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

MANEUVERING BOARD TRAINING SYSTEM: ANALYSIS AND REDESIGN

AD A139496

DTIC
ELECTE
MAR 2 8 1984

E

DTIC FILE COPY

**NAVY PERSONNEL RESEARCH
AND
DEVELOPMENT CENTER
San Diego, California 92152**

84 03 28 002

MANEUVERING BOARD TRAINING SYSTEM: ANALYSIS AND REDESIGN

Edwin Hutchins
Timothy McCandless
Glenn Woodworth
Bancroft Dutton

Reviewed by
Richard C. Sorenson

Approved by
J. W. Tweeddale

Released by
J. W. Renard
Captain, U.S. Navy
Commanding Officer

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NPRDC TR 84-19	2. GOVT ACCESSION NO. AD-A137496	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) MANEUVERING BOARD TRAINING SYSTEM: ANALYSIS AND REDESIGN		5. TYPE OF REPORT & PERIOD COVERED Technical Report
		6. PERFORMING ORG. REPORT NUMBER 41-83-05
7. AUTHOR(s) Edwin Hutchins Glenn Woodworth Timothy McCandless Bancroft Dutton		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Navy Personnel Research and Development Center San Diego, California 92152		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS ZF63-522-801-013-03.03
11. CONTROLLING OFFICE NAME AND ADDRESS Navy Personnel Research and Development Center San Diego, California 92152		12. REPORT DATE February 1984
		13. NUMBER OF PAGES 28
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Maneuvering board
Relative motion

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

Ship handling in traffic is a difficult task and the consequences of error can be grave. The effort described herein explores the application of microcomputer technology to the problem of training procedures used to determine appropriate ship maneuvers. A maneuvering board training system was developed, implemented, and installed in a fleet training school. Instructors found the system useful but did not use it regularly because of the complexity of the user interface. Weaknesses in the

interface were identified, and the interface was redesigned to make it more usable. The redesign process highlights the tradeoffs that must be made between the system's conceptual coverage and the complexity of the user interface.

FOREWORD

This report was prepared under exploratory development task ZF63-522-801-013 (Training for Decision Making and Problem Solving). The overall goals of the task are to explore the application of microcomputer technology for training declarative and procedural knowledge and to develop standalone computer-based instruction. The current effort explores the application of microcomputer technology to the problem of training procedures used to determine appropriate ship maneuvers.

This report is intended for use by training system designers and for training commands that include the maneuvering board course in their curricula.

J. W. RENARD
Captain, U.S. Navy
Commanding Officer

JAMES W. TWEEDDALE
Technical Director

Accession For		
NTIS GRA&I	☒	
DTIC TAB	☐	
Unannounced	☐	
Justification		
By		
Distribution/		
Availability Codes		
Dist	Avail and/or Special	
A-1		

v

SUMMARY

Problem

A microcomputer-based maneuvering board training system to teach students to solve relative motion problems was developed, implemented, and installed in a "C" school at the Fleet Combat Training Center, Pacific. Instructors felt that the program was of value in communicating the concepts of relative motion to students but did not use it regularly because of the complexity of the user interface.

Objective

The objective of the effort described herein was to identify the problems in the system and redesign the user interface so that the system's instructional value would be more directly available to students and instructors.

Approach

Problems were identified by (1) interviews with instructors and students and (2) having two novices attempt to learn how to use the system. In the redesign process, the principles of cognitive science and man-machine interface design were applied to identified problems. This portion of the research was conducted in collaboration with the Cognitive Science Laboratory of the University of California at San Diego.

Results

The system was redesigned and implemented in accordance with principles of man-machine interaction studies.

Conclusions

The redesigned system represents a significant improvement over its predecessor. It has been installed in the Operations Specialist "A" school of the Fleet Combat Training Center at Dam Neck, Virginia, and is now in regular use in that school. The redesign process itself highlights the importance of the tradeoffs that must be made between the conceptual coverage provided by the training system and the complexity of the user interface.

CONTENTS

LIST OF FIGURES

INTRODUCTION

Problem and Background

Ship handling in traffic is a difficult task and the consequences of error can be grave. Because of the time required to execute maneuvers in large ships, accurate predictions of the effects of various maneuvers must be continuously available. Aboard most surface ships, these predictions are based on analyses performed on a plotting sheet called the maneuvering board. Because the maneuvering board is used to analyse the motion of objects relative to each other, an operator must understand the principles of relative motion, a conceptually difficult task. Fleet training commands report high failure rates in courses that require this knowledge.

The Navy Personnel Research and Development Center (NAVPERSRANDCEN) is conducting a program exploring the use of microprocessor technology, which is especially suitable for training operators to use the maneuvering board. It is very difficult to represent the changing dynamic relations among ships and the multiple points of view required to understand relative motion in static instructional media such as print or chalkboard. Yet they can be clearly represented by an interactive graphics display driven by a simulation model of ship movements.

Systems being developed under the microprocessor technology program are implemented at a user activity, operated by those who will use the final system, refined in accordance with strengths and users noted by operators, sent back into the world of users, and so on. This in-situ development plan ensures that researchers do not end up creating a system that meets their theoretical expectations but not the needs of the users.

A microcomputer-based maneuvering board training system was developed as part of the microprocessor technology program (Hutchins & McCandless, 1982). After several design/test cycles, version 3.0 of this system was installed at the Fleet Combat Training Center Pacific in San Diego, California in the fall of 1981. This system used interactive graphics to provide training in the conceptual basis of relative motion problems. It presented a simultaneous display of relative and geographic depictions of the motions of ships (see Figure 1). The relative depiction showed how the motion of ships would appear on the radar screen of one of the ships involved; and the geographic depiction, how the motion of ships would appear to an observer on a stationary platform high above the ocean. The student could generate any desired ship interaction scenario by entering data about the participating ships (up to 26) into a large menu-like region at the bottom of the display screen, and the system would automatically incorporate the entered information into a mathematical simulation model of the scenario described. The scenario could then be run forward or backward in time. While the simulation ran, the relative and geographic displays were automatically updated to show the tracks of the ships. In addition, the system provided a very general but complicated set of commands for the control of such aspects of the display as scale and the choice of automatic or manual recentering of ships on the geographic plot.

In the 3.0 version, the ship interaction scenario facility provided the user with a great deal of latitude in the construction of the scenarios and the control of the display. The relative display assumes that the observations depicted are made from one of the ships in the scenario, referred to as "ownship." The construction of a minimal scenario requires that the parameters describing the motion and positions of ownship and at least one other ship be specified. Ownship is characterized by a course and speed and an initial position (x,y) on the geographic display. Other ships in the scenario are given labels (a letter

1

designation A-Z), so that they can be distinguished from each other, and also have courses, speeds, and an initial position relative to ownship (given in bearing and range from ownship). As can be seen in Figure 1, the motion parameters of only one contact ship are displayed at any one time. If the scenario includes more than one contact ship, the motion parameters of any one can be displayed as the user desires, but the parameters of all the others will not be displayed.

