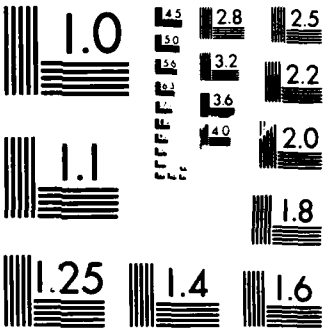
AD-A183 181 REPRESENTATION OF INFORMATION IN SOFTWARE AND DATA BASES(U) GEORGE MASON UNIV FAIRFAX VA D A BOEHM-DAVIS ET AL 30 JUN 87 TR-GMU-87-P01 1/1

UNCLASSIFIED N00014-85-K-0243 F/G 12/5 NL

FNL
8 87
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

George Mason University

TR-87-GMU-P01

REPRESENTATION OF INFORMATION IN SOFTWARE AND DATA BASES

DEBORAH A. BOEHM-DAVIS
ROBERT W. HOLT
MATTHEW KOLL

U

SECURITY CLASSIFICATION OF THIS PAGE

AD-A183 181

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION: U	1b RESTRICTIVE MARKINGS: --
2a SECURITY CLASSIFICATION AUTHORITY: --	3 DISTRIBUTION AVAILABILITY OF REPORT: Approved for Public Release; Distribution Unlimited
2b DECLASSIFICATION / DOWNGRADING SCHEDULE: --	
4 PERFORMING ORGANIZATION REPORT NUMBER(S): TR-GMU-87-P01	5 MONITORING ORGANIZATION REPORT NUMBER(S): --
6a NAME OF PERFORMING ORGANIZATION: George Mason University; 6b OFFICE SYMBOL (If applicable):	7a NAME OF MONITORING ORGANIZATION: Office of Naval Research
6c ADDRESS (City State and ZIP Code): Fairfax, Virginia 22030	7b ADDRESS (City State and ZIP Code): Arlington, Virginia 22217-5000
8a NAME OF FUNDING SPONSORING ORGANIZATION: Perceptual Science Program; 8b OFFICE SYMBOL (If applicable): Code 1142PS	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: Contract N00014-85-K-0243
8c ADDRESS (City State and ZIP Code): Arlington, Virginia 22217-5000	10 SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO: 61153N 42; PROJECT NO: RR 04209; TASK NO: RR0420901; WORK UNIT ACCESSION NO: 4424191

11 TITLE (Include Security Classification)

(U) Representation of Information in Software and Data Bases

12 PERSONAL AUTHOR(S)

Boehm-Davis, D.A., Holt, R.W., & Koll, M.

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year Month Day)	15 PAGE COUNT
Final	FROM 85JAN14 TO 87JUN30	87 JUNE 30	19

16 SUPPLEMENTARY NOTATION

--

17 COSATI CODES		
FIELD	GROUP	SUB-GROUP

18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)

Human computer interaction; organization of information; mental models; associational model. relational model.

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

A research program that examined the ways in which information can be represented in software and data bases is described. In the software production arena, structure was imposed on programs through the application of program design methodologies in the design phase. Modification performance and mental models formed during the experiment were examined as a function of design methodology for student and professional programmers. The data suggested that program structure had a much stronger influence on the student programmers. Further, the data showed that where program design led to different mental models, it also led to differences in modification performance. In the data base arena, the focus of the research was on the impact of structuring the information contained in the data base on the retrievability of information from that data base. Most existing data bases are organized around one of three data structures: hierarchical, network, and

20 DISTRIBUTION AVAILABILITY OF ABSTRACT: ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION: U
22a NAME OF RESPONSIBLE INDIVIDUAL: J.J. O'Hare	22b TELEPHONE (Include Area Code): (202) 696-4502; 22c OFFICE SYMBOL: Code 1142 PS

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

#19. Continued

relational. Each of these formats imposes certain restrictions on the way in which information is represented and on the way in which a user can add to, retrieve, or change the information contained in the data base. This research tested the hypothesis that the process of retrieving information from a data base is aided by tailoring the format of the information in the data base to the format of the information to be retrieved. The data showed that performance, as measured by both speed and accuracy, was in fact best when the format of the data base matched the type of information being retrieved. Further, the results showed that performance was influenced by prior exposure to a particular information presentation format. Overall, the data from both studies suggest the important role that the underlying cognitive representation plays in determining performance.

