kinds of reasoning abilities displayed above?

- How can we provide the ability to judiciously select the correct model for a given problem? Consider how our view shifted from one grounded in physics, to one oriented towards chemistry, to one grounded in electronics, as the need arose.

o How can we provide the ability to simplify a complex model, selecting out just the "relevant" details? Consider what a drastic, yet useful, simplification Figures 1 and 2 are of the actual devices. (Consider too what a misleading statement it was, above, to say "Even given the very small amount of information in Figures 1 and 2 ..., the description of the events makes sense." It makes sense precisely because the right level of detail was chosen. How might we get a machine to do that?)

o For that matter, how do human engineers do all these things?

Unanticipated Events As A Focus

Unanticipated events like the blockage of the water separator are an appropriate focus for this paper because this symposium aims to identify research issues for future attention rather than incremental improvement to current practice. Some useful techniques already exist for simulation, fault insertion, and creation of error recovery procedures for foreseeable events. Additional work is in progress on techniques for error avoidance and in designing systems that are error tolerant. There is also a well-established approach to producing robustness through man-machine combinations: divide the work so that the more routine tasks fall to the machine and rely on the human for resourceful responses to atypical events. All of these are appropriate, important, and will continue to contribute to system design.

But new research issues arise in part by asking what relevant things we don't know how to do very well, or at all. From that perspective, unanticipated events present a set of interesting and important challenges, providing an appropriate focus for this paper.

They also lead to increased concern about transparency. Other rationales already exist for transparency, including giving users an understanding of the system's reasoning so they know when to rely on the conclusions, and the importance of keeping the system accessible to human comprehension and possible intervention. Dealing with unanticipated events adds additional motivation, most visible in the question of system override: to determine whether a system's response is based on inappropriate assumptions (e.g., an inappropriate model), we need first to know what those assumptions are. Transparency helps make this possible.

Agenda

Our discussion now proceeds in three basic steps. First, to help make clear the difficulties involved in robustness, we explore briefly some non-solutions to the problem. Second, we identify two broad categories of attack that are likely to offer some leverage on the problem: developing models and reasoning methods powerful enough to handle unanticipated events, and developing techniques for coping with situations where only imperfect models are available. Finally, we describe a number of specific research topics that will help to develop the models, methods and techniques needed to produce robustness and transparency.

SOME NON-SOLUTIONS TO THE PROBLEM

Before proposing a new attack on a problem, it's worth asking whether the problem can be tackled with known techniques. We consider three plausible approaches and explore why each of them fails to provide the degree of robustness we believe is necessary.

One traditional approach is the use of man-machine combinations, relying on the human to handle non-routine situations. This is, of course, useful and can be quite effective over a range of problems. In the fuel cell problem of STS-2, for instance, routine monitoring was handled automatically, while exceptions were analyzed by human experts.

It is also clear, however, that systems currently being designed and used are sufficiently complex that this will no longer be sufficient, unless we can make our automated assistants smarter. Some nuclear power and chemical processing plants, for instance, are complex enough that non-routine events lead to massive overload on human information handling abilities. So many alarms were triggered during the Three Mile Island accident, for instance, that not only was it effectively impossible to interpret them, even detection became problematic as multiple alarms masked one another. Somewhat more immediately relevant, during shuttle mission STS-9 an alarm was triggered more than 250,000 over 3 days, due to an unanticipated thermal sensitivity in a Spacelab remote acquisition unit, along with an oversight in user software.

It is likely that similar and perhaps higher levels of complexity will be involved in the space station. As a result, we need to do more than rely on the human half of the team to handle all exceptions. We need to upgrade the ability of our machines to interpret, diagnose, and respond to unanticipated events, enabling man-machine combinations to remain effective in the face of complex systems and novel environments.

A second route of attack on the problem might appear to be the creation of more reliable software through improved software engineering, program verification, or automatic programming[1] Unfortunately all of these solve a problem different from the one at hand here. The issue is illustrated in Figure 3: techniques for production of reliable software all assist in ensuring that a program matches its specifications. Unanticipated events, however, will by definition not show up in the specifications. The problem here is not so much one of debugging code, it is the creation and debugging of the model and specifications.

Finally, given its wide popularity, we might ask what expert system technology[2] might be able to contribute to the difficulties we face. Here too the answer is that they have little to offer. The fundamental limitation in these systems arises from the character of the knowledge they use. Traditional expert systems gain their power by collecting empirical associations, if-then rules that capture the inferences human experts have learned thru experience. We refer to them as empirical associations to indicate the character of the knowledge they capture - associations, typically between symptoms and diseases, gathered as a result of human experience.

Importantly, those associations are typically heuristic rather than causal; i.e., they capture what experts have observed to happen without necessarily being able to explain why it should be so. A medical diagnosis system, for example, might have a rule of the form "a college student complaining of fatigue, fever, and sore throat is likely to have mononucleosis." The rule offers useful guidance even if the experts cannot provide a detailed causal (i.e., physiological) explanation for why the conclusion follows. Indeed the power of the technology comes in part from the assistance it provides in accumulating large numbers of fragmentary rules of thumb for tasks for which no well-defined causal theory exists.

One important consequence of this kind of knowledge, however, is a kind of brittleness. Current generation expert systems are idiots savant, providing impressive performance on narrowly defined tasks and performing well when the problem is exactly suited to the program's expertise. But performance can degrade quite sharply with even small variations in problem character. In general the difficulty arises from a lack of underlying theory: since the rules indicate only what conclusions follow and not why, the program has no means of dealing with cases that "almost" match the rule, or cases that appear to be "minor" exceptions. Indeed, they have no notion of what "almost" or "minor" could mean.

