

AD-A246 171



NAVAL POSTGRADUATE SCHOOL
Monterey, California

2



DTIC
FLECTE
FEB 21 1992
S D D

THESIS

DEVELOPMENT OF A GRAPHICAL INTERFACE
FOR A MAINTENANCE MANAGEMENT
DATABASE SYSTEM

by

Jeffrey J. Mahoney

September, 1991

Thesis Advisor:

C. Thomas Wu

Approved for public release; distribution is unlimited.

92-03718



92 2 12 170

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED		1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE			
4. PERFORMING ORGANIZATION REPORT NUMBER(S)		5. MONITORING ORGANIZATION REPORT NUMBER(S)	
6a. NAME OF PERFORMING ORGANIZATION Computer Science Dept. Naval Postgraduate School	6b. OFFICE SYMBOL (if applicable) CS(52)	7a. NAME OF MONITORING ORGANIZATION	
6c. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943		7b. ADDRESS (City, State, and ZIP Code)	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (if applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c. ADDRESS (City, State, and ZIP Code)		10. SOURCE OF FUNDING NUMBERS	
		PROGRAM ELEMENT NO.	PROJECT NO.
		TASK NO.	WORK UNIT ACCESSION NO.
11. TITLE (Include Security Classification) DEVELOPMENT OF A GRAPHICAL INTERFACE FOR A MAINTENANCE MANAGEMENT DATABASE SYSTEM			
12. PERSONAL AUTHOR(S) Mahoney, Jeffrey, J			
13a. TYPE OF REPORT Master's Thesis	13b. TIME COVERED FROM 9/90 TO 9/91	14. DATE OF REPORT (Year, Month, Day) 1991 September 26	15. PAGE COUNT 52
16. SUPPLEMENTARY NOTES THE VIEWS EXPRESSED IN THIS THESIS ARE THOSE OF THE AUTHOR AND DO NOT REFLECT THE OFFICIAL POLICY OR POSITION OF THE DEPARTMENT OF DEFENSE OR THE U.S. GOVERNMENT.			
17. COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	Database, Openscript, User-Interface	
	SUB-GROUP		
19. ABSTRACT (Continue on reverse if necessary and identify by block number)			
Winomms is a prototype graphical interface designed to support the Navy's goal of paperwork reduction. Designed to replace the existing interface of the Navy's Maintenance Data System program, "MicroOmms", Winomms provides an intuitive easy to learn and use graphical environment that greatly enhances productivity for shipboard maintenance requirements.			
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS		21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a. NAME OF RESPONSIBLE INDIVIDUAL C.T. Wu		22b. TELEPHONE (Include Area Code) (408) 646-3391	22c. OFFICE SYMBOL CSWq

Approved for public release; distribution is unlimited.

Development of a Graphical Interface
for a Maintenance Management
Database System

by

Jeffrey J. Mahoney
Lieutenant, United States Navy
B.A., University of Kansas

Submitted in partial fulfillment
of the requirements for the degree of

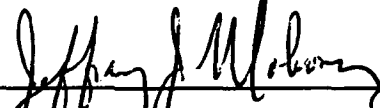
MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

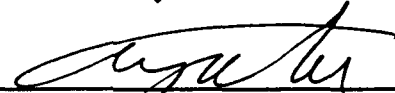
NAVAL POSTGRADUATE SCHOOL

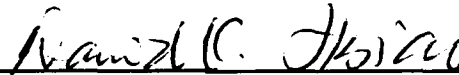
September 1991

Author:

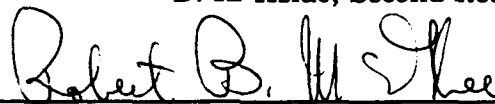

Jeffrey J. Mahoney

Approved by:


C. T. Wu, Thesis Advisor



D. K. Hsiao, Second Reader



Robert B. McGhee, Chairman
Department of Computer Science

ABSTRACT

Winomms is a prototype graphical interface designed to support the Navy's goal of paperwork reduction. Designed to replace the existing interface of the Navy's Maintenance Data System program, "MicroOmms", Winomms provides an intuitive easy to learn and use graphical environment that greatly enhances productivity for shipboard maintenance requirements.

Accession For	
NTIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/ _____	
Availability Codes	
Dist	Avail and/or Special
A-1	



TABLE OF CONTENTS

I. INTRODUCTION	1
A. THE NEED FOR A BETTER INTERFACE	1
B. WHY USE A GRAPHICAL USER INTERFACE ?	2
C. A BETTER WAY TO USE AN EXISTING DATABASE SYSTEM	3
D. OUTLINE	3
II. THE PROBLEM STATEMENT	5
A. BACKGROUND	5
B. EXISTING GRAPHICALLY BASED MICRO-COMPUTER SYSTEMS	7
III. APPLICATION ENVIRONMENT AND DESCRIPTION	9
A. DESCRIPTION OF "WINOMMS" ENVIRONMENT	9
1. What is "Winomms" ?	9
2. The Application Environment	9
3. The Development Software: TOOLBOOK	10

B.	DESCRIPTION OF "WINOMMS" APPLICATION	10
1.	Why Winomms was Developed	10
2.	The Problems With the Existing System	11
3.	How Winomms Corrects These Problems	11
C.	SPECIFIC CONSTRAINTS FOR THIS APPLICATION	12
1.	Who Will Use This Application	12
2.	Relation between Design Goals and TOOLBOOK	13
3.	Difficulties in using TOOLBOOK	14
D.	DESCRIPTION OF THE PROGRAM	15
1.	System Requirements	15
2.	Implementation Tools	16
3.	Interface Between DOS, Windows, and Toolbook	17
E.	DESCRIPTION OF THE MAIN MODULES OF THIS APPLICATION	18
1.	Log On Screen	18
2.	Main Menu Screen	18
3.	User Choice Cards	20
4.	Form OPNAV 4790-2k Screen	22
5.	Use of Pop-Up Menus	23
6.	Selecting a Deferred OPNAV 4790-2K	24
7.	Completing a Deferred OPNAV 4790-2K	26