TR-GMU-87-P01

REPRESENTATION OF INFORMATION IN SOFTWARE AND DATA BASES

Final Report

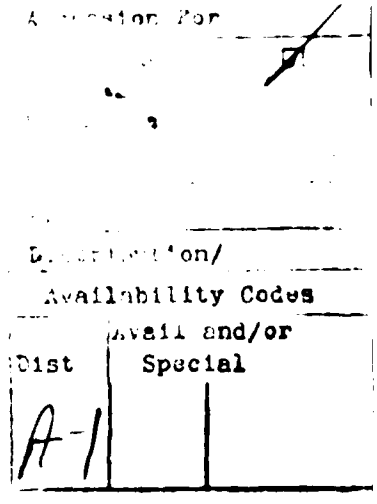
Deborah A. Boehm-Davis, Robert W. Holt, and Matthew Koll

George Mason University
4400 University Drive
Fairfax, Virginia 22030

Submitted to:

Office of Naval Research
Perceptual Science Program
Arlington, Virginia 22217

Contract: N00014-85-K-0243
Work Unit: NR-4424191



June 1987

TABLE OF CONTENTS

INTRODUCTION

For the past few years (14 JAN 85 - 30 JUN 1987), we have been concerned with the ways in which information can be represented in software and in data bases. Specifically, we have been concerned with the impact that different ways of structuring information has on comprehension in these two arenas.

In the software production arena, structure was imposed on programs through the application of program design methodologies in the design phase. Program design methodologies are techniques that provide strategies to programmers for structuring solutions, with the goal of improving the final program design. One fundamental difference among these methodologies is the criterion used to decompose the problem into smaller units. The methodologies basically vary in the way and extent to which modularization of the code is accomplished. On one end of this dimension is in-line code, where all of the procedures are contained in the main routine of the program. On the other end of the dimension are techniques which attempt to hide data structure implementation details and/or algorithm behavior from the main process as the basis for structuring the programs (such as object-oriented design or Parnas' information-hiding technique). Falling in between these two are techniques which rely on functional structure alone as the basis for structuring the problem, such as functional decomposition, or top-down design.

Using this dimension to classify methodologies, we generated programs decomposed in each of these ways, and evaluated the effects of these decompositions on the modifiability of the resulting code.

In the data base arena, the focus of the research was on the impact of structuring the information contained in a data base on the retrievability of information from that data base. While more and more information is being stored in large data bases, the question of how best to format that information has not been studied experimentally.

Most existing data bases are organized around one of three data structures: hierarchical, network, and relational. Each of these formats imposes certain restrictions on the way in which information is represented and on the way in which a user can add to, retrieve, or change the information contained in the data base. Specifically, data bases are used to represent information about objects, qualities of objects, and relationships among those objects. The three approaches differ in the way they permit users to access and manipulate the information in the data base. This, in turn, affects the format in which the information can be presented to the user and can affect the ease with which a user can come to understand and use the information in the data base to answer questions.

In a hierarchical system, objects are represented by their relative position to other objects in the data base. Thus, any given object can only be fully understood when seen in context; that is, between its superior and subordinate objects. Networks provide a more general structure than hierarchies since networks allow objects to have multiple superiors and subordinates. In this approach, relationships are represented by pointers which direct the user to information related to each object. In both of these approaches, access to the objects occurs by following information through the structure, so information must be retrieved following a sequenced path. The relational model is based on the matematical theory of relations. In these models, objects and relationships are not differentiated, rather, information is represented as sets of related objects, and their relationships, eliminating the need for separate operations to manipulate objects and relationships.

While the impact of these approaches on the physical representation of the data base and on the operations required to manipulate information stored in the data base is fairly well-known and understood, the impact on user understanding of the system has not been considered or investigated. Psychological research on learning and memory suggests that performance may be affected by the similarity between people's internal representations of information and the

external presentation format of information. Our reseach tests the hypothesis that the organization of the data base which best matches the organizational style of the user and the informational requirements of the questions will lead to the best performance.

RESEARCH

In this research program, we have completed two experiments. The first examined the impact of structure on software maintainability while the second examined the impact of structure on data base use.