"FIGURING IT OUT"

Having reviewed some existing technology that does not appear capable of providing the degree of robustness needed, we turn now to considering what kinds of ideas and technologies would help solve the problem.

The basic thrust of our argument is quite simple. As size and complexity of systems increase, we see a decrease in the opportunity to do an exhaustive a priori analysis and pre-specify appropriate responses. The space station will likely be complex enough to preclude such analysis; the novelty of the environment increases the chance of unanticipated challenges.

To deal with such situations we need a new approach to building intelligent systems, one based on a simple premise: when you can't say in advance what will happen, the ability to "figure out" how to respond becomes much more important. Where knowledge-based systems, for instance, "know" what to do because they have been given a large body of task-specific heuristics, we require intelligent systems capable of figuring out what to do.

This ability should play a supporting role and is clearly not a replacement for existing approaches. Where we can anticipate and analyze of course we should, and where we can construct effective fault tolerant systems we should. But as system complexity grows and the number and seriousness of unanticipated events increases, we need the flexibility and breadth of robust problem solving systems to deal with them.

The key question, of course, is how to construct systems with this property. In the remainder of this paper we suggest several ways of looking for answers to that question.

MODELS AND ENGINEERING PROBLEM SOLVING

Faced with an unanticipated event in a complex system, a powerful way to figure out what to do is by reasoning from an understanding of the system, a model of "how it works." A behavioral model, for instance, can be of considerable help in dealing with complex software like an operating system. In dealing with a complex physical device, a model of structure and function (schematics and block diagrams), along with an understanding of causality can be essential in understanding, interpreting and debugging behavior[3].

How might we proceed, for example, when faced, with a set of sensor readings from the fuel cells that indicate malfunction but do not match any known pattern of misbehavior? The most robust solution appears to be grounded in knowing how it works, i.e., creating and using models that capture structure, behavior, and causality at an appropriate level of detail. We need to know what the component pieces are, how they each work, how they are interconnected, and so forth.

We argue that, in the most general terms, the creation, selection, and use of appropriate models is the most powerful approach to the problem[4]. It is in many ways the essence of engineering problem solving. Since, as we discuss in more detail below, models are abstractions, the process of model creation and selection is essentially one of deciding which abstraction to apply. Faced with a complex system to be analyzed, an engineer can bring to bear a powerful collection of approximations and abstractions.

As a relatively simple example in electrical engineering, for instance, an engineer may decide to view a circuit as digital or analog, linear or non-linear. But even to approach the problem as one of circuit theory means we have made the more basic assumption that we can model the circuit as if signals propagated instantaneously, and hence ignore elecrodynamic effects. Models and their underlying abstractions are thus ubiquitous in this kind of problem solving.

We believe that an important source of power in the problem solving of a good engineer is the ability to create, select, use, and understand the limits of applicability of such models. Consequently, we believe that a powerful approach to building robust problem solving programs is to identify and capture the knowledge on which that modeling ability is based. Similarly, a powerful approach to building transparent problem solving problems is to make that knowledge explicit in our programs. One general thrust of the research we suggest is thus broadly concerned with advancing our understanding of model creation, selection, and use, and demonstrating that understanding by creating programs capable of doing such things.

A second general thrust is made feasible by the fact that the space station is an engineered artifact, a device intended to accomplish a specific purpose whose design is under our control. As a result, we can also ask, how can we design in such a fashion that dealing with unanticipated events is easier? That is, given the inevitability of encountering such events and the difficulty of reasoning about them in complen systems, how should we design so that the reasoning and analysis task becomes easier? We speculate, for instance, about what "design for comprehensibility" might mean.

Other approaches we discuss that share the same basic mindset include understanding (and hence capturing in programs) "common sense" physical reasoning, and exploring the origins of robust problem solving in people, whose graceful degradation in performance is so markedly different from the behavior of automated systems.

We refer to this set of approaches as "making the best situation" because they have in common the assumption that it is in fact possible to model the system and approach the problem by asking how we can facilitate model creation and use.

But what about the alternative? how can we get robust behavior in situations where no effective model yet exists, in situations where the only available models are incomplete or insufficiently detailed for the task at hand? We term that set of alternatives "making the best of the situation," to suggest that, lacking a model to reason from, we have to fall back on some less powerful methods. In this we speculate very briefly about research in using multiple, overlapping but incomplete models.

MODELS AND PROGRAMS

Since much of our discussion is focused on models -- creating them, using them, and determining their limitations -- it is worth taking a moment to review briefly some of their fundamental properties. Since we will for the most part be concerned with embodying those models in computer programs, it is similarly worth reviewing briefly the relation between models and programs, understanding the role the computer plays in all this.

The Role of the Computer

Let's start with the role of the computer. Given the size and complexity of the space station, extensive use will have to be made of software to automate tasks like monitoring and control. Any such program inevitably embodies a model of the task at hand. Even a program as simple as one that monitors CO_2 and displays a warning when the level exceeds a threshold has, implicit in it, a much simplified model of the sensing device, the environment (e.g., that CO_2 is uniformly dispersed), what levels of CO_2 are safe, etc. Since models and computer programs are often so closely intertwined, it is important to understand what the model can contribute and what the computer can contribute.