IV. DESIGN AND IMPLEMENTATION	28
A. DESIGN GOALS AND ISSUES	28
1. Winomms Design Goals	28
2. Winomms Design Issues	28
B. IMPLEMENTATION ISSUES	30
1. Overview of Implementation	30
2. Implementation Details	32
a. Screen Layouts	32
b. Navigating in Winomms	34
V. APPLICATION SUMMARY	36
A. FACTORS AFFECTING PERFORMANCE	36
B. TYPICAL APPLICATION PERFORMANCE	36
VI. CONCLUSION	38
A. SUMMARY OF RESEARCH	38
B. REVIEW OF RESULTS	38
C. FUTURE DEVELOPMENTS AND UPGRADES	39
LIST OF REFERENCES	40
BIBLIOGRAPHY	41
DISTRIBUTION LIST	42

LIST OF FIGURES

Figure 1 Winomms MainMenu	19
Figure 2 User Selection Card	20
Figure 3 OPNAV 4790-2K Form	22
Figure 4 Pop-Up Menu Codes	23
Figure 5 Selecting a Deferred OPNAV 2K	25
Figure 6 Completing an OPNAV 2K	27

I. INTRODUCTION

The use of relational database management systems on microcomputers has increased. Although there are many different ways to present information and allow user manipulation of this information, the most efficient method is using a graphically based interface. This thesis addresses the issue of graphically based user interfaces on microcomputers for accessing and manipulation of an existing relational database. The purpose of this graphical user interface is twofold: first, allow even the occasional user to quickly produce productive results with a minimum of training time and computer experience and second, to design the new interface in such a way that the experienced system user does not feel alienated from a system he is accustomed to.

A. THE NEED FOR A BETTER INTERFACE

A common microcomputer database management system interface is the menu system. This method presents a numbered set of choices to the user who must then enter the appropriate number and press the enter key. If many choices are available then the screen is either confusingly cluttered with text or the user must step through a series of menus to find the selection required. Computer user studies have shown that the ideal amount of information to

present on any one screen is four to six items. [REF 1:pp. 18-22]. Use of more than this optimal amount causes confusion or detracts from efficient use of time by causing the user to spend too much time reading screens to find the choice he is looking for. This type of interface is clearly undesirable. A better interface is needed to exploit the power of modern microcomputers and increase productivity of the user.

B. WHY USE A GRAPHICAL USER INTERFACE ?

Studies have shown [Ref. 1:pp. 1-24] that redesign of the computer interface can make a substantial difference in learning time, performance speed, error rates, and user satisfaction. Of many different approaches tried, the graphical interface has the widest acceptance. Evidence of the popularity of the graphically based interface can be seen by examining almost any current commercial software product for microcomputers. These products all contain some type of graphically based interface that includes the use of pull-down menus, pop-up menus, and mouse-selectable icons that open windows into other system or area applications. Use of a graphically based interface allows the user to make selections with one keystroke or one click on a mouse. This method is superior to an interface that requires typing in often cryptic and difficult to remember commands.

C. A BETTER WAY TO USE AN EXISTING DATABASE SYSTEM

The purpose of this thesis is to present the principles of a modern graphically based user interface that will act as a "front-end" interface residing on top of an underlying relational database. This differs from most current graphical interface systems that are designed concurrently with the data or system that they will control.

This thesis will implement a prototype graphical front-end that will interface with an existing relational database system replacing the existing menu driven interface. It will show that this graphical interface is superior to the menu driven system. The issues of user learning time and increased productivity due to the use of a well designed graphical user interface will be addressed during the description of this implementation.

D. OUTLINE

This research shall present one major topic per chapter. The major focus of this thesis is to explore a good graphical interface for an existing database, and determine how could such an interface be implemented on the IBM compatible personal computer. A prototype is developed to illustrate these concepts. The thesis is organized as follows:

Chapter II describes some of the aspects of current menu-driven interfaces that need changing and possible solutions to these problems.

Some of the research being done in this area is examined followed by the specific areas of this problem that this thesis addresses.

In Chapter III a description of the prototype application called "Winomms", is presented. This chapter shows how the issues involved in graphical interfaces are used in the prototype. Specific constraints of this application and who it is designed for are also discussed.

Chapter IV describes the design goals and how they affected the implementation of the prototype.

In Chapter V a summary of the results of this research and the performance it achieves will be discussed. Issues such as space and environment requirements are also detailed in this chapter.

Chapter VI is the concluding chapter and primarily addresses the accomplishments of this research, areas for improvement and areas of future research. The topics listed above are presented to facilitate a clear understanding of the issues involved in designing a graphical interface for an existing system. The goal of this research was in part to develop a graphical front-end that would smoothly interface with an existing database system and provide a more efficient, easier to use system for the end user.