Scale : 1:1	Time : 20:00		Scale : 1:1	
Motion : stopped	ToTime : >20:00		GeoGraph: ON	
Course : 0	Ship : none		Course : 0	
Ref: Speed : 0.0			Speed : 0.0	
X : 0.0	DRM : 90		Bearing : 90	
Y : 0.0	RelSpeed: 0.0		Range : 0.0	

ARROWS, SPACE to change,)-Frwd, (-Rev, N(ew, T(rans, A(uto, L(esson, Q(uit'
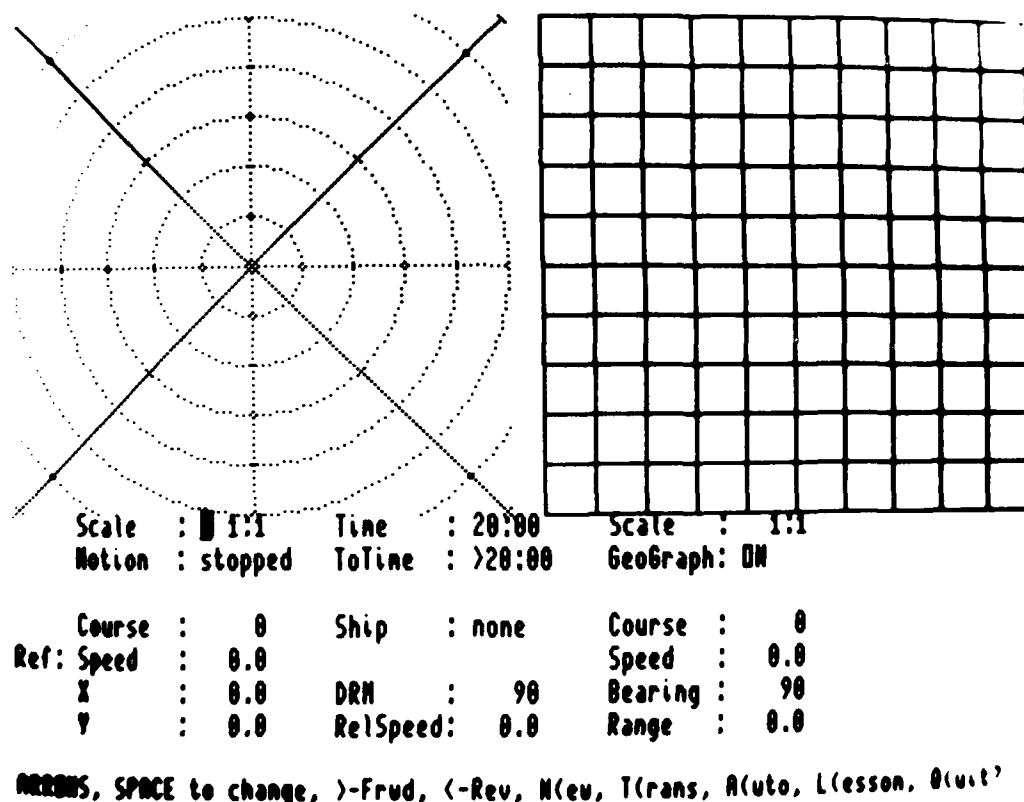
Figure 1. Display of maneuvering board training system (version 3.0).

Note. The relative plot is on the left and the geographic plot, on the right. The upper two lines of text contain options for controlling the simulation and display; the next four lines, for setting and displaying the motion parameters of ownship and other ships; and the bottom line, for controlling the simulation and the display.

To make the facility more general in the sense of being able to represent and depict any conceivable ship interaction scenario (e.g., from a student's workbook or a description of a historical event), the user was provided with control over the display. For example, the scale of both displays defaulted to 1000 yards per grid division but could be set to any

scale up to 5000 yards per division by the user. Such a scale choice might be required to accommodate a problem with a very large initial range. Regardless of the scale chosen, if ownship has a nonzero speed, it will eventually steam off the edge of the geographic plot. So that the interaction display would still be useful when this happens, methods were provided to get ownship back onto the geographic plot. Again, to give the student the ability to model any scenario, he could set a beginning time for the scenario and step it forward or backward one tick at a time or jump it forward or backward to any specified time.

The attempt to make the facility general led to a number of problems with the interface. For example, although the 3.0 version allowed scenarios with up to 26 ships to be constructed, a scenario involving even a few ships would never be interpretable, even if a student had the stamina to construct one. A representation of the simultaneous relative motions of several ships simply contains too much information to be understood, even by an expert. Relative motion is so complex that understanding a single ship interaction requires a lot of effort. Even in the real world, multiple ship interactions are understood as a set of pairwise interactions.

Giving the user control over such display parameters as scale, initial time, and repositioning on the geographic plot consumes valuable screen space on a display with limited resolution. Control of these parameters is not directly related to the principles of relative motion and competes for the student's limited mental resources (Norman & Bobrow, 1975). Further, the appropriate use of any of these assumes a level of understanding of relative motion that is probably beyond most beginning students. Even worse, the behavior of some of them (e.g., scale and repositioning) could be a source of confusion.

In addition to the problems caused by attempting to have the system do too much, there were difficulties in the mechanics of the interaction of the student with the system concerning the actions required of the user to enter data, monitor the system's behavior, and control the display.

Purpose

The purpose of the work described herein was to identify the problems in the system related to the use interface and, based on results, to redesign the user interface so that the system would have more instructional value to the students and instructors.

APPROACH

Interface problems were identified by (1) conducting interviews with students and instructors and (2) having two novices use the 3.0 version of the system and note errors made and aspects of the system that were difficult to use.[1] Problems identified were analyzed and eliminated to the extent possible in collaboration with the Human-Machine Interface group of the Cognitive Science Laboratory, University of California, San Diego (UCSD).

---

[1]Woodworth, G., & Dutton, B. In-house report on the maneuvering board system, undated.

RESULTS

Identified Interface Problems

The most important problems identified in the user interface to the version 3.0 system are listed below. Brief examples are provided where appropriate.

1. <u>Inconsistent data entry protocol</u>. To enter information about a ship, one gives the ship a letter designation (A-Z) and then specifies ship parameters (e.g., initial relative position, course, speed, etc.). The actions required to enter different types of information are not uniform. For example, to enter a ship designation, the user positions the cursor and types a letter character that is accepted as soon as it is typed. To enter a ship parameter, the user positions the cursor, types a space to clear the previous entry, types the new entry, and then types either "space" or "return" to accept the new entry. To achieve goals through interaction with the system, the user must have a model of the mapping of actions onto effects (Black & Sebrechts, 1981; Norman, 1983; Moran, 1981a). When data entry actions are inconsistent, the user's memory load is increased because of the many different action sequences required to accomplish the same high-level goal. In addition, inconsistent data entry actions present additional opportunities for error and all of the problems that are attendant to committing errors. The solution is to have a single consistent data entry action wherever possible.

2. <u>Negative transfer from common experience</u>. The space bar is used both to clear an entry and to accept it, but it is not used for its expected purpose; that is, for moving the cursor toward the right of the screen. This not only violates user expectations, but also increases the user's frequency of error and frustration. When the system uses the space bar for novel and inconsistent purposes, it violates the user's implicit assumptions about the mapping of actions onto effects. This can be especially frustrating, because such assumptions are normally unquestioned, thus making it difficult for the user to discover the source of the problem.

3. <u>Inability to undo unwanted or mistaken actions</u>. If the user forgets to designate a ship before entering its parameters, the system allows the parameters to be entered. However, if the user then enters a ship designation letter, all of the parameter entries are cleared. There is no way to undo the actions. There are really two problems here. First, the system will, without prompting for confirmation, take actions and produce states that are very unlikely to be unintentional. Some such states can be avoided entirely by changing the nature of the interaction so that the user cannot take certain actions in some environments (Black & Sebrechts, 1981; Maguire, 1982). The second problem is that, once such a state is produced and noticed, there is no way to go back to the previous state. Both of these problems add to user frustration and waste time. Solving the second problem requires making a new operation, the "undo," known to the user. Solving the first is preferable because the friendliness of the system is then transparent to the user.