The computer brings to the task a number of useful properties. It offers, for example, a vast increase in information processing power. This power, in turn, makes possible the construction and use of models that are orders of magnitude larger than any we could create by hand. The power is useful even with simple models, where it makes possible determining less obvious consequences, as in cases when straightforward search in chess can determine the long-term consequences of a move.

The computer has also facilitated the construction of many different kinds of models, including those that are non-numeric. As a result of work in computer science generally and AI in particular, we now routinely build and compute with models that are symbolic, qualitative, and incomplete. Symbolic models embody non-numeric inferences (e.g., "if the current shuttle pilot is Joe, on screen 1 display elapsed time, Houston time, and fuel levels "). Qualitative models[5] describe and reason about behavior using the language of derivatives commonly employed by engineers (e.g., "if the voltage at node N3 increases then rate of discharge of capacitor C4 will decrease"). Most current expert systems are based on models that are incomplete, in the sense that they cover a number of specific cases (e.g., "if valve V3 is open and the tank temperature is high, then close valve V4"), but may leave unspecified what action to take in other cases (e.g., what to do if V3 is closed).

Work in AI and cognitive science has facilitated understanding and capturing other types of models as well, including mental models[6], the vastly simplified, occasionally inaccurate but effective representations of mechanism and causality that people use in dealing with the world. My mental model of how the telephone system works, for instance, is quite a bit different from reality, but quite useful.

The computer also brings to the table a strong degree of "mental hygiene." Models expressed in English and left to human interpretation produce a notoriously wide variety of conflicting results. The remarkably literal-minded character of computer-based models enforces a degree of precision that we might not otherwise achieve in areas outside of those handled with formal mathematical analysis.

Expressing a model in a program also makes it far easier to test it by example, since determining its predictions is a matter of running the program rather than working out the consequences by hand. This in turn facilitates finding ambiguities, oversights, and limitations, and thus aids in extending the model.

All of these are useful and important properties. But for our purposes even more important is what the computer doesn't bring to the task, what embodying the model in a program does not do. It does not by itself provide either robustness or transparency. Simply put, robustness and transparency are properties of models and systems, not properties of programs that may be monitoring or controlling those systems.

This simple observation has two important consequences. First, it means that we cannot create robust or transparent systems simply by developing software. It will not do, for instance, to design a highly complex system and then develop an equally complex piece of software that attempts to monitor, interpret, and perhaps control it. Layers of complexity will only make it more difficult to deal with novel situations.

Perhaps the simplest demonstration of the futility of this approach comes in dealing with events that may be outside the range of applicability of the program. The more complex the underlying system, the more complex the program needed to interpret it, i.e., the more complex the model of that system needs to be. And the more complex the model is, the more difficult it becomes to determine whether it is based on assumptions that do not hold for the current situation, and hence the current events are outside its range of applicability.

Second, if robustness and transparency are properties of models and systems, not properties of programs, it follows that they cannot be grafted on, they must be designed in. That is, we need to understand how to design in such a fashion that the resulting systems have those properties, and how to create models that have those properties. One of the research strategies we suggest in this paper is to turn this question around, and ask how the desire for systems with these two properties can be translated into constraints on system design. That is, is it possible to design in such a way that the resulting systems are easy to model robustly and transparently.

Robustness and Transparency in Models

We have argued that robustness and transparency are properties of systems and models rather than of programs and that a primary route to resourceful systems is the creation of models with these properties. But that isn't easy. To see why not, we examine the kinds of things that commonly get in the way.

Three common sources of failures of robustness are incompleteness, information overload, and incorrect level of detail. Models may be incomplete because information that should have been included was omitted. A particularly relevant example arose in the Solar Max repair during Mission 41-C. The initial attempt to attach to the satellite failed because additional, undocumented hardware had been added to the satellite near the attachment point, preventing the mating of the satellite and the attachment device. The lesson here is the obvious one: you can't reliably figure out what to do if your picture of the device in question is incomplete.

A second source of failure of robustness -- information overload -- occurs when information processing ability available is overwhelmed by the amount of data or the size of model. The data rate may be so high that it cannot be interpreted fast enough. The model itself may be so large that it outstrips the processing power available. The issue here is the same for man or machine: in either case the available processing power may be insufficient to use the model. The lesson here is the need to ensure that the models we build are computable with the power available.

Information overload is frequently a result of the third common source of failure: selecting the wrong level of detail, in particular choosing too low a level. Attempting to model the behavior of a digital circuit using quantum mechanics might be an interesting challenge, but would surely drown in detail. If, on the other hand, too high a level is chosen, the model omits relevant phenomena. For example, some circuit designs that are correct when viewed at the digital level may in fact not work due to effects that are obvious only when viewed at the analog level.

All of this leads us to a fundamental difficulty in designing and using models. Robustness depends in large measure on completeness of the model. Yet all models are abstractions, simplifications of the thing being modeled, so no model can ever be entirely complete. Nor in fact would we want it to be. Much of the power of a model arises from its assumption that some things are "unimportant details," causing them to be omitted. There is power in this because it allows us to ignore some phenomena and concentrate on others; it is this license to omit some things that reduces the information processing requirements of using the model to within tolerable levels.

But there is as a result a fundamental tension between completeness (and attendant robustness) and complexity. If we make no simplifying assumptions we drown in detail; yet any simplifying assumption we make may turn out to be incorrect, rendering our model incomplete in some important way. This in turn raises interesting questions, further explored below, including how we select an appropriate model, i.e., an appropriate set of simplifying assumptions, and how we might recover in the event that we select one that is inappropriate.