II. THE PROBLEM STATEMENT

A. BACKGROUND

Modern Naval ships have become increasingly complex. The engineering, weapons, and communication facilities have all benefitted from technological advances. These technical advances require a large volume of technical manuals and maintenance documents to be kept onboard. In an ongoing effort to manage the ever increasing paperwork load the U.S. Navy has invested a substantial amount of manhours and funding in the acquisition, maintenance, and training requirements for onboard data maintenance computer systems. To support these requirements, one such system currently in use is the Shipboard Naval Automated Data Processing (SNAP II) system. This mini-computer based system is a text oriented data management tool designed to handle supply part ordering, maintenance documentation, and administrative functions. A subset of this system has also been implemented as the Micro Computer Onboard Maintenance Management System (MicroOmms). MicroOmms, designed and implemented on an IBM compatible personal computer, was developed as a low-cost, maintenance management system capable of being implemented on existing IBM compatible computers in the fleet.

In order to facilitate ease of use and minimum training time for new users, MicroOmms was designed to duplicate the look and feel of SNAP II. This implementation presents the user with a text-based selectable menu driven system. There is no use of colors or any graphical information. The user must select from a menu and enter numbers or function key choices to navigate through the MicroOmms system. As the user progresses through the menus available, backtracking to the main menu becomes increasingly difficult. It is possible that up to twelve repeated keystrokes can be required to return to the main menu from a data entry screen that the user decided not to use. The ability to return easily and quickly to the main menu in any menu-driven system is important for several reasons: As is implied in the name, the main menu is the controlling screen for the system in use. From the main menu all others options are branched out in layers or levels depending on the breath and scope of the underlying database system. As the user navigates through the system, he is normally getting further away from the top level where the main menu resides. Upon completion of the user's selected option, a well designed interface will allow instant exit from the system, with appropriate updates reflecting the task just completed, or a way to instantly return to the main menu.

Many current systems, as noted above, require many keystrokes to navigate back to the main menu. This method is time consuming and serves no useful purpose. A well designed graphical interface will always allow easy

exit from the area the user is currently using. The user should be able to return to his last choice, return to the main menu, or exit the system from any point in the system with a minimum of keystrokes.

Although this micro processor implementation addresses the problem of a paperwork overload it is clearly not optimal. The user interface is awkward to learn and use. With the current implementation of MicroOmms there are no facilities provided to improve the user interface. It is very important to provide a system that meets the needs of the modern technically based operating Naval forces and yet be easy to implement, learn, and use.

B. EXISTING GRAPHICALLY BASED MICRO-COMPUTER SYSTEMS

One solution to this problem is an on-going series of projects being developed at the Naval Postgraduate School known as "ARGOS" [REF 2]. This project seeks to address the issue of a paperwork overload onboard Naval vessels by automating many of the routine reports and documentation required in the daily operations of a Naval ship.

The prototypes for this system are implemented on Apple Macintosh computers using the HyperCard environment development system. The key factor of the "ARGOS" project is its extensive use of a graphical interface.

The major features of "ARGOS" including its similarities and differences from this research are worth pointing out since they address some of the key

issues current research is stressing in developing a graphical based user interface.

"ARGOS" offers the user an integrated graphical interface with a carefully designed underlying data structure that is easy to use and learn. The prototype implemented for this research, "Winomms" also offers an integrated graphical interface but is different from "ARGOS" because it is a top layer of interface or a "front-end" for an existing database system. While this approach places some restrictions on the design of the interface due to the constraints of previously defined data structures, it can reduce the amount of development time for a functional system. One of the key issues examined in this thesis is the feasibility of developing an interface for an existing system or to design the system and the interface together.

The "ARGOS" project is being developed as a series of highly integrated modules that can be adapted to interface together. This application was prototyped as a stand alone system. This restriction was necessary since the underlying database that this research interfaces with was also designed as a stand-alone system. The primary goal of this application was not to develop an integrated system but to significantly improve the learning time and productivity of an existing system. This goal is accomplished by replacing the existing systems menu based interface with a graphically oriented interface that accomplishes both of the goals stated above.

III. APPLICATION ENVIRONMENT AND DESCRIPTION

A. DESCRIPTION OF "WINOMMS" ENVIRONMENT

1. What is "Winomms" ?

Winomms is a Microsoft Windows application [REF 3]. It was developed using Asymmetric's Windows application development software TOOLBOOK [REF 3]. This combination of development and application software was chosen because it is ideally suited for rapid development of graphically based interfaces.

2. The Application Environment

The Microsoft Windows environment is graphically oriented. The user navigates through various options available by selection of icons, representing a particular application. Generous use of pull down and pop-up menus is also employed primarily using a mouse device. This method of interface is employed extensively in "Winomms" because it lends itself well to the central issue of a good graphical interface: ease of use. This ability to launch an application and navigate through out the system without having to remember and type numerous commands is highly desirable.

3. The Development Software: TOOLBOOK

TOOLBOOK is a Windows application development software. It allows creation of graphically oriented programs that are completely integrated into the Windows environment. The major feature of TOOLBOOK is its ability to allow rapid prototyping of a system. Through the use of pre-defined tools a graphical environment can be rapidly and easily developed.

The Winomms environment then, is the Windows environment. This means that Winomms will run like any other Windows application. It may be launched by opening the file from a Windows pull-down menu or by simply clicking with a mouse on the Winomms icon. This feature is desirable in a graphically based system because it allows a great degree of continuity switching from one application to another.

B. DESCRIPTION OF "WINOMMS" APPLICATION

1. Why Winomms was Developed

This research was developed to show one of many possible ways of greatly improving an existing menu based interface with a graphically based system. By employing judicious use of icons and buttons the end user is easily able to navigate through what was previously a complicated set of actions.