4. <u>No feedback about what the system is doing</u>. The system can require a noticeable amount of time to execute computations to perform a certain task (e.g., moving the ships to their appropriate positions for some user specified time), but the user gets no indication that anything is happening during this time. Using a computer as a resource in the learning process is a type of interactive problem solving. One of the most important aspects of sharing the problem-solving load is keeping track of what the other participants are doing. Monitoring the behavior of the partner (in this case, the program) is part of the normal problem-solving routine. Failure to provide feedback about the system's behavior in terms the user can readily understand makes the interaction

unnatural and leads to errors (Norman, 1983). Users misplan their own behavior based on mistaken assumptions about what part of the total task the system has done or is doing.

5. Command names that are not good pointers to the functions they perform. While running a scenario, ships will move out of the geographic display window. Thus, to permit the student to carry on the simulation, a facility to "translate" the window over the ship is provided. The command name that invokes that facility manually is T(rans). Students had trouble understanding the meaning of this and other command names. This is another problem relating to the user's model of the mapping of actions onto effects. The user's understanding of this mapping depends upon at least the following factors: (1) "customary meanings" of the command terms (Black & Sebrechts, 1981), (2) the implicit categories of effects (e.g., display a scenario, set a speed) and actions (e.g., enter a number, call a command) recognized by the user, (3) notions of "reasonable" ways of doing things with the system (Bott, 1979), (4) the perceived domain of the entire set of commands, and (5) the contrastive relations of each command name with hypothesized "meanings" for the other command names. The customary meanings of command words is often the first factor to come to mind, but the others are also important. They are logical extensions of the principle that the meanings of words in language (command names being a small subset) do not reside in the words themselves but in the interpretation given them by the hearer/reader. Well-chosen command names help novice users develop a clear model of the system, which helps them to infer and remember the meanings of command names. A common problem here is that, because the system designer has an intimate knowledge of the semantics of the commands, as embodied in the code that executes them, he tends to choose command names that are meaningful in the context of a model of the system as an evolving body of software. However, the user needs a different type of model: one that is simple, is consistent, and refers to the system's behavior rather than to its internal workings (Moran, 1981a, 1981b). The command names seen by the user should suggest a simple and functionally adequate model rather than a complicated implementationally correct one. For these reasons, command names that seem natural or meaningful to a system designer may in fact be quite misleading to the novice who does not yet and, given poor command names, may never have a useful model of the system. The user's notion of reasonable ways to interact with the system depends on his or her prior experience. In some user populations, such knowledge can be useful both in designing the interaction and in choosing command names. When considering the "domain" of the entire command set (or subsets) and the contrastive relations among them, it must be recognized that user hypotheses about the meanings of command names do not arise in a vacuum. Likewise, command names should not be chosen without regard to the set within which they will be embedded. In the original system, command names were necessarily short because many items were competing for screen space at all times. This need for brevity put serious restrictions on their descriptive richness.

6. Failure to indicate the units or ranges of acceptable entries. Distances are commonly expressed in miles, yards, or thousands of yards. A student wishing to set a value in a problem gets no guidance on what the system expects or accepts. At best, this problem forces a guessing game on the user, leaving him to discover the units in which entries are to be made and the ranges within which entries will be expected. At worst, it may invisibly violate the user's implicit expectations and thereby lead to errors that are very difficult for the user to debug.

7. No way to distinguish those things on the screen that the user can act upon (e.g., ship parameters) from those that he cannot act upon (e.g., direction of relative motion, relative speed). This is a problem because it makes the construction of a model of the mapping of actions to effects more difficult than it need be for the user. Some items that

the user may wish to manipulate respond as expected, while others do not. Users faced with such a situation are likely to attempt to develop idiosyncratic hypotheses about why some things are sensitive and others are not. For example, in the case of the domain of relative motion (see #9 below), some parameters can be specified by the user and others cannot, leading to user confusion.

8. <u>Crowding on the screen.</u> The bottom part of the screen especially is visually very busy, making it difficult for users to find what they are looking for. When the screen is crowded, users must either devote resources to searching for items or maintain a complex cognitive map of the screen. The former strategy is time-consuming, and the latter can be a disaster if the screen configuration changes in different modes of operation.
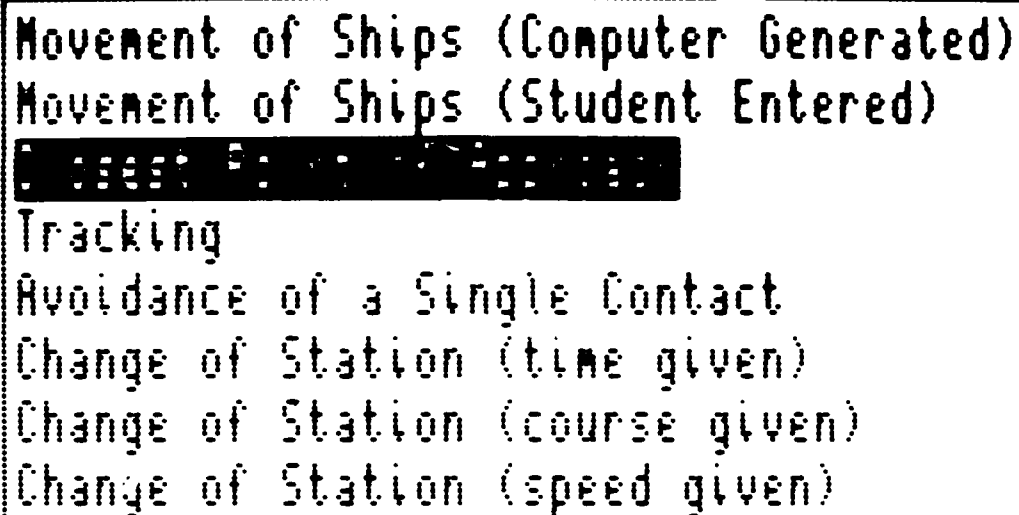
9. <u>Domain sensitivity (i.e., the way available facilities organize the user's actions is not sensitive to the representation of problems in the domain).</u> Students would sometimes like to be able to specify the directions of relative motion (DRM) and the relative speed as givens rather than having them computed. The system does not permit this option. This may be more of an instructional fault than an interface fault, but it should be considered in the interface design process. To limit the computational complexity required of the program, certain manipulations of the ship simulation, the most important of which involved the specification of the components of the velocity vector triangle, were not provided for. The velocity vector triangle is made up of three vectors—one for ownship, one for maneuvering ship, and one for the relative motion between the ships. Each vector has a direction and a magnitude; in most cases, the entire triangle is determined by any four of its six component parts. In the 3.0 version, it was possible to specify the vector triangle only by specifying the magnitude and direction of ownship and maneuvering ship vectors. In a sense, these are the only vectors that have a measurable physical existence; however, in the domain of relative motion, it is often desirable to specify other combinations of components of the triangle as givens and then solve for ownship's vector course or speed. Limiting the interaction by not permitting this option could lead students to form theories of the domain that are simply false.

The Redesigned System

In response to the problems described above NAVPERSRANDCEN staff members and researchers at the UCSD Cognitive Science Lab began the redesign effort in the summer of 1982. After Center personnel had explained the nature of the conceptual difficulties that appear to be responsible for the high failure rates in Navy courses that deal with relative motion to the UCSD group, the entire program structure was reconceived and reformulated. To facilitate communication, two microprocessor systems were moved from NAVPERSRANDCEN to UCSD and two NAVPERSRANDCEN team members spent several weeks programming at UCSD.