Software Research

In this experiment (Boehm-Davis, Holt, Schultz & Stanley, 1986), professional and student programmers were provided with each of three problems and asked to make modifications to the code.

Programs were created using each of three design approaches: straight serial structure (in-line code), structure emphasizing functional units of the program (functional decomposition), and structure emphasizing larger object-oriented modules of the program (object-oriented). These program structures were used to write programs for each of three problems. The problems involved a real-time data base system (host-at-sea system), a database with files (military address system), and a program constructing large linked-list data structures (student transactions system). Ease of maintenance for these programs was examined by presenting programmers with either simple or complex modifications to be made to the code and measuring the amount of time required to make those modifications.

The object-oriented modularization was predicted to be most compatible with the users' internal representations of the software problems posed and thus produce the best overall performance. A further expectation was that increasing structure would increase ease of modifiability. Thus, the in-line code should produce the worst performance since it does not have any structure. Both functional

decomposition and object-oriented design were predicted to lead to superior performance.

These predictions were also consistent with the demands placed on the programmers. The in-line code does not provide any structure to the program; therefore, maintenance programmers will need to build a cognitive structure as they read through and try to comprehend the program. The functional decomposition will outline modules for each function and hence provide a starting structure to programmers; however, the programmers will be required to redefine and integrate these functions into the real-world specifications for the problem, which will require some additional time for program comprehension. The object-oriented code provides one module for each real-world object, or design decision, in the system. The data and functions associated with that object are already integrated in each module. This representation scheme should allow for direct translation to the specifications, and thus, should lead to maximum performance. However, a there was a possibility that the integration of both data and functions within a module would lead to enough increased complexity to offset the benefits that should accrue from increased structure. These hypotheses were tested in this research.

Data Base Research

In this experiment (Boehm-Davis, Holt, Koll, Yastrop, & Peters, in press), the effects of three different data base formats on the information retrieval performance of users was examined: graphical, tabular, and verbal forms of two data base domains (airline and thesaurus). Questions were formulated that required users to search through the data base to determine the correct response. Questions were designed to emphasize three types of information: graphical, tabular, and verbal. Performance was examined by measuring the amount of time required to answer the questions and by the proportion of questions answered correctly. In addition, the participants filled out a questionnaire which asked for demographic information and subjective reactions to the different data base formats. The prediction was that

performance would be maximized, both in terms of speed and accuracy, when the physical format of the data base matched the type of information that the question emphasized.

CONCLUSIONS

Software Research

The data provided by this research allowed us to make several interesting observations about the role that structure plays in determining modification performance. They also provide insights into the similarities and differences between student and professional programmers.

The completion-time data suggest that modification performance is influenced by an interaction between the structure of the problem and the type of problem presented. This interaction was only statistically significant for the student programmer group, but the pattern of results was similar for the two groups of programmers. The major differences between the two groups were in solution speed and in the effect of the object-oriented structure on the difficulty of the host-at-sea buoy system. The professional programmers modified the military address and student-transaction list systems faster than the student programmers, but modified the host-at-sea buoy system in approximately the same amount of time as the student programmers. While the object-oriented version of the host-at-sea buoy system required significantly more time to modify than the other versions of that problem for both groups, the effect was much more pronounced for the student programmers.

For both groups, substantially less completion time was observed for the simple modifications. This difference between simple and complex modifications was also reflected in significant differences in the number of editor transactions for both groups of programmers and for the number of editor sessions, chunks, and relations recalled for the professional programmers. This confirms that the complex

modifications were indeed more difficult than the simple modifications. The complex modifications required changes in several locations of the code while our simple modifications required changes in only one location in the code.

For the student programmers, ease of modification also interacted with problem structure. This interaction revealed that for the simple modifications, problem structure did not influence ease of modification. For the complex modifications, the functionally decomposed code was easiest to modify, the in-line code was slightly more difficult to modify, and the object-oriented code was most difficult to modify. This suggests that structure, per se, is not as important as the particular type of structure.

For both groups of programmers, there was a significant difference in the completion times and number of editor transactions required to modify the three systems. In all cases, the military-address system was the easiest, while the student-transaction list and host-at-sea buoy systems were roughly equal in difficulty, and more difficult than the military-address system.