2. The Problems With the Existing System

The menu system for MicroOmms, the system this research used to prototype a new interface for, is awkward to use and requires far too many key strokes to execute a function. For example to return to the main menu requires the user to type in the number 15 and press the enter key as many as nine times before he is finally back to the main system menu. Winomms, through use of a script attached to an icon, allows this same action "return to main menu" to take place with one click of a mouse from any screen on the system.

The existing system frequently has seven to twelve options in varying places on the screen that can be chosen. This is in addition to whatever information is currently being displayed from the underlying database. This situation is clearly undesirable. These screens are very confusing and cluttered. It often takes even a frequent user an excessive amount of time to search the screen for the next option desired, find the necessary key sequence to press and finally execute the action.

3. How Winomms Corrects These Problems

This application was developed with the goal of providing a graphical interface that alleviates the cluttered screens and numerous keystrokes required of many menu based interfaces. The user is presented with clear, intelligently colored screens that contain a minimum of items to select from. Navigation through the system is accomplished through the use of a "point-and-click" with a mouse device scheme on clearly defined icons and buttons.

Quick entrance and exit from the system is maintained throughout the system. This feature is available through the use of buttons labeled "Return to Main Menu" and "Exit Winomms" that are present on every screen.

C. SPECIFIC CONSTRAINTS FOR THIS APPLICATION

1. Who Will Use This Application

This application is designed to interface with the MicroOmms database system developed a subset of the U.S. Navy SNAP II system. It contains the same terminology and the same description of options available as the menu based interface it replaces. This feature allows the experienced MicroOmms user to easily adapt to a new and better interface. The average system user, the fleet Divisional Petty Officer, will find this system far superior to the current system. As a new user of this system, he will be able to produce deferred OPNAV 4790-2K reports with a minimum of effort, including looking up special codes and information required on these reports such as Allowance Parts Lists "APL's" and Equipment Identification Codes "EIC's".

The Winomms application is also designed to improve the productivity of experienced users by allowing faster navigation through the system and to reduce the learning time of new users by providing a clearly defined easy to select set of options to choose from.

2. Relation between Design Goals and TOOLBOOK

The application software TOOLBOOK provides a structure that is comparable to that of an ordinary book. A TOOLBOOK application consists of one or more pages that combine together to make up a book. A completed book is one large file that is executed within the Windows and TOOLBOOK programs. Each page of a book can contain almost any number of buttons, fields and graphical or animation images and can contain scripts that execute programmed functions such as navigation to another page or opening and reading a database file.

All TOOLBOOK pages contain backgrounds which are the underlying structure of the page. The key feature of the TOOLBOOK background is that it is portable. Once a background is designed, such as the OPNAV 4790-2K page (see Figure 3), it may be duplicated and used on as many pages as desired. The first page is analogous to a table of contents and succeeding pages make up the sections of the book. This concept lends itself well to the Main Menu of this application. The Winomms Main Menu as shown in Figure 1 is a single page that is the table of contents for this application. From this page the user can either close the book by selecting the "Exit Winomms" button or navigate to the area of the book desired by selecting the appropriate choice. This menu scheme is the electronic equivalent of opening a book, browsing through the table of contents and then turning to the page of interest.

The design goal of a mouse driven interface is particularly well supported by TOOLBOOK. TOOLBOOK provides an interpreted script language called Openscript. Openscript is easily programmed to react to all mouse movements and left and right "clicks" of the mouse buttons. This is done by means of commands such as "to handle MouseEnter ... ". This script will be activated when the user moves the mouse pointer into any object such as a button or text field, that the script is attached to and could cause actions such as color changes to highlight the current area the user is using.

Color schemes are also well supported by TOOLBOOK and are an integral part of this application. Any page or object on that page may be colored by use of a TOOLBOOK provided color palette. Each of the choices available from the Winomms Main Menu are color coded. The same color is then used as border outlines on subsequent choices or text pages throughout that area selected. Therefore if the user selects "Maintenance Data Systems" from the cyan colored Main Menu button, he will always know he is in the "MDS" section of Winomms by the presence of cyan borders on all the pages he navigates to.

3. Difficulties in using TOOLBOOK

Simplicity in the form of uncluttered screens, good color schemes and quick navigation were all goals of this research that were largely supported by TOOLBOOK. The primary difficulty with TOOLBOOK is achieving quick navigation. Since Openscript is an interpreted language, such actions like

opening the book, navigating to various pages and in particular accessing database files are slow. This slowness, however is relative to running an application straight from MS-DOS, the standard disk operating system for IBM compatible computers.

Since TOOLBOOK applications run from interpreted scripts in the Windows graphical environment (which is still using DOS interrupts) they are slow due to the large overhead involved in bit-mapped graphics processing by the CPU. The only current solution to this problem is to invest in hardware that speeds up graphics processing such as graphics co-processor cards or main memory cached fast CPU's like the Intel 486.

This application running on an 386 IBM compatible with 4 Megabytes of RAM is still faster than the existing application that it front-ends (MicroOmms) running on an 286 IBM compatible.

D. DESCRIPTION OF THE PROGRAM

1. System Requirements

This application requires an IBM PC compatible computer that has the following features:

- INTEL 286, 386, or 486 CPU
- 40 Megabyte Hard drive

- 2 Megabytes Ram
- Microsoft Windows Version 3.0
- Runtime version Toolbook

The list above details the minimum system requirements to effectively run this application. Performance is affected positively by the addition of more system RAM, faster hard drives, and a fast, cached CPU.

2. Implementation Tools

This application is a Windows 3.0 application. This program will only run in the Windows environment. Additionally, a runtime version of Toolbook is required.

Windows 3.0 is a graphical interface environment for IBM PC compatible computers. It provides a easy to use method of launching applications. Liberal use of small graphic images (icons) that represent an application are used.