Because the simultaneous displays of relative and geographic plots had been so highly rated by the instructors who used the 3.0 system, it was decided to maintain a ship-motion facility that would permit students to construct and view dynamic ship interaction scenarios. In addition, the new system is designed to provide the students with practice in the six basic types of maneuvering board problems: closest point of approach (CPA), tracking, avoiding course, and three types of change of station problem—speed given, course given, and time given. The top-level menu (Figure 2) permits the user to choose either a ship interaction scenario or a specific type of ship interaction problem to be solved.

Select the type of problem you want.

```
Movement of Ships (Computer Generated)
Movement of Ships (Student Entered)
[highlighted region]
Tracking
Avoidance of a Single Contact
Change of Station (time given)
Change of Station (course given)
Change of Station (speed given)
```

Figure 2. The top-level menu for version 4.0.

Note. The student can choose any of the activities listed by positioning the highlighted region on the desired choice and pressing the "do-it" key.

Ship Interaction Scenarios

The revised system includes two modes for setting up a ship scenario: the student-entered and the computer-generated modes. In the student-entered mode, which is modeled directly on the facility provided in the 3.0 version, the student specifies the initial parameters. In the computer-generated mode, the student simply indicates his desire to view a ship interaction scenario and the system randomly generates a meaningful one. These modes are described below.

Student-entered Mode. To avoid some of the problems noted above in connection with the construction of ship interaction scenarios in the 3.0 version, a number of simplifications were incorporated in the revised system. The most major of these is the restriction of scenarios to two ships. Since all the principles of relative motion can be realized in scenarios involving only two ships, nothing conceptual was sacrificed by the restriction. Its advantages are that it (1) obviates the need to assign labels to contact ships, (2) prevents the graphical display of the scenario from becoming so cluttered as to be unreadable, and (3) guarantees that the information regarding the motion parameters of the contact ship are always visible to the student.

Several features of the system were removed or changed to simplify the display and reduce the number of controls to which the user needs to attend:

1. Control over scale of the displays was removed as an option; the scale is now fixed at 5000 yards per grid division. Consequently, scenarios involving short ranges and some geometries suffer somewhat in resolution.

2. The ability to reposition ownship on the geographic display was completely removed. As a result, scenarios cannot be viewed indefinitely with the reference ship on the geographic display. This, however, sacrifices very little, since the contact ship is unlikely to remain in the vicinity of the reference ship for long periods of time and the purpose of the display is to show the relative positions of the two ships.

3. The options dealing with the setting of initial scenario time, ticking backward in time, and jumping to distant points in time (either forward or backward) have been removed. The grids on the relative and geographic displays were denser than need be in the 3.0 system. In the redesigned system, the number of concentric range circles in the relative plot was reduced, and the rectangular coordinate grid in the geographic grid was removed completely. Since the use of the simultaneous displays primarily provides a qualitative sense of the motion of the ships, the removal of grid lines does not affect the functional value of the graphical simulation but does give the display a cleaner appearance (see Figure 3).

REFERENCE SHIP	MANEUVERING SHIP			
■■■■■■	COURSE =	DRM =	BEARING=	I'm Done
SPEED =	SPEED =	SRM =	RANGE =	Start Over
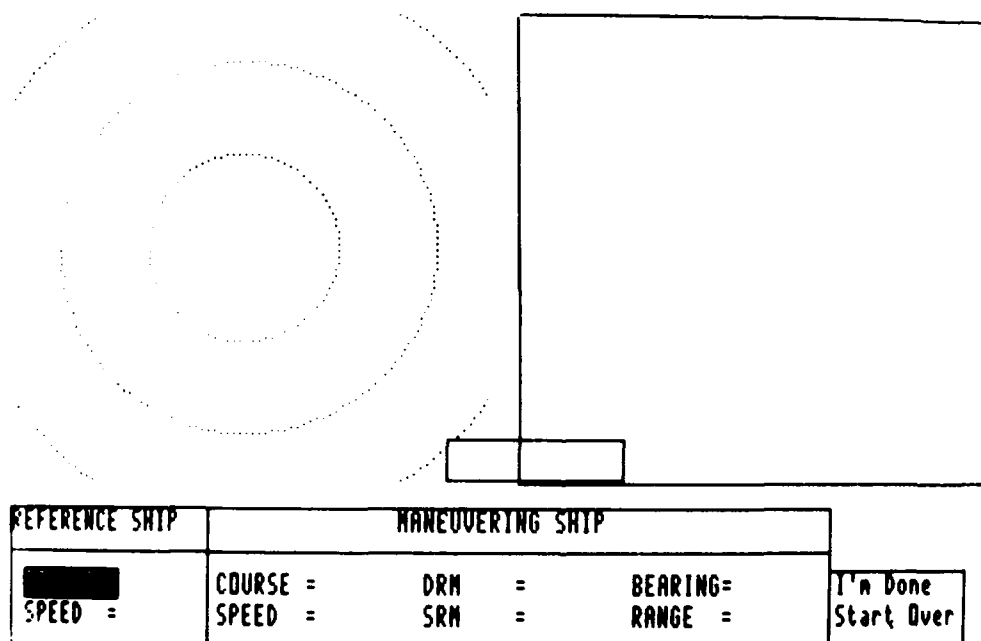
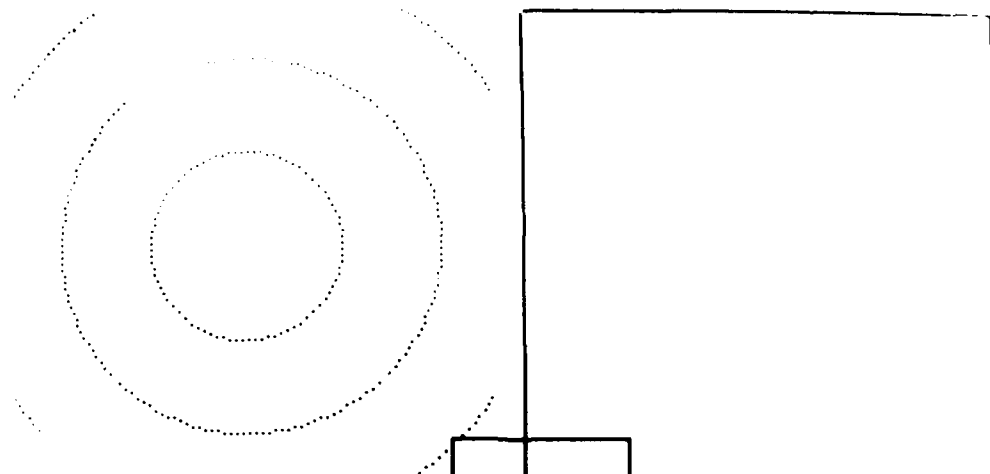Enter course in degrees (0-359) and press ENTER

Figure 3. Movement of ships (student-entered mode) display.

Note. The relative plot is on the left and the geographic plot, on the right. This is the state of the display before data are entered. Compare to Figure 1 for cleanliness.