RESEARCH TOPICS

In this section we discuss in broad terms a number of research topics relevant to the overall goal of building systems that are both robust and transparent. For the most part, we proceed from the assumption that getting machines to assist in significant ways with reasoning about situations like the STS-2 fuel cell problem will require that they have appropriate models. We then ask how those models can be created and indeed how we can design the device from the outset in such a way that the model creation process is made simpler.

Model Selection and Creation

Selecting and creating models is perhaps the most fundamental issue in solving engineering problems and an important determinant of the robustness of the solution. It is a skill that is in some ways well known: it's what good engineers have learned to do through years of experience. The goal here is to understand that skill and experience well enough that it can be embodied in a program, allowing automated assistance in selecting and creating appropriate models.

In almost any design or analysis problem, the most basic question is how to "think about" the object in question, i.e., how to model it. Given the acknowledgment that all models are abstractions, it is futile (and as we have suggested, inappropriate) to seek perfect completeness and robustness. That in turn means that the modeling decision concerns what to pay attention to, i.e., what properties of the object are relevant to the task at hand and which can safely be ignored. Hence the goal is to find a model with two properties. First it should be complete enough that it handles the important phenomena. Second it should be abstract enough that it is computable and capable of producing a description at a useful level of detail (i.e., even if it were possible, it would be of little use to produce a picosecond, microvolt-level analysis of a circuit whose digital behavior is of interest). But naming the goal is easy; the research challenge is in finding a more precise understanding of what it means to "consider the task" and to determine when a model is "complete enough", "abstract enough", and at an appropriate level of detail.

One possible route to understanding the nature and character of models is to define the kinds of abstractions commonly used in creating them. This might be done by determining what kinds of abstractions are commonly (and often implicitly) employed by engineers. What are the rest of the terms like digital, analog, linear, etc.? Is there just an unstructured collection of such terms or is there, as we would guess, some sort of organizing principle that can be used to establish an ordering on them? If so, we might be able to say more concretely what it means to proceed from a more abstract to a more precise model and might be able to develop programs capable of such behavior. It is unlikely that there is a simple, strict hierarchy that will allow us to move in a single, unambiguous direction. Much more likely we will find a tangled graph of models; part of the task is to sort out the different kinds of interconnections likely to be encountered.

A second possible route to understanding the nature of models arises from the simple observation that models ignore details. Perhaps then different kinds of models can be generated by selecting different combinations of details to ignore. The task here is to characterize different "kinds" of details; the ideal set of them would not only generate known models but might suggest additional models as well.

By either of these routes -- studying the kinds of abstractions used or the kinds of details ignored -- we might be able to produce an array of different kinds of models. That brings us to the problem of model selection, determining which to use in a particular situation. Some assistance may be provided by knowing how the array of models is organized, i.e., what it means to be a "different kind of model." The difficulty arises in determining what the important phenomena are in the problem at hand and selecting a variety of model capable of dealing with it. How is it that a human engineer knows which approximations are plausible and which are likely to lead to error?

It is unlikely that we will ever be able to guarantee that the knowledge used for model selection is flawless or that the models given to the program are flawless. We thus need to confront the problem of detecting and dealing with models that are inappropriately chosen for the task at hand or that are incomplete in some relevant detail. Human engineers at times make the wrong selection or use a faulty model, yet are capable of detecting this and dealing with it. How might we get machines to do the same?

Finally, note that progress on model selection will have an important impact on the some-what loaded issue of system override. If, as we have argued, unanticipated events are inevitable, simply having a detailed model is not enough: events may occur that are outside the range of applicability of the model. This can be a particularly difficult problem because it concerns deciding "how to think about" the problem.

We argue that override is fundamentally a decision that a particular model is inappropriate. Consider the example of a program monitoring and controlling life support. We might be tempted to override its decisions if they seem sufficiently different from our own, but why should they differ? The most basic answer seems to be that the model the program is using to interpret sensor readings is inappropriate, i.e., based on assumptions that are not valid in the current situation.

The only objective way to discover this is by determining why that model was chosen, what approximations it embodies, and what the limitations are on those approximations. Since much of this information was used to make the model selection to begin with, leverage on the override problem can come from understanding model selection and, importantly, from making explicit both the model itself and the assumptions underlying it. This would give us reasonably objective grounds for the override decision, since the model and its underlying assumptions will be available, and can be examined and compared to the current situation. It also reminds us how important it is that such information be made explicit, rather than left implicit in the program code or the mind of the program author.

Model Specification Needs To Be Less Trouble Than It Is Worth

We have repeatedly stressed the importance of models as a basis for robust reasoning about complex systems. But specifying those models is not an easy task, for several reasons. At the simplest level the issue is volume: there is an enormous amount of information to be captured. Existing design capture systems don't deal well with the problem because they don't make the information collection process easy enough, nor do they offer sufficient payoff once the information is entered to provide a motivation for doing it. They are in general more trouble than they're worth.

For design changes in particular, it is today often easier simply to try out the change and then (maybe) go back and update the specification database. In the case of Solar Max, for instance, perhaps no one knew about the additional hardware because it had been added at the last minute and never documented. The problem of documenting code is similar: it's often easier to try it out, then document. Often the documentation never gets done because it simply isn't viewed as critical to the undertaking.