Other common graphical interface items include pull-down menus and pop-up menus. One of the most compelling reasons for developing this research application in the Windows environment is that all Windows application behave in very similar ways. Starting, pausing and stopping a Windows application is done the same way for any application. The same methods hold true for opening files, viewing directories and printing operations. This makes Windows an ideal environment for this application. The same familiar

commands used for any other application are used in this application. The learning time is significantly reduced because of this feature.

3. Interface Between DOS, Windows, and Toolbook

The most used operating system on an IBM PC compatible is either PC-DOS or MS-DOS. Although Windows gives the appearance of being an operating system it is not. Windows resides on top of and interfaces with the systems DOS. The advantage that Windows provides is it's management of system memory. With few exceptions, DOS applications are limited to 640 Kilobytes of RAM for running a program. Windows takes advantage of a systems extended or expanded memory and effectively allows an application to use 4 Megabytes or more of RAM.

Just as Windows runs in the DOS environment, Toolbook runs only in the Windows environment. A Toolbook application consists of one file or program that will only execute under a runtime or production version of TOOLBOOK. A Toolbook application will always have a file extension of ".TBK" and can be executed by any of the several methods available to launch any other Windows application. Thus to the end user, a Toolbook application is simply another Windows program available to run by simply clicking the mouse pointer on the appropriate icon.

E. DESCRIPTION OF THE MAIN MODULES OF THIS APPLICATION

1. Log On Screen

One of the governing features of this application is its similarity to the existing application, MicroOmms. The log on screen is an example of this. Upon opening the application, the user is presented with a graphical screen titled "Winomms" that quickly fades to the log on screen. Here, the user is prompted to enter his ID name or number and his password.

2. Main Menu Screen

Successful log on takes the user to the main menu screen. This screen demonstrates several of the user interface features employed throughout this application. As depicted in Figure 1 on the next page, the options available to the user are selected by pointing the mouse and clicking on the desired choice.

The top of this screen and all Winomms screens contains a title that shows the user where he is in the system. Also included in every screen is the fast exit option "Exit Winomms". As is true with any Windows application, this screen can be made smaller, moved, or "minimized" to the size of an icon. These features add a great deal of flexibility. The user can minimize his current screen (which freezes the application in its current state) perform some other task(s) and then "click" on the Winomms icon to return to where he was in the system.

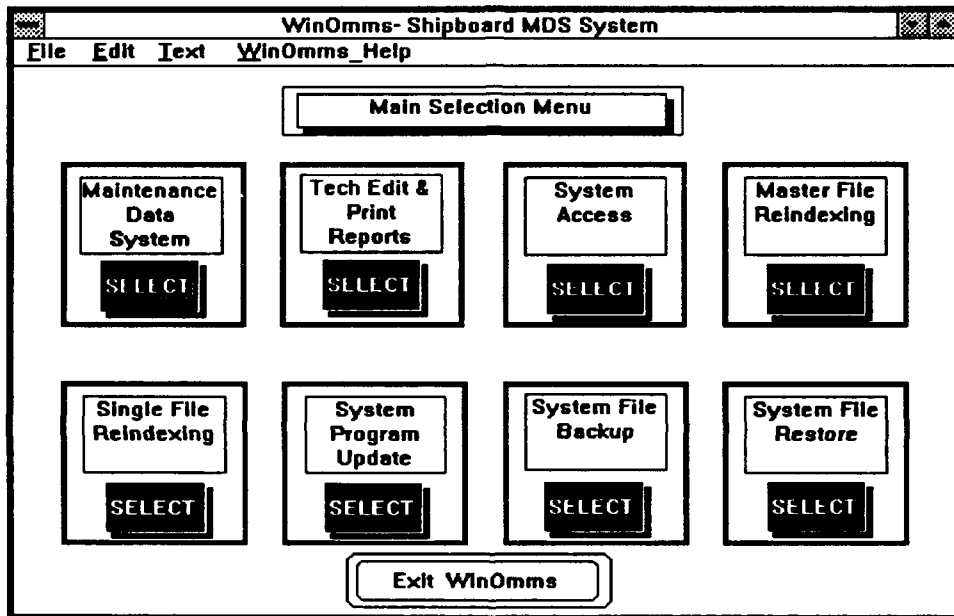


Figure 1. Winomms Main Menu

3. User Choice Cards

Figure 2 below shows one of the "index type cards" of Winomms used to select options.

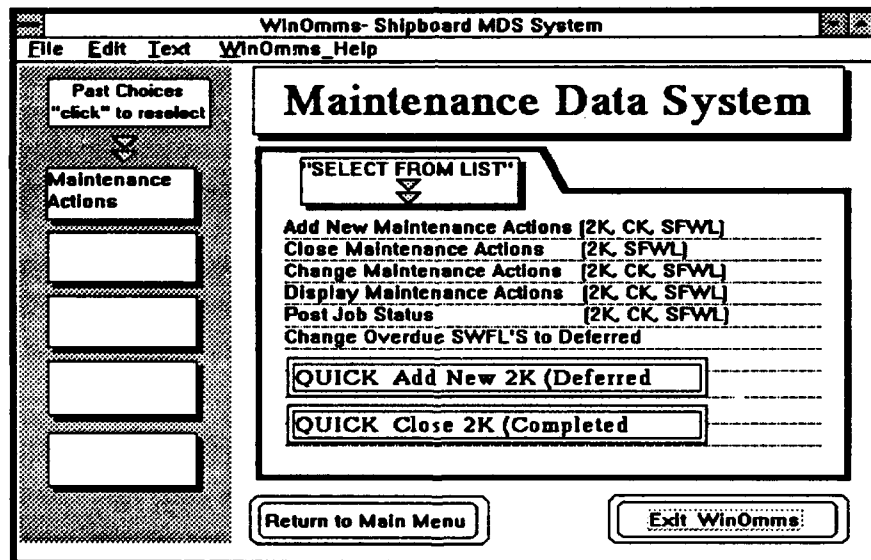


Figure 2. User Selection Card

The title at the top "Maintenance Data System" indicates where in the system the user is. This sub-section was arrived at by clicking "Maintenance Data System" from the Main Menu.