With the items listed above removed from the system, the student retains control over constructing and running a scenario involving two ships. In setting up a typical scenario, a student might want to specify a course and speed for ownship, a course and speed for a contact ship, and an initial bearing and range of the contact. Because of the interrelationships among courses and speeds of ships and the relative motions of the ships with respect to each other, the specification of the two courses and speeds uniquely determines a direction of relative motion (DRM) and a relative speed of the contact with respect to ownship. This can be seen when the motions of the ships are represented as vectors and every two-ship interaction is described by a vector triangle. The vector triangle consists of three vectors, representing the motion of ownship, the other ship, and the relative motion between them. Since each vector has a direction and magnitude, the complete triangle consists of six components. In the case described above, when four components are specified, the other two can be computed. Generally, any four of these components uniquely determine the other two; however, there are instances in which specifying something other than the ship's courses and speeds can be instructionally effective. For example, once the initial bearing of the contact is determined, if the problem is constructed such that the DRM is the reciprocal of the initial bearing, a collision is guaranteed. In this case, the student might specify the course and speed, initial bearing, DRM, and relative speed of the contact and let the system compute the course and speed of ownship that would be required to intercept the other ship. Since leaving the student relatively unconstrained with respect to the entry of parameters may encourage the student to think about the relationships among the parameters, it seemed a very desirable feature. Understanding these relationships and the ways in which ship motion parameters constrain each other is a critical aspect of mastery of this domain.

Unfortunately, there are instances where knowing the values of four components will not enable the student to determine the values for the other two. Sometimes the four values will allow no solution, and sometimes they are consistent with two possible solutions for the values of the other components. If the student were completely unconstrained in the entry of parameters, such computationally untractable situations would eventually arise. While such situations might be very instructive for an advanced student, they would quite likely be very confusing to a novice. Although it is desirable to provide as much flexibility in data entry as possible, there is a point where a little extra flexibility adds a great deal of conceptual, computational, and interface difficulty.

It turns out that computationally untractable situations never arise if the parameters of ship motion are entered in pairs representing complete motion vectors. In the 4.0 version, students can enter any two vectors in the triangle, leaving the system to solve for the third, but they are constrained to enter both the direction and the magnitude of each entered vector. This constraint retains valuable learning situations in that students can enter the DRM and relative speed with a ship motion vector and let the other ship motion vector be determined. Furthermore, this constraint fits the pattern of availability of information in the world (i.e., normally, if one has knowledge of one component of a ship's motion vector, the other component will also be known). This constraint is enforced by capturing the cursor as soon as the student has finished entering one component of a vector. The cursor is placed on the menu location of the other component of the vector and a solid box appears around the region of the menu containing the parameters. A message appears in the documentation line requesting the entry of the parameter and the cursor will not move from that menu item until the student has entered the value. For example, if one enters a ship's speed, the cursor moves to the location of the course item, draws a box around both the course and speed items, and prompts for the entry of the ship's course. When the student has completed the entry of both components of the vector, the solid box becomes a dashed line box (see Figure 4).

9

Figure 4. Data entry to the movement of ships (student-entered mode) display.

Note. Ownship vector has been fully specified and is surrounded by a dashed box. The maneuvering ship's vector is only partially specified (course) and the cursor is trapped in the solid box until maneuvering ship's speed is entered.

For three reasons, it seemed useful to permit the student who has finished constructing a scenario to change his specification of the ship motion parameters that define a scenario:

1. The student may have simply made an error and would like to correct it without having to reenter all of the data.

2. It is instructionally useful to run a scenario under one set of parameters and then run it (or a portion of it) under slightly different conditions.

3. The automatic computation and display of the unspecified side of the vector triangle presents the student with an interactive vector calculator. For example, the student could enter a DRM and relative speed and then inspect the effects of a series of changes in one of the ship motion vectors on the other ship motion vector.

Like the situation with data entry, the facility that permits alterations in scenario parameters cannot be completely unconstrained. Once a vector triangle has been defined, a change in any of the vectors could be accommodated by an infinite number of combinations of changes in the other two vectors. For example, suppose one changes the

magnitude of the relative motion vector in a vector triangle. This change could be accommodated by changing the direction of one of the ship motion vectors, the directions of both ship motion vectors (spreading them a little further apart), the magnitude of one or both ship motion vectors, or by combinations of any of these (see Figure 5). Obviously, an interface that would permit the student to specify where the effects of his changes should be accommodated would be very complex and would require in advance just the sort of knowledge that this facility is designed to impart.
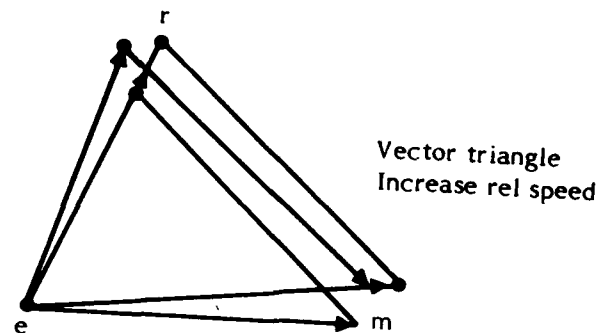

Figure 5. The vector triangle.

Note. Here are a few of the many ways that a change in one component of one vector, relative speed, could be accommodated by changes in one or more of the components of the other vectors.

The difficulty was solved by adding a constraint on the student's actions on the system. In this case, the constraint is as follows: Once the student has entered two complete vectors and the system has computed the third, the student is not allowed to change the values of the vector that has been determined by the system. When the system determines a vector, a solid box appears around its components, and the cursor will not move to the locations of the parameters of the determined vector. Any changes to either of the vectors that the student originally entered will be accommodated by changes in the vector that was originally computed by the system. In this way, the system provides reasonable but not complete flexibility in the entry of parameters and the maximum amount of flexibility in revision that is possible without unnecessarily increasing the complexity of the interface.

It should be noted that the constraints that have been implemented to keep the interface usable impose structure on system's behavior that may lead the student astray. For example, a student interacting with this system may implicitly infer that the only way to build a vector triangle is by specifying complete vectors. After all, the system will not permit a vector triangle to be constructed any other way. Such an understanding is not only incomplete, it is incorrect. While this is a potentially dangerous situation, the alternative is to provide a system that is so complex no student could learn from it.

Computer-generated Mode. To reduce the cost to the user in time and resources of setting up a meaningful ship interaction scenario to watch on the simultaneous relative and geographical displays, a facility was built that is capable of automatically generating entire sample scenarios. Using this facility, the student can either view the scenario as generated or rearrange it by changing ownship's course and/or speed.

11

Practice Problems

If the student chooses a problem type rather than the ship interaction scenario, the system presents an instance of the problem type posed as a word problem. These problem statements are modeled on those that students find in their workbooks (Figure 6). The problems themselves are generated randomly within a set of constraints that guarantee that every problem is meaningful. All CPA problems, for example, are constructed such that the CPA between ships has not yet been reached at the (simulation) time the problem is posed to the student.

Avoiding Course Problem

Your course is 350 (T), 20 knots. At 1430 a ship bears 296 (T) degrees, 23900 yards. At 1447 he bears 298 (T) degrees at 16300 yards. At 1501, come left to avoid. him by 5000 yards at present speed.

```
                  YOUR SOLUTION                      SYSTEM SOLUTION

 ┌──────────────────────────────────────────────────────────────────┐
 │ Avoiding Course       >   325                                      │
 │                                                                    │
 │ Bearing at CPA        >   4                                        │
 │                                                                    │
 │ ███████████████████   1514                                        │
 │                                                                    │
 │ Compare my answers with the system's answers                       │
 │ Get a DIFFERENT type of problem                                    │
 └──────────────────────────────────────────────────────────────────┘
```

Enter Time of CPA in 24 hour time (0001-2400) and press ENTER key

Figure 6. A typical problem statement screen (avoiding course problem).

Note. When the student chooses to do an avoiding course problem, the system poses the problem in text form and prompts for the student's solution. Here the student has just entered the last of his answers. An expanded description of what the system expects, the units required, and the procedure for completing the entry are all specified in the documentation line at the bottom of the screen. After the student enters his answers, he can compare his answers to the system's answers or get a problem of a different type.