We again find significant differences between student and professional programmers when we examine the impact of program design on the cognitive representations formed. The mental models of the professionals were primarily affected by the difficulty of the assigned modification, while the mental models of the students were primarily affected by the structure and content of the programs. This may indicate that the professionals are better at getting at the kernel of the task to be performed and are less influenced by "peripheral" aspects of the programming task such as the surface structure of the program or the content of the program.

Overall, the data suggest that problem structure, problem type, and ease of modification affect performance. The data suggest that the pattern of results is similar for professional and student programmers; however, the extent of each condition depends on the group to which the

programmer belongs. That is, professionals are less influenced by different program structures and contents than are students. This is not surprising given the profiles of the two groups. Students averaged 0.2 year of experience (with a range of 0 - 1 year) while professionals averaged 3.5 years (with a range of 1.5 - 12 years). The professionals were familiar with slightly more programming languages and operating systems while both groups were familiar with approximately the same number of program design methodologies. Both groups of programmers reported that they relied on the same pieces of documentation, which suggests qualitative similarities in their strategies for solving problems.

Increasing program structure did not lead to a significant increase in ease of modifiability. The functionally-decomposed code was the easiest to modify, while the in-line code was slightly more difficult, and the object-oriented code was even more difficult. An examination of the reports from the participants after they had completed the experiment suggested a trade-off between program structure and ease of modifiability. The object-oriented code was the most modularized so that this program structure required more passing of information from module to module. It would appear that the cognitive demands required to keep track of information is greater than its reduction by the increased modularity.

The notion of cognitive demand as a determinant of performance is supported by the relationship between cognitive representations and modification performance. In this research, we found that differences in cognitive representation seemed to be mirrored in differences in modification performance. That is, no differences were found between the cognitive representations of professional programmers as a function of program design and there were similarly no performance differences. For student programmers, differences in both cognitive representations and modification performance were found as a function of program design and content.

These data suggest that when a program design leads to a different cognitive representation, it will also lead to performance differenes in modifying that program. The data also show that professional programmers were less sensitive to design changes. This suggests that a relevant part of student programmer training might be training in concentrating on the essential data structures and processes in the program rather than on the superficial form of the program. This is a quite different emphasis than emphasizing a particular "true" route to constructing a program, such as those suggested by advocates of program design methodologies.

Data Base Research

The results from the second study provided a number of interesting insights into the impact of data base format on the ability to retrieve information from that data base. Most importantly, the data supported the hypothesis that the information retrieval process is aided by tailoring the format of the information in the data base to the format of the information to be retrieved. Performance, as measured both by speed time and by accuracy, was best when the format of the data base matched the type of information to be retrieved.

In addition, a number of other observations about the effects of structure could be drawn from the results. The participants were able to retrieve information significantly more quickly and with fewer errors when using the airline data base than when using the thesaurus data base. The main effect of data-base format, collapsed across the two data-base domains, suggested that questions were significantly more quickly answered using the graphical format than with either the verbal or tabular formats. The main effect for question type suggested that tabular questions were answered significantly more quickly than verbal questions, and these were both answered significantly more quickly than graphical questions. In terms of the percent correct, the main effect of data base format suggests that participants using the verbal format answered a significantly higher proportion of questions correctly than participants using either the tabular or graphical formats. Also in

terms of percent correct, the tabular and verbal questions had a significantly higher overall proportion correct than the graphical questions.

After being shown each of the three data base formats, the participants were asked to rate how much they liked each format and how easy it was to answer questions from each format. Those participants who worked with either the verbal or tabular formats expressed equal preferences for the verbal and tabular formats, and they significantly preferred both of these to the graphical format. Participants who worked with the graphical data base format also expressed equal preferences for the verbal and tabular formats. However, these participants significantly preferred the graphical format over the other two formats. This suggests that people have a tendency to prefer the format they worked with during the experiment.

This influence did not extend to a more general measure of format preference. Before being shown all three formats of the data base, the participants were asked which information format they generally prefer. The format of the data base they worked with did not have a significant influence on this more general preference rating.

The data indicate that while pre-existing preferences for information display format play a role in performance, experience with an alternate form of representation can lead to changes in preference. This suggests that rapid prototyping could be an important tool in specifying system requirements. A customer unfamiliar with a potentially useful information display format could gain experience with the new format through an early prototype and then choose a more appropriate format for the final system.