An important feature of a good user interface is the ability to backtrack. This means being easily able to retrace steps that led you to where you currently are in the system. Winomms accomplishes this objective in this card selection area (see Figure 2) by providing the "Click to Select Previous Choices" button. As the user selects from an available option on one of the cards in this screen, that choice made is placed in selected order into the boxes under the "Previous Choices" button. To retrace or undo a choice the user simply clicks on the appropriate box and the system returns to the point where that choice was made.

A feature added to this application that is not found on the existing application is the "Quick Add... and Quick Complete..." buttons. The existing system required the user to step through up to 7 screens before finally getting to the data entry screen for an OPNAV 4790-2K. This method is employed to allow the user to select from the database 4 or 5 identifying numbers and names that would be pre-filled in on the blank form. To facilitate rapid accomplishment of this particular task, Winomms allows one click of this button to go to the data entry form. Form filling is then easily accomplished through an attractive multi-colored screen. If the user still requires to have the

pre-filled in data on his form, then the longer method of accessing other databases can still be employed.

4. Form OPNAV 4790-2k Screen

As shown in Figure 3 below, this screen allows rapid and easy data entry in the Navy standard deferred maintenance form, OPNAV 4790-2k.

The screenshot displays a software window titled "WinOmms-Shipboard MDS System" with a menu bar containing "File", "Edit", "Text", "WinOmms", and "Help". The main window shows "Page 1 of 3" and the form title "OPNAV 4790-2K Deferred Action". The form is organized into several sections:

- Work Center:** EM81
- JSN:** 0048, **EIC:** 239111
- Insuro Number:** n/a
- APL/AEL:** 045-0983471
- MND:** 4, **STA:** 2, **CAS:** 7, **DFR:** 5
- Equip. Name:** 400 HZ HG
- Serial Number:** 04-AH6781
- Safety:** V
- Location:** 5-116-E-2
- Discovered Date:** 01234
- Alterations:** none
- INSURO ITEMS:** Number: n/a, Suffix: []
- SFMM EXP:** 12, **Def Date:** 01235
- SFMM RPT:** 16, **Deadline Date:** 01248

On the right side of the form, there are buttons for "Clear All Fields", "Next Page", "Previous Page", and "Stop". At the bottom of the window, there are three buttons: "Return to Main Menu", "Return to Last Choice", and "Exit WinOmms".

Figure 3. OPNAV 4790-2k Form

When this screen appears, the cursor is automatically positioned in the first data entry block and that block is high-lighted. The TAB key moves the cursor to the next data field, that field is highlighted and the previous field is un-highlighted. This process continues for the remainder of the page.

Additionally, the user may click with the mouse on any field desired to place the cursor there. This feature allows quick correction of errors or omissions.

5. Use of Pop-Up Menus

Another good interface feature of this application is the use of pop-up menus that allow accurate and fast entry in critical data fields that often cause rejection of this form due to errors. As depicted in figure 4 below, the "When Discovered" block options are presented with the appropriate required codes. By clicking on the desired option, the correct code is automatically placed in the entry field.

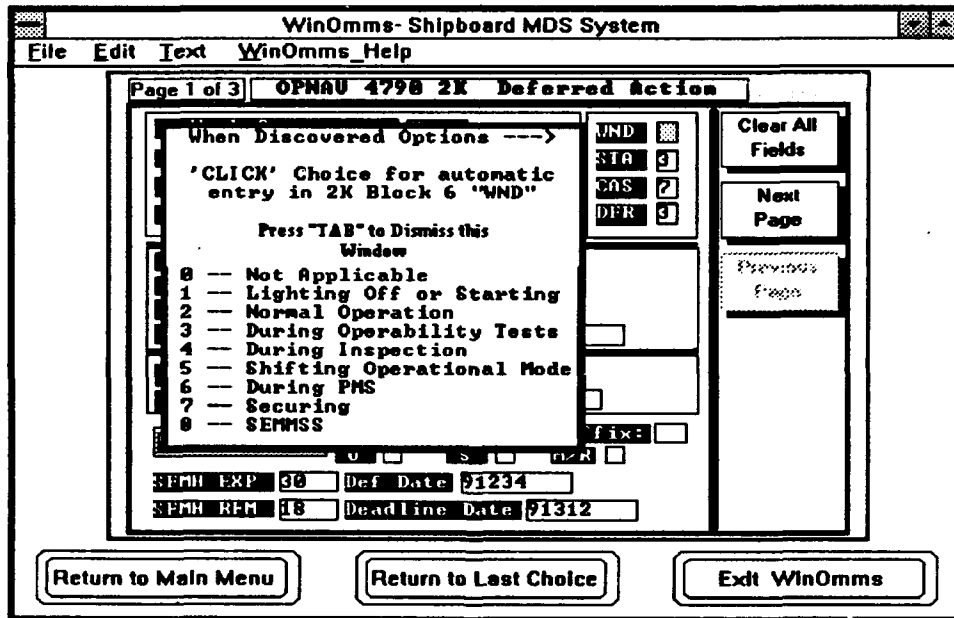


Figure 4. Pop-Up Menu Codes

When the user is satisfied with this first page of this data entry form he clicks on the "Next Page" button and continues in the same manner as described above. Upon completion of the OPNAV 4790-2K, the user selects the "Save" button on the last page and the form is then placed into the appropriate area of the underlying database.