Below the problem statement on the screen is a menu of prompts for answers to the problem and, below that, a menu of other things the student can do from that screen. The intended use of this screen is for the student to work the posed problem on an actual plotting sheet and then enter his answers in the appropriate spaces. If the student selects to have his answers compared to the system's answers, the system will display its own answers. In addition, if the student has entered any answers that are not within tolerances,[2] they are labeled "incorrect." If the student has not entered any answers, the system simply displays the correct answers. In this way, the student can get feedback about his answers to problems. Unfortunately, learning that an answer is wrong may be of little use to a student. A student wishing more useful feedback can elect either to see the solution plotted or see the simultaneous plot depiction of the interaction described in the problem (see Figure 7).

Avoiding Course Problem

Your course is 350 (T), 20 knots. At 1430 a ship bears 296 (T) degrees,
23900 yards. At 1447 he bears 298 (T) degrees at 16300 yards. At 1501,
:one left to avoid him by 5000 yards at present speed.

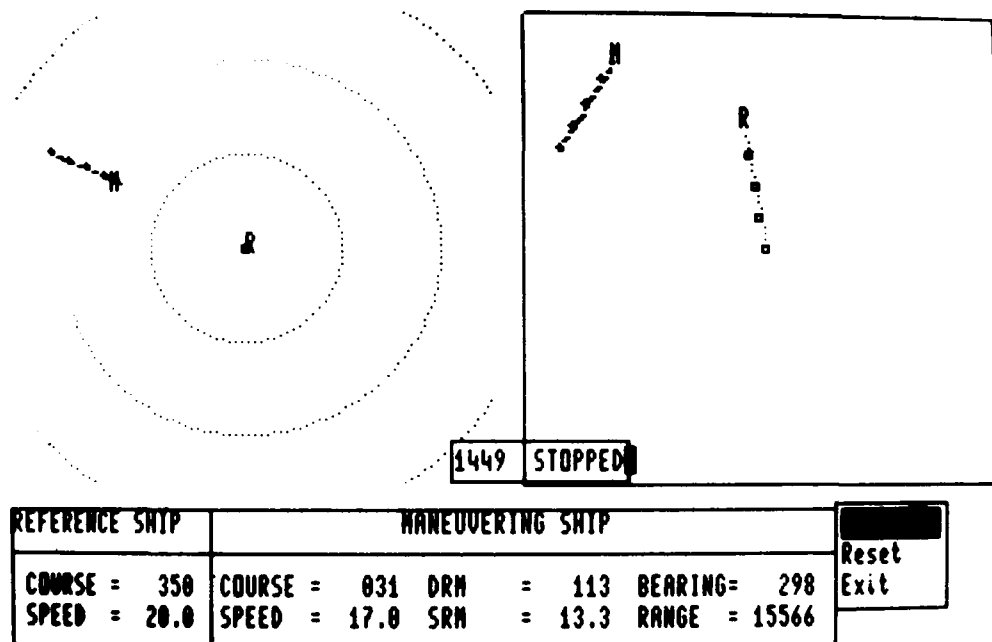|	YOUR SOLUTION		SYSTEM SOLUTION
Avoiding Course	) 325	incorrect	320
Bearing at CPA	) 4	correct!	
Time of CPA	) 1514	correct!	

```
See ship movements on the relative and geographic plots
███████████████████████████
Get another problem of the SAME type
Get a DIFFERENT type of problem
```

Figure 7. Problem statement screen with answers.

Note. Here the student has entered his answers to the problem. Correct answers are supplied where the student was incorrect. The bottom of the menu lists the options that the student can choose at this point.

---

[2]Tolerances used by instructors in the classroom are +/- 3 degrees on courses, and +/- 10% on ranges and times.

13

The motions of the ships involved in the problem is depicted in Figure 8. This screen uses the same simultaneous display used for the computer-generated movement of ships option from the top-level menu. This is like the computer-generated scenario in that the system has created the scenario, but it differs in that the student can neither enter nor change the ship parameters. It is believed that the use of this facility reinforces the connection between the points and lines that are drawn on the maneuvering board plotting sheet and events in the world of ships (e.g., the appearance of a contact on a radar screen or the geographic motion of the ships over the face of the ocean). The ability to see a problem in terms of these connections seems to be instrumental in detecting and correcting certain types of procedural errors.

REFERENCE SHIP	MANEUVERING SHIP					
						Reset
COURSE = 350	COURSE = 031	DRM = 113	BEARING= 298			Exit
SPEED = 20.0	SPEED = 17.0	SRM = 13.3	RANGE = 15566			

Start and Stop movement of ships

Figure 8. The version 4.0 movement of ships (computer-generated mode) display.

Note. The scenario given in the problem the student has been attempting to solve is depicted with simultaneous display of relative and geographic plot. The student can run the simulation through the scenario, reset the simulation to the initial conditions specified in the problem statement, or exit this screen—returning to the problem statement screen. If the student runs through the scenario, the appropriate change of course required to avoid collision will be shown.

14

The other option available to the student after attempting to solve the problem posed by the system is to view a step-by-step display of how the solution to the problem should be constructed on a plotting sheet (Figure 9). Using this facility, the student can step through a graphic depiction of the solution to the very problem he just attempted to solve. The system presents text descriptions of each step in the procedure. When the student has had a chance to read the description of the step, he is shown graphically how that step should have been carried out for the problem in question. The appropriate points are plotted and flashed on the screen to bring them to the student's attention. The student controls the rate of presentation of the steps by indicating when he is ready to see the next step. This facility represents a major advance over a system that simply checks for the correctness of student answers, because the step-by-step construction of problem solutions is precisely the skill the student ultimately needs to learn.
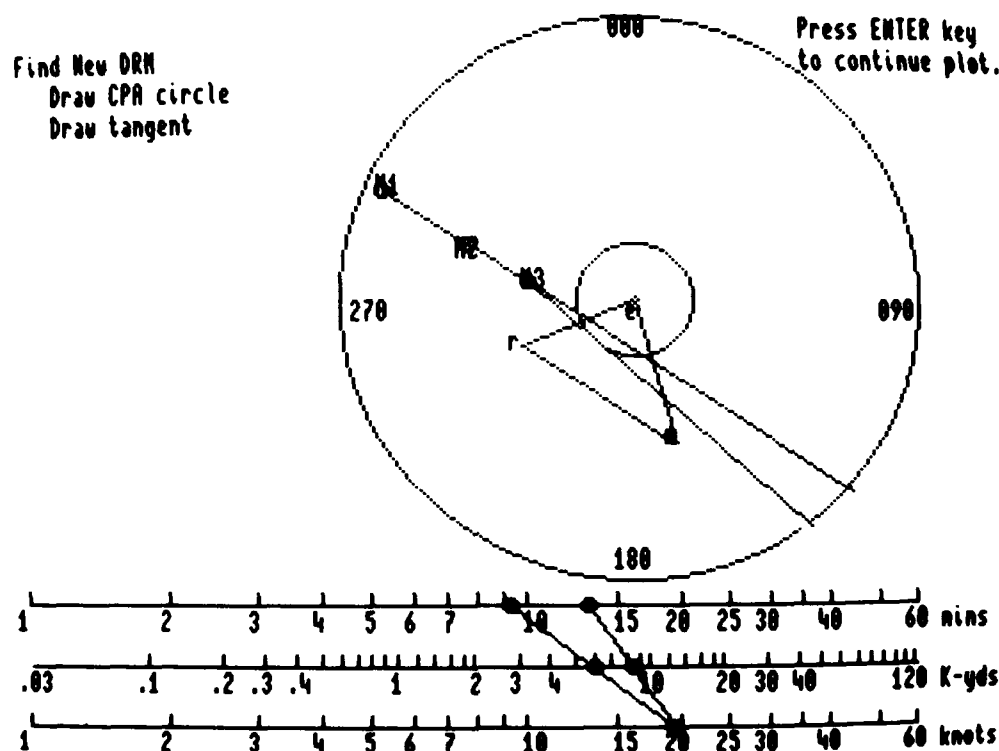

Figure 9. The graphical display of the solution procedure.