General Conclusions

The results from both studies suggest that structure plays an important role in determining performance when people try to understand either software or data bases. In software mofification, problem

structure, type of problem, and type of modification affect modification performance for student and professional programmers. Further, the data suggested that when a program design leads to a different cognitive representation, it will also lead to performance differenes in modifying that program.

The data also showed that professional programmers were less sensitive to design changes. This suggests that a relevant part of student programmer training might be training in concentrating on the essential data structures and processes in the program rather than on the superficial form of the program. This is a quite different emphasis than emphasizing a particular "true" route to constructing a program, such as those suggested by advocates of program design methodologies. The importance of tailoring the format of the information in a data base to the format of the information to be retrieved was shown. Results from this latter work showed that both performance and preference are influenced by prior exposure to a particular information presentation format. Overall, the data from both studies suggest the important role that the underlying cognitive representation plays in determining performance.

SCIENTIFIC PERSONNEL WHO WORKED ON PROGRAM

Deborah A. Boehm-Davis

Robert Cooper

Jane Holloway

Robert W. Holt

Matthew Koll

Robert Peters

Alan C. Schultz

Philip W. Stanley

Gloria Yastrop

TECHNICAL REPORTS

Boehm-Davis, D. A., Holt, R. W., Schultz, A. C., & Stanley, P. (1986, May) The role of program structure in software maintenance. (Technical Report TR-GMU-86-P01). Fairfax, VA: George Mason University (AD A168775).

ARCHIVAL PUBLICATIONS

Boehm-Davis, D. A., & Ross, L. S. (1985) Program design methodologies: Structuring the software development process. In Proceedings of the IEEE Man, Systems, and Cybernetics Annual Meeting.

Boehm-Davis, D. A. (1985) Methodology and problem representation in programming. In Proceedings of the Human Factors Society 29th Annual Meeting (p. 30) Santa Monica CA: Human Factors Society.

Boehm-Davis, D. A. (1987, May) Documentation in the product development cycle. In Proceedings of INTERFACE '87 Rochester N.Y.

Boehm-Davis, D. A., Holt, R. W., Koll, M., Yastrop, G., and Peters, R. (in press) The effects of different data base formats on information retrieval. In Proceedings of the Human Factors Society 31st Annual Meeting Santa Monica CA Human Factors Society.

Boehm-Davis, D. A. (in press) Software [illegible] In M. Helander (Ed.) Handbook of Human Computer Interaction Amsterdam North Holland

Curtis, B., Sheppard, S. B., Baily, E. P., Baily, J., & Boehm-Davis, D. A. (in press) Experimental evaluation of software documentation formats. Journal of Systems and Software

Holt, R. W., Boehm-Davis, D. A., Schultz, A. (in press) Mental representations of programs for student and professional programmers. In D. Gilmore & B. Shackel (Eds.) Empirical Studies of Programmers II

TECHNICAL REPORTS DISTRIBUTION LIST

OSD
Dr. Earl Alluisi
Office of the Deputy Under Secretary
of Defense
OUSDRE (E&LS)
Pentagon, Room 3D129
Washington, D.C. 20301

DEPARTMENT OF THE NAVY

Mr. Philip Andrews
Naval Sea Systems Command
NAVSEA
Washington, D.C. 20362

Dr. L. Chmura
Computer Sciences & Systems
Code 5592
Naval Research Laboratory
Washington, D.C. 203650

Dr. Stanley Collyer
Office of Naval Technology
Code 222
800 North Quincy Street
Arlington, VA 22217-5000

Dean of the Academic Departments
U.S. Naval Academy
Annapolis, MD. 21402

Director Technical Information Division
Code 2627
Naval Research Laboratory
Washington, D.C. 20375-5000

Dr. Robert A. Fleming
Human Factors Support Group
Naval Personnel Research &
Development Center
1411 South Fern Street
Arlington, VA. 22202-2896

Dr. Eugene E. Gloye
ONR Detachment
1030 East Green Street
Pasadena, CA 91106-2485

Mr. Jeff Grossman
Human Factors Laboratory, Code 71
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr. Charles Holland
Office of Naval Research
Code 1133
800 N. Quincy Street
Arlington, VA 22217-5000