6. Selecting a Deferred OPNAV 4790-2K

When a maintenance action or requirement is entered into the system it is considered "deferred" until several blocks on the OPNAV 4790-2K form are filled in. These blocks are the "completion section" of the 2K form.

To complete a deferred 2K, the user must search the database, retrieve the required deferred action, and fill in the completed section of the form. In typical Navy fleet operations the blocks labeled "JSN" or Job Sequence Number, and "SUMMARY" which is a one line description of the deferred action, are used to identify any particular job.

Using "JSN" as the primary key, the database is indexed and three fields are presented to the user: The Work Center, which uniquely identifies the user's shipboard work area designation, the JSN, and the one line Summary. The user can then easily scroll through the view presented (see Figure 5 next page) and select the deferred maintenance he wishes to complete.

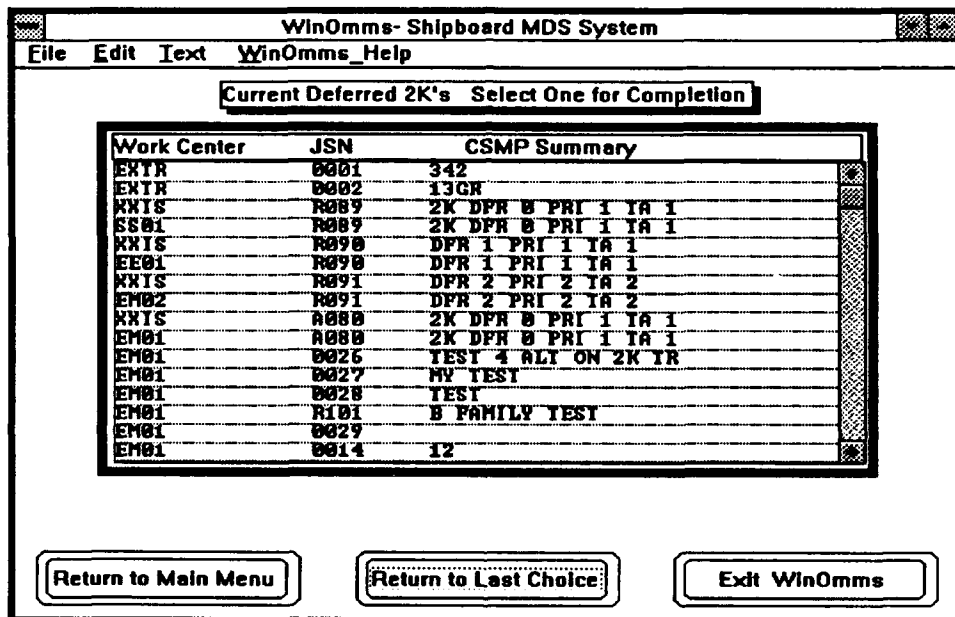


Figure 5. Selecting a Deferred OPNAV 2K

7. Completing a Deferred OPNAV 4790-2K

Figure 6 on the next page shows the screen that is presented upon the user's action of selecting a deferred action to complete. This screen was designed to facilitate easy and rapid data entry. Only 6 blocks require filling in. They are presented in the center of the screen with the remainder of the filled in form shown in the background.

This data entry screen is consistent with all Winomms data entry screens. The cursor is placed in the first block that requires filling in, in this case "Action Taken", and that block is high-lighted to present a clear view of where the data is to be entered.

Once the user enters the completion data he clicks on the "Enter Completed 2K into System" button. This action writes the newly entered data into the database, updating the appropriate record.

WinOmms- Shipboard MDS System

File Edit Text WinOmms Help

Enter Completed 2K into System **OPNAV 4790 2K** **View Completed 2K just Entered**

02535 KRIS 1988 0400000 1012100
 0001527 51853D, HOLIMETE 2 2 2 2
 2610D
 Plan Date: 872315
 Inspec: Inspec Inspec Inspec Inspec Inspec Inspec
 NPM: Exp: 8002 Date: 872315 Code: 8001 Condition: 8001
 Act: 872315 Comp Date: Act Date: Inspec: Note:

REMARKS: 02K 2 PRI 2 PA 2

COMPLETED ACTIONS SECTION

Action Taken **SPPH Total**
Comp Date **Active Maint**
Trouble Iso **Hcr Reading**

0000 0000 Test Winomms Prototype 6/21
 0000 0000 0000 0000 0000 0000
 0000 0000 0000 0000 0000 0000
 0000 0000 0000 0000 0000 0000
 0000 0000 0000 0000 0000 0000

Main Menu **Choose Another 2K to Complete** **Exit WinOmms**

Figure 6. Completing an OPNAV 2K

IV. DESIGN AND IMPLEMENTATION

A. DESIGN GOALS AND ISSUES

1. Winomms Design Goals

The design goal of this research was to prototype a logically arranged, easy to use and learn interface for an underlying database system. The prototype is developed in the Microsoft Windows application environment which allows extensive use of colors, pull-down and pop-up menus, and mouse selectable options. Since the development software, TOOLBOOK, supports all of these attributes the design issues were based on what would be the most efficient and consistent method to implement the desired features of this graphical interface.