The problem is both organizational and technical. Organizational issues arise because design documentation is typically of least use to the original designer, who is most familiar with the object. There should be a value structure within the organization that makes clear the importance of supplying complete design specifications and emphasizes that, as in Solar Max, the consequences of even minor omissions can be serious.

But there is a more radical position on this issue that is surely worth exploring. It ought to be impossible to create or modify a design without doing it via a design capture system. Put slightly differently, there should be a design capture system so useful that no one would think of proceeding without it. The thought is utopian but not so far afield as it might seem. Existing VLSI design tools, for example, providing sufficiently powerful functionality that no major design would be done without them. Even their basic functions -- schematic capture and edit, design rule checking, simulation -- provide sufficient payback to make them worth the trouble.[1]

Existing tools also illustrate important limitations: they capture the final result, but not the rationales, not the design process. An effective system would be one that was useful from the earliest "sketch on the back on an envelope" stage, and that captured (and aided) every step and decision along the way. The result would be a record that included not only the final design, but its intended functionality, all rationales for the design choices, etc.

The technical problems in creating such a system include standard concerns about a good interface, such as ease of use and portability; paper is still hard to beat. But the issues go considerably deeper than that. Engineers find communication with each other possible in part because of a large shared vocabulary and base of experience. Communication with a design capture system should be based on similar knowledge; the identification and representation of that knowledge is a sizable research task.

The relevant vocabulary includes concepts about structure (shape, connectivity, etc.) and behavior (what the device should do). Both present interesting challenges. While connectivity is relatively straightforward, a compact and appropriate vocabulary for shape is not obvious. Behavior can sometimes be captured by equations or short segments of code, but descriptions in that form soon grow unwieldy and opaque. We need to develop a vocabulary for behavior capable of dealing with considerably more complex devices.

There is also the problem of unspoken assumptions. If design capture systems simply transcribe what is expressed literally, forcing every fact to be made explicit, the description task will always be overwhelming. We need to understand and accumulate the knowledge and design conventions of engineers so that the system can make the relevant inferences about what was intended, even if not expressed.

Designing for: Testability, Diagnosability, Analyzability, Comprehensibility, Transparency,...

We have argued that the complexity of the station and the novelty of the environment preclude an exhaustive a priori analysis of contingencies and require instead an ability to figure out what to do in the face of unanticipated events. We have suggested that this in turn is best facilitated by "knowing how things work," i.e., having a model of structure and behavior.

The complexity of the systems we design clearly has an impact on both how easy it will be to create such models and how easy it will be to reason with them once they exist. Since we are in fact designing the station (rather than trying to model a naturally occurring system), it is worth asking what can be done at the design state to facilitate model creation and model use.

Design for Testability Design for testability is one relatively well known approach in this category[7]. It acknowledges that newly manufactured devices have to be exhaustively tested to verify their correct operation before they are placed in service and suggests that we design in ways that facilitate this task. Substantial effort has been devoted to this in circuit design, with some success. Given the likely need for equipment maintenance and the difficulty of a house (station?) call by service technicians, it will be useful to design the station in such a way that basic diagnostic tests can easily be run on devices that may be malfunctioning. Where well known concepts like ensuring that signals are observable and controllable are likely to carry over easily, part of the research task here lies in extending techniques developed for simple digital circuits to deal with much larger subsystems.

Design for Diagnosability Designs for diagnosability is a less well understood task. Where testing involves methodically trying out all of the designed behaviors of the device, diagnosis is a process of reasoning from the observed symptoms of malfunction to identify the possibly faulty components. Diagnostic power is measured in part by discrimination ability: more powerful diagnostic reasoning techniques implicate fewer components. But some problems are inherently ambiguous -- a device may be designed in such a way that the observed symptoms must correctly implicate a large number of different components. Design for diagnosability would involve designing in a way that avoids this situation. Put more positively, it would mean designing in ways that seek to minimize the number of components implicated by a malfunction.

One very simple observation along this line can be made by considering the topology of the device: the only subcomponents that can be responsible for an observed symptom are those that are "causally connected" to it. In an electronic circuit, for example, the most obvious causal connections are provided by wires. More generally, there must be some sequence of physical interactions by which the error propagates from its source to the point where it is observed. The fewer such interactions, the fewer candidate subcomponents. Simply put, this argues for "sparse (modular) designs," i.e., those with relatively few interconnections.

Designs with uni-directional components (i.e., those that operate in a single direction and have distinct inputs and outputs, like logic gates and unlike resistors), also have smaller candidate sets. In devices with uni-directional components there is a single direction of causality, giving us a notion of "upstream" and "downstream" of the symptom. Only components that are upstream can be responsible for the symptom.

Diagnosis also involves probing, i.e., taking additional measurements inside the device, as well as generating and running tests designed to distinguish among possible candidate subcomponents. We might also examine design styles that facilitate both of these tasks.

<u>Designing for Analyzability, Comprehensibility, Transparency</u> Given our emphasis on being able to figure out what to do, perhaps the most fundamental thing to do early on is what might be called design for analyzability or comprehensibility. If we have to think about how the device works and reason through the possibly subtle effects of an unanticipated event, then let's at least make that easy to do. This may be little more than the traditional admonition to "keep it simple," here given the additional motivation of on-the-spot analysis and response.

Simplicity in design will aid in making that easy; it may present additional virtues as well. Simplicity often produces transparency, an important component in people's willingness to accept automated assistance with critical tasks. Simplicity will help achieve NASA's design goal of allowing crews to intervene at low levels in any station subsystem.