Human Factors Branch
Code 3152
Naval Weapons Center
China Lake, VA 93555

Human Factors Department
Code N-71
Naval Training Systems Center
Orlando, FL 32813

Human Factors Engineering
Code 441
Naval Ocean Systems Center
San Diego, CA 92152

CDR Thomas Jones
Code 125
Office of Naval Research
800 North Quincy St.
Arlington, VA 22217-5000

Dr. Michael Letsky
Office of the Chief of Naval
Operations OP-01B7
Washington, D.C. 20350

Lt. Dennis McBride
Human Factors Branch
Pacific Missile Test Center
Point Mugu, CA 93042

LCDR Thomas Mitchell
Code 55
Naval Postgraduate School
Monterey, CA 93940

Dr. Jerry Muckler
Human Factors Department
Naval Submarine Medical
Research Lab
Naval Submarine Base
Groton, CT 06349-5900

CAPT W. Moroney
Naval Air Development Center
Code 602
Warminster, PA 18974

Dr. A.F. Norcio
Compter Sciences & Systems
Code 5592
Naval Research Laboratory
Washington, D.C. 20375-5000

CDR James Offutt
Office of the Secretary of Defense
Strategic Defense Initiative Organization
Washington, D.C. 20301-5000

Perceptual Science Program
Office of Naval Research
Code 1142PS
800 North Quincy Street
Arlington, VA 22217-5000 (3 copies)

Dr. Randall P. Schumaker
NRL A.I. Center
Code 7510
Naval Research Laboratory
Washington, D.C. 20375-5000

LCDR T. Singer
Human Factors Engineering Division
Naval Air Development Center
Warminster, PA 18974

Mr. James Smith
Code 121
Office of Naval Research
800 North Quincy Street
Arlington, Va. 22217-5000

Special Assistant for Marine
Corps Matters
Code OOMC
Office of Naval Research
800 North North Quincy Street
Arlington, VA 22217-5000

DEPARTMENT OF THE ARMY

Director, Organizations and Systems
Research Laboratory
U.S. Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333-5600

Dr. Edgar M. Johnson
Technical Director
U.S. Army Research Institute
Alexandria, VA 22333-5600

Dr. Milton S. Katz
Director, Basic Research
Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333-5600

Technical Director
U.S. Army Human Engineering Laboratory
Aberdeen, Proving Ground, MD 21005

DEPARTMENT OF THE AIR FORCE

Dr. Kenneth R. Boff
AF AMRL/HE
Wright-Patterson AFB, OH 45433

OTHER GOVERNMENT AGENCIES

Defense Technical Information
Center
Cameron Station, Bldg. 5
Alexandria, VA 22314 (2 copies)

Dr. Clinton Kelly
Defense Advanced Research
Projects Agency
1400 Wilson Blvd
Arlington, VA 22209

Dr. Alan Leshner
Division of Behavioral and Neural Sciences
National Science Foundation
1800 G. Street, N.W.
Washington, D.C. 20550

Dr. M.C. Montemerlo
Information Sciences &
Human Factors Code RC
NASA HQS
Washington, D.C. 20546

OTHER ORGANIZATIONS

Dr. H. Van Cott
NAS-National Research Council
(COHF)
2102 Constitution Avenue, N.W.
Washington, D.C. 20418

Dr. Allen Newell
Department of Computer Science
Carnegie-Mellon University
Pittsburgh, PA 15213

Dr. Jesse Orlansky
Institute for Defense Analyses
1801 N. Beauregard Street
Alexandria, VA 22311

Dr. Richard Pew
Bolt Beranek & Newman, Inc.
10 Moulton Street
Cambridge, MA 02238

Dr. Scott Robertson
Department of Psychology
Rutgers University
Busch Campus
New Brunswick, N.J. 08903

Ms. Bonnie John
Department of Psychology
Carnegie-Mellon University
Pittsburgh, PA 15213

Dr. Thomas G. Moher
Department of Electrical Engrg. & Computer Science
Univ. of Illinois at Chicago
P.O. Box 4348
Chicago, IL 60680

Dr. Jay Elkerton
University of Michigan
Department of Industrial & Operations Engineering
Center for Ergonomics
1205 Beal Avenue
Ann Arbor, MI 48109

END

8-87

DTIC