Note. Here the student is seeing some of the intermediate steps in the procedure required to solve the avoiding-course problem. At the left side of the screen, natural language descriptions of the steps appear one at a time. With the appearance of each step, the graphical actions required by that step appear on the simulated maneuvering board plot at the right. Nomogramic computations are shown at the bottom of the screen, just as the student should have done them on his MANBOARD nomogram.

General Interface Properties

The overall structure of the system is shown in Figure 10. Each screen configuration contains menu items that, when selected, will produce other screen configurations.

Select the type of problem you want.

Movement of Ships (Computer Generated)
Movement of Ships (Student Entered)
████████████████████████
Tracking
Avoidance of a Single Contact
Change of Station (time given)
Change of Station (course given)
Change of Station (speed given)
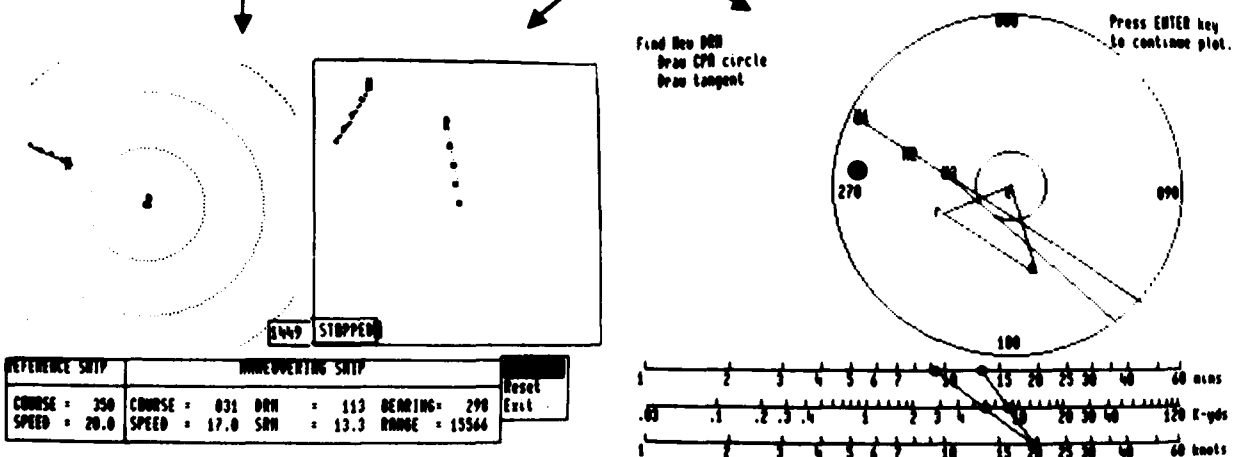
Avoiding Course Problem

Your course is 350 (T), 20 knots. At 1430 a ship bears 296 (T) degrees, 23900 yards. At 1447 he bears 298 (T) degrees at 16300 yards. At 1501 come left to avoid him by 5000 yards at present speed.

YOUR SOLUTION   SYSTEM SOLUTION

Avoiding Course	325
Bearing at CPA	4
████████████████	1514

Compare my answers with the system's answers
Get a DIFFERENT type of problem

Enter Time of CPA in 24 hour time (0001-2400) and press ENTER key

Find New DRM
Draw CPA circle
Draw tangent

Press ENTER key
to continue plot.

000
270
090
100

0449 STOPPED

REFERENCE SHIP	MANEUVERING SHIP			Reset Exit
COURSE = 350	COURSE = 031	DRM = 113	BEARING= 298	
SPEED = 20.0	SPEED = 17.0	SRM = 13.3	RANGE = 15566	

Start and Stop movement of ships

Figure 10.   Overall structure of the system.

16

Use of menus. As can be seen from the figures, all interactions with the system are conducted by use of menus. A highlighted region on the menu indicates the present potential selection and a documentation line at the bottom of the screen provides an expanded description of the currently highlighted command. Changes in selection are made by using arrow keys on the keyboard to move the highlighted region around in the menu. Commands are executed by pressing a "do-it" key, while the highlighted region is positioned on the desired menu selection.

Menus have some nice properties with respect to the mapping of actions onto effects (Perlman, 1981; Norman, 1983). The meanings of individual commands are easier to establish since the whole contrast set of relevant and related commands is visible at once. The problem of user goal formation is alleviated somewhat by having examples of reasonable things to do present. Further, it should be easier for the user to induce a model of what the system can and cannot do when the available commands appear as menu items. The addition of a documentation line as a part of the menu system permits command expansion for greater clarity and less ambiguity. The units and ranges of arguments to commands are also specified in the documentation line in a form that can be consulted if needed but is otherwise unobtrusive. Every time the cursor moves to a new menu item, the information appropriate to that menu item appears in the documentation line at the bottom of the screen.

Having a single "do-it" key that accomplishes menu selection and all data entry solves the problem of inconsistent data entry and permits the user to form a simpler mapping of actions onto effects. In the 4.0 system, there are only two user action protocols. To select a menu item, the user positions the cursor on the desired selection and presses the "do-it" key. To enter a data value, the user positions the cursor, keys in the desired value (rubout is permitted), and presses the "do-it" key. In each of these cases, the user receives immediate feedback from the system. In the case of menu selection, the system does whatever the menu selection specifies. In the case of data entry, the entered number is flashed when the "do-it" key is pressed to indicate to the student that the value has been entered into the system.

System feedback. In the 4.0 system, the student gets continual feedback concerning the activities of the system via highlighting of entries, updating of simulation clock, and changes in the appearance of the screen. The situations that were most troubling in this regard in the 3.0 system (e.g., moving the scenario forward or backward in time by large jumps) have been removed altogether from the 4.0 system.

Functionally differentiated screen configurations. Rather than try to serve all functions with a single multipurpose screen display, in the 4.0 system, different screens appear for different jobs. This encourages the users to build models of the system's behavior that recognize the difference among the various activities. It also alleviates much of the crowding on the screen, although the system is still up against the limits of the screen resolution in the display of solutions to complex problems. The explicit representation in menus of where in the system, in the sense of what sort of screen, the user can go from his present location makes it easy for the user to get from one sort of activity to another.

CONCLUSIONS

In this cycle of the design/test/redesign process, significant improvements to the user interface have been accomplished. A consideration of what was problematic in the early

system and what features solve problems in the later system sheds light on the needs of users. In addition to improving the user interface, the redesign of the system gives the student better feedback about his own performance and more direct practice of the skills required on the job. The simplified interface to the system will permit students to master the use of the system quickly and devote themselves more immediately to learning the subject matter domain (Sebrechts, 1983).

In the redesign of several features in the training system, decisions had to be made about the tradeoff between increased conceptual coverage and increased interface complexity. This tradeoff was felt most acutely in the redesign of the "student entered" mode of the motion of ships facility.