Finally, simplicity may also produce robustness by assisting in determining when a model is inappropriate. We argued above that the override decision is part of the model selection process and could be facilitated by making explicit the simplifying assumptions underlying each model. Those assumptions might not always be specified completely, at times it may be necessary to determine what they are. This is likely to be easier to determine if the model itself can be analyzed easily.

Robustness Requires Common Sense

Current expert systems are brittle in part because they lack common sense knowledge, that large collection of simple facts about the world that is shared across a culture. At the simplest it may include facts such as physical objects have mass and take up space, that two things cannot occupy the same space at the same time, or that objects that are unsupported will fall. In the absence of such an underpinning of world knowledge, the system must interpret its rules with complete literal mindedness and can do little in situations in which the rules "almost" apply.

Consider for example a rule in a medical diagnosis expert system specifying in part that "the patient is between 17 and 21 years old." Does the rule apply if the patient is 16 years 11 months old? How about 16 years 5.9 months? Our common sense knowledge of the world tells us that the human body doesn't change discontinuously, so the rule is probably still relevant. Compare this with a rule that says "If the postmark date is after April 15, then the tax return is late." Here we know, again from common sense knowledge, that there is in fact a discontinuity. Each of these chunks of common sense is simple enough and easily added to a system; the problem is finding and representing the vast collection of them necessary to support the kind of reasoning people do with so little effort.

For engineering problem solving of the sort relevant to our concerns here there is another layer of what we might call engineering common sense that includes such facts as, liquids are incompressible, all objects are affected by gravitational fields, but not all objects are affected by electromagnetic fields, electromagnetic fields can be shielded, and so forth. Engineers also know large numbers of simple facts about functionality, such as what a valve does, and why a door is like a valve.

The research task here is the identification, accumulation, organization, and interconnection of the vast numbers of simple facts that make up common sense (Lenat et al., 1986) and engineering common sense. Only with this body of knowledge will we be able to create systems that are more flexible and less literal minded.

What is the Source of Human Robustness?

Since robustness in problem solving is a common trait of experienced engineers, we ought to take the obvious step of examining that behavior and attempting to understand its origins. What is it that human experts do, what is it what they know, that allows them to recognize and deal with inadequate models? Why is it that human behavior seems to degrade gracefully as problems become more difficult, rather than precipitously, as is the case with our current programs? Part of the answer may lay in the number of and variety of models they can use, along with their body of common sense knowledge.

Multiple Models

Thus far our approach has focused on creating robustness by reasoning from detailed models. But how can we get robust behavior in situations where no effective model yet exists? One quite plausible reason for this would be incomplete information: even assuming we know all the limits of the models we have, selection of an appropriate one might depend on a fact about the system or environment that we simply don't have yet. In this section, we speculate on one possible approach to such problems.

One idea explored to some degree in the HEARSAY system (Erman, et al., 1980) for speech understanding involves the use of multiple knowledge sources, each dealing with a slightly different body of knowledge. Our imperfect knowledge about the task -- interpreting an utterance as a sentence -- means that none of the knowledge sources can be guaranteed to be correct. The basic insight here is to employ a group of cooperating experts, each with a different expertise, in the hope that their individual weaknesses are distinct (and hence will in some sense be mutually compensated) but their strengths will be mutually reinforcing.

A similar technique might be useful in engineering problem solving: lacking any one model believed to be appropriate, we might try using a collection of them that appear to be plausible and that have somewhat different conditions of applicability. Even given such a collection, of course, there remains the interesting and difficult problem of deciding how to combine their results when the outcomes are (as expected) not identical.

SUMMARY

We have argued that the complexity of the station and the novelty of space as an environment makes it impossible to predict and analyze all contingencies in advance. The hostility of the environment means the consequences of failure are substantial. In such situations, robustness and transparency become essential properties of the systems developed. Systems are robust to the extent that they can deal with events that have not been specifically anticipated and analyzed. They are transparent to the extent that they can make their reasoning comprehensible to an observer.

Given the inevitability of unanticipated events, robustness is best accomplished by "figuring out" what to do, rather than relying on a list of predetermined responses. But "figuring out," the sort of analysis and reasoning routinely done by engineers, can only be done if you "know how it works," i.e., have a model of the device. We thus believe that a key source of power in engineering reasoning is the collection of models engineers use, along with the approximations and abstractions that underlie the models. One major thrust of research then should be directed toward understanding the processes of model creation, selection, and simplification.

Given the serious consequences of working from incomplete information, a second major thrust should be devoted toward model and design capture. Existing systems for VLSI design are effective enough to make them essential tools, and hence effective in some aspects of design capture. We need to provide similar levels of tools for all varieties of design and need to understand how to capture design rationales as well as the final result of the design process.

Given the difficulty of the reasoning process even with complete information, we suggest turning the question around and asking what we can do at design time to make the reasoning task easier. We have speculated about what design for testability, diagnosability, and comprehensibility might mean, and suggest further exploration there as well.

Finally, it appears that additional leverage on the problem is available from examining human performance to determine the source of robustness in our own problem solving behavior, and from compiling the large body of common sense knowledge that seems to be a source of graceful degradation in human problem solving.

ACKNOWLEDGMENTS

Support for the preparation of this report came in part from a research grant from Digital Equipment Corporation, from the Defense Advanced Research Projects Agency of the Department of Defense, under Office of Naval Research contract N00014-84-K-0124, and from a research grant from the Wang Corporation. This paper benefitted significantly from comments on early drafts by Walter Hamscher, Brian Williams, Reid Simmons and Dan Weld.

NOTES

1. Rich and Waters, eds., Artificial Intelligence and Software Engineering, Morgan Kaufmann, 1986, is a recent survey of attempts to use AI approaches to this problem. It provides a historical overview and a wide-range view of the problem with extensive references. Also see the IEEE Transactions on Software Engineering.

2. Davis, Buchanan, Shortliffe, Production rules as a representation, Artificial Intelligence, February 1977, Pp. 15-45, provides an early overview of MYCIN, the first purely rule-based expert system. Waterman, A Guide to Expert Systems, Addison Wesley, 1986, is a recent text oriented toward commercial applications of the technology and provides a large set of examples and references.

3. Bobrow, ed., Qualitative Reasoning About Physical Systems, North-Holland, 1984, is the book version of the December 1984 issue of Artificial Intelligence, a special issue on that topic. Nine articles illustrate the variety of models and tasks attacked, including diagnosis, design verification, behavior prediction, etc.

4. Relatively little work addresses this topic directly. Patil, Szolovits, and Schwartz, Causal understanding of patient illness in medical diagnosis, Proc Seventh Intl Jt Conf on AI, Pp. 893-899, explores the combined use of three different kinds of models in diagnostic reasoning. Hobbs, Granularity, Proc Ninth Intl Jt Conf on

AI, Pp. 432-435 speculates on ways of producing coarser grained models from fine grained ones.

5. See the deKleer, Williams, and Forbus articles in Bobrow, op. cit.

6. See, for example, Gentner and Stevens, Mental Models, Lawrence Erlbaum, 1983.

7. Breuer, A methodology for the design of testable large-scale integrated circuits, Report SD-TR-85-33, January 1985, Space Division, Air Force Systems Command, provides a wide-ranging overview of different testability techniques.

REFERENCES

Eichoefer, Gerald

1985 MITRE Corp. Report. (Adapted from July 16 report)

Lenat, and Prakash, and Shephard

1986 Using common sense to overcome brittleness. Pp. 65-85 in AI Magazine. Winter.

Erman, and Hayes-Roth, and Lesser, and Reddy

1980 The hearsay-II speech understanding system: integrating knowledge to resolve uncertainty. Pp. 213-254 in Computing Surveys. June.