The sum of the individual design choices define a location in a hypothetical tradeoff space. Such a space is depicted in Figure 11. Of course, all of the interface complexity of a system is not necessarily there in support of conceptual coverage. In the 3.0 version of the maneuvering board program, for example, the facilities for controlling scale, time, and repositioning of ships contributed virtually nothing to the program's ability to communicate the concepts of the domain, but they did contribute to the complexity of the interface. The ability to create multiple ship scenarios, however, added greatly to the complexity of the interface and also supported some concepts that could not otherwise be represented. The movement in the tradeoff space achieved by simply eliminating the scale, time, translation, and multiple ship facilities is shown in Figure 11 by the dotted line. It represents a large decrease in complexity and a small decrement in conceptual coverage. The heavy solid line in the figure represents the locations in the hypothetical tradeoff space that would be occupied by a series of programs produced by continually increasing conceptual coverage while keeping the increase in interface complexity to a minimum. This is a line of least complex systems. In the case of the subject matter domain that was to be taught by this program, it seems that this line makes a dog-leg. There is a point beyond which a small increase in conceptual coverage becomes very costly in terms of interface complexity. As long as the corner of the dog-leg is at a point that provides both acceptable conceptual coverage and an interface that can be comprehended, the ideal point in the tradeoff space is at the corner of the dog-leg.

In retrospect, the redesign process for this program can be seen as an effort to move toward this ideal point. The redesigned system certainly provides greater conceptual coverage with less interface complexity than its predecessor. The movement of the system through the tradeoff space caused by the redesign process is shown by the dashed line in Figure 11. As new developments in interface technology become available, the shape of the line of least complex systems and the location of the ideal point in the tradeoff space will change.
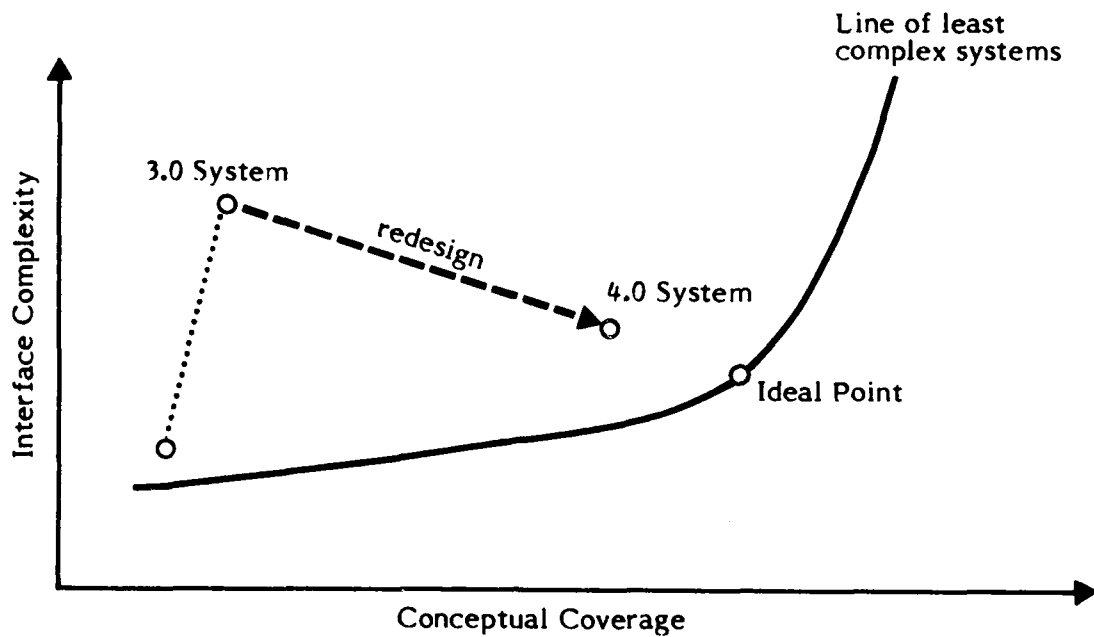
Figure 11. A hypothetical design space.

Note. Since there are no established measures for the axes of this space, the scale is unknown. This is an expression of qualitative relations among the various locations in the design space. The dotted line represents the movement of the system through the space that could have been achieved by simply removing some of the "bells and whistles" from the system. The dashed line represents the movement accomplished by the redesign process.

REFERENCES

Black, J. B., & Sebrechts, M. M. An invited article, facilitating human-computer communication. Applied Psycholinguistics, 1981, 2, 149-177.

Bott, R. A. A study of complex learning: Theory and methodologies. Doctoral dissertation, University of California, San Diego. Dissertation Abstracts International, 39, 471-B, 1979.

Hutchins, E., & McCandless, T. P. MANBOARD: A graphic display program for training relative motion concepts (NPRDC Tech. Note 82-10). San Diego: Navy Personnel Research and Development Center, March 1982.

Maguire, M. An evaluation of published recommendations on the design of man-computer dialogues. International Journal of Man-Machine Studies, 1982, 16, 237-261.

Moran, T. P. The command language grammar: A representation for the user interface of interactive computer systems. International Journal of Man-Machine Studies, 1981, 15(1). (a)

Moran, T. P. An applid psychology of the user. Computing Surveys, 1981, 13(1), 1-11. (b)

Norman, D. A. Four stages of user activities (Manuscript). Institute for Cognitive Science, University of California, San Diego, 1983.

Norman, D. A., & Bobrow, D. G. On data-limited and resource-limited proceses. Cognitive Psychology, 1975, 7, 44-64.

Perlman, G. Two papers in cognitive engineering: The design of an interface to a programming system and MENUNIX: A menu-based interface to unix (user manual) (Rep. No. 8105). University of California, San Diego, November 1981.

Sebrechts, M. M. Cognitive guidelines for computer-aided instruction: A navigational case study. To appear in: Proceedings of the 27th annual meeting of the Human Factors Society, Norfolk, 1983.

DISTRIBUTION LIST

Military Assistant for Training and Personnel Technology (ODUSD(R&AT))
Chief of Naval Operations (OP-115) (2), (OP-135C4), (OP-140F2), (OP-39), (OP-987H)
Chief of Naval Material (NMAT 0722)
Deputy Chief of Naval Material (Technology)
Chief of Naval Research (Code 200), (Code 270), (Code 440) (3), (Code 442), (Code 442PT)
Chief of Naval Education and Training (00A), (00A2), (N-6), (N-21), (N-311)
Chief of Naval Technical Training (016)
Commandant of the Marine Corps (MPI-20)
Commander Navy Recruiting Command (Code 20)
Commander Naval Military Personnel Command (NMPC-013C)
Commanding Officer, Naval Aerospace Medical Institute (Library Code 12) (2)
Commanding Officer, Naval Technical Training Center, Corry Station (Code 101B)
Commanding Officer, Naval Training Equipment Center (Technical Library) (5), (Code N-1)
Commanding Officer, Office of Naval Research Branch Office, Chicago (Coordinator for Psychological Sciences)
Commander, Army Research Institute for the Behavioral and Social Sciences, Alexandria (PERI-ASL), (PERI-ZT), (PERI-SZ)
Program Manager, Life Sciences Directorate, Bolling Air Force Base
Commander, Air Force Human Resources Laboratory, Brooks Air Force Base (Manpower and Personnel Division), (Scientific and Technical Information Office)
Commander, Air Force Human Resources Laboratory, Williams Air Force Base (AFHRL/OT)
Commander, Air Force Human Resources Laboratory, Wright-Patterson Air Force Base (AFHRL/LR)
Headquarters, Coast Guard
Commanding Officer, U.S. Coast Guard Research and Development Center, Avery Point
Institute for Defense Analyses, Science and Technology Division
Defense Technical Information Center (DDA) (12)

23

END

(FILMED)

5-84

DTIC