R. DAVIS

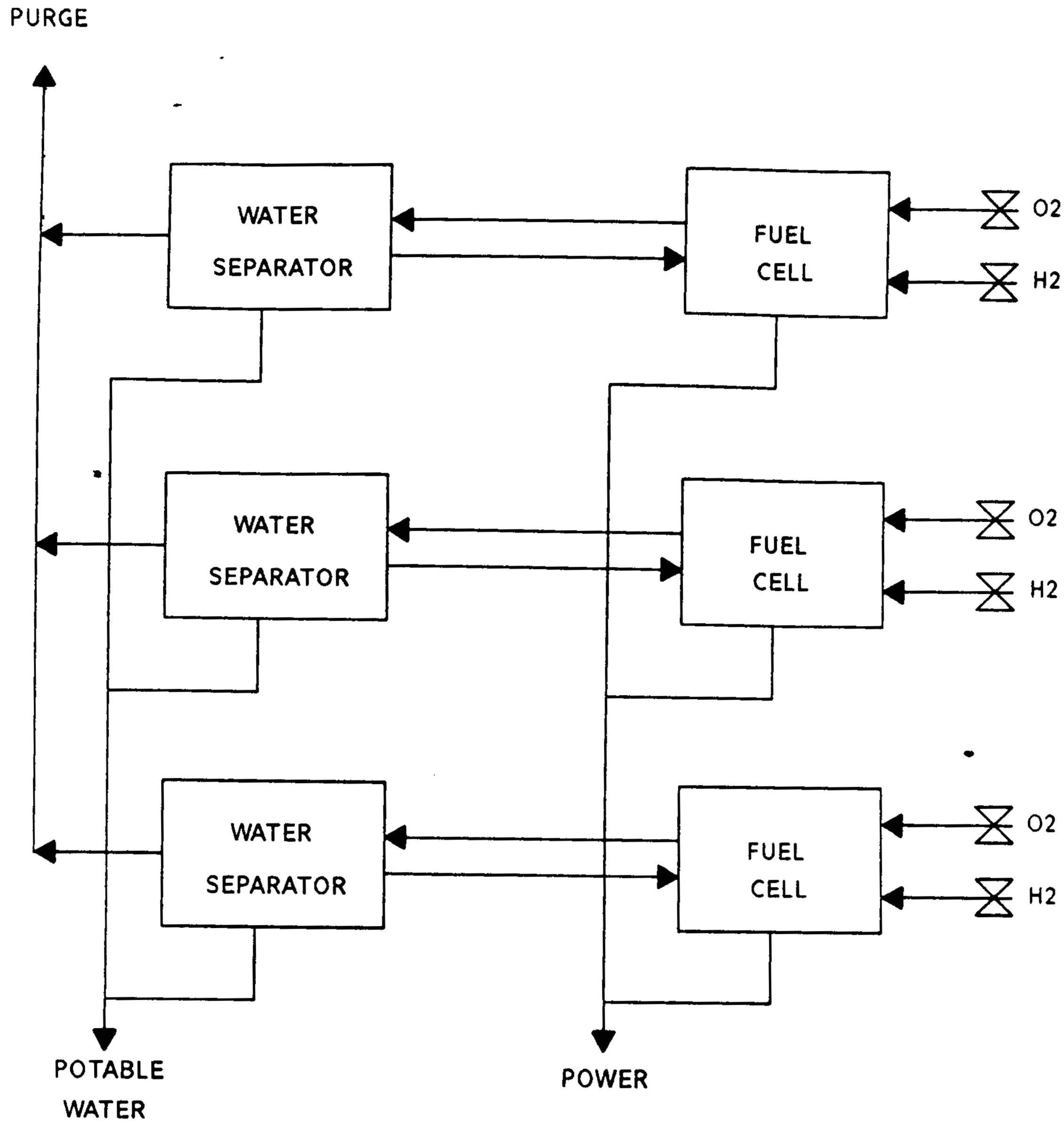

Figure 1: The fuel cell and water separation system.

R. DAVIS

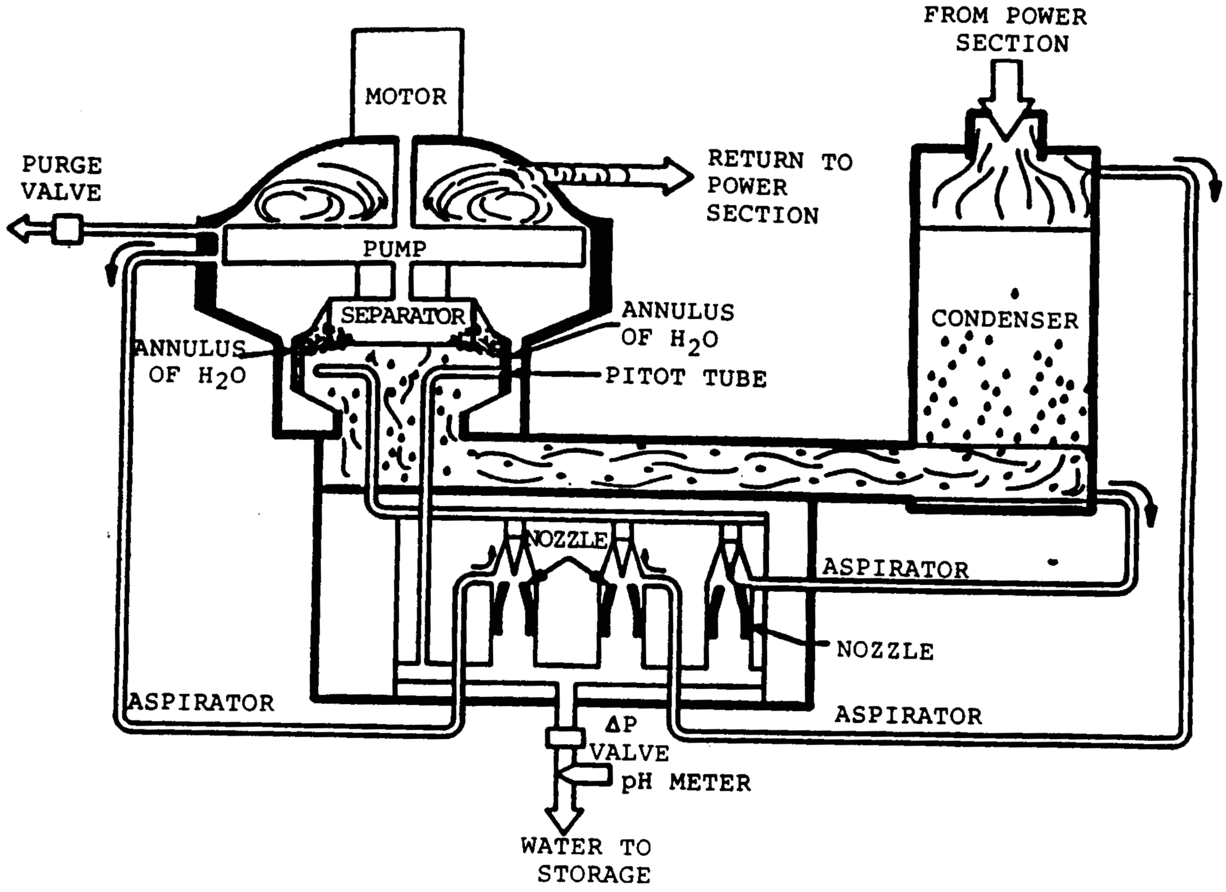

Figure 2: Details of the water separation unit.
(Adapted from MITRE Corp. report of 16 July 1985 by Gerald Eichhoefer.)

model and specifications.

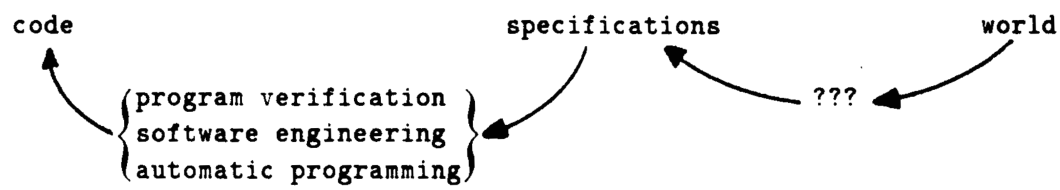

Figure 3