2. Winomms Design Issues

The overall scheme of this interface is based on a hierarchical structure that closely emulates the hierarchical structure of the existing menu driven system. This means that the existing interface was based on a series of levels that the user navigated through. The top level represented the main menu and from there, the user navigated downward branching out laterally when required to obtain information necessary to the completion of his task. TOOLBOOK proved to be well suited to emulating this design scheme due to its hierarchical structure. Each required user screen is represented as a page

of the book "Winomms". Since pages are on the same hierarchical level, lateral branching to access required information was easily accomplished. Hierarchical levels on any page are represented by the pages background and the various button, fields, and graphical images.

A key design consideration that is well supported by TOOLBOOK was that much of the time consuming navigation to other screens in the existing system was eliminated by the use of pop-up windows and menus. This design feature allowed for example a context specific help window to be hidden on the page the user is currently on. To view this help window the user only has to point and click with the mouse device. This type of design is easier to use and requires a fraction of the time that the menu based system does to navigate to a help screen then navigate back to where the user was.

The first stage of designing Winomms was a through familiarization with the existing system, MicroOmms. As noted above, this existing system is designed on a hierarchical series of layers that access various standard database files. One important objective of this research was to try to design a new interface that co-existed with the underlying structure and would not require any modification to the underlying structure. This consideration required a considerable amount of research into the numerous areas of the current interface that accessed the databases. The scope of this research was then limited to following one particular functional area (the Maintenance Data System area) to its completion.

Once familiar with the existing interfaces structure and its relation to the underlying databases the next stage was to determine the best way to implement a graphically based interface that provided all of the capabilities of the old system while meeting the standards of this research.

B. IMPLEMENTATION ISSUES

1. Overview of Implementation

The best way to implement the design goals of this research using TOOLBOOK was to divide the current system into functional areas. Each major functional area would be represented on the main menu. Since a TOOLBOOK application is analogous to a book, the main menu would be the equivalent of the books table of contents. From this starting point the user can then select one of the books "chapters" which would be the starting page of a functional area.

This concept of a book was driven by the use of TOOLBOOK's page feature. A page can contain any number of text fields, record fields, and graphic images or buttons to facilitate navigation. Additionally, a pages background is homogeneous and can be copied, providing a template that can be used on any page in the book. This TOOLBOOK feature was used to allow design of uncluttered screens that contain only relevant information to the area the user is working in. Additional information is then displayed on subsequent pages that are all identical.

With the TOOLBOOK page as the primary structure in this application two other key structures were also used: the scrollable recordfield and a simulated index card. The recordfield is a TOOLBOOK feature that will allow the input of record field values from a database. The scrolling feature is an additional tool that allows viewing of a large number of text lines or record field information a few lines at a time. By implementing use of this features it was possible to read in an entire database on one screen yet present only several lines at a time to the user. By simple use of the mouse device clicking on the down arrow of the scroll bar, additional data is scrolled into the view window.

Since this application involves numerous options and databases that may require accessing such as looking up part numbers, simulated index cards were designed to represent standard 3 by 5 inch index cards. The purpose of this feature is to provide a familiar and easy view of user options that can be selected with the mouse device. Implementing this feature required decomposition of the user option screens on the current interface. This was not detrimental to the experienced user since this decomposition was carefully done to preserve functional areas of the system and each grouping of index cards represents the same logical path of choices available on the current system. The result of implementing this design feature is that while retaining all of the previously available options, the user now has a much enhanced view of available choices to make.

2. Implementation Details

a. Screen Layouts

The dominate theme of this interface research is to provide a system that is simple and easy to use. This implementation issue can be accomplished through the reduction or elimination of excessive keystrokes, and the elimination of cluttered screens. To accomplish this goal, Winomms makes extensive use of buttons and graphic images that are mouse selectable and therefore require no keystrokes at all. Additionally, the screens have been carefully arranged to present only the information required for the function being carried out at any time.

TOOLBOOK allows the attaching of a script to any object on the screen. This can include the page itself or any graphical image, text field, or button. This script is activated when the object is selected by clicking on it with the mouse pointer. The action initiated by the script can range from showing a pop-up window of further choices or a context-sensitive help screen to allowing navigation to another page in the book.

The decision was made in this implementation to accomplish this simplicity issue by providing clearly labeled graphic images for the user to select with the mouse device. Although it would have been possible to provide text entry boxes that would more closely resemble the original interface, the use of graphic images is more consistent with the Microsoft Windows environment that relies extensively on use of icons. The design decision to use

graphic images and buttons for selections maintains the consistency of a Microsoft Windows application and also provide a system that is easy to use.

A key consideration in this phase of the interface design was the arrangement and physical size of the buttons, images and text on any given screen. To prevent overcrowding of the screens it was often necessary to decompose some of the current interfaces screens into smaller modules. This design decision led to a much cleaner look where the user would not be overwhelmed with data to look at. Additionally, with the exception of an "EXIT" button, only functions directly related to the screen in view were placed on that screen. To allow quick exiting of Winomms or a return to the main menu, it was necessary to place navigation scripted buttons on every page of the application. To maintain consistency, these buttons have the same look and are located in the same relative place on every page.

The decision on whether to use a button or graphical images in TOOLBOOK is primarily a decision of what caption or text is required to accurately convey the meaning or function that button or icon represents. Since buttons and graphical images can be of any size from very small to full page, the deciding factor is the amount of text required. A TOOLBOOK button can only contain a caption of one line of text and is limited by the horizontal length of the button. A graphical image in which a TOOLBOOK text field is embedded is limited in textual content only by the desired size of the field.

Based on the considerations listed above, this application used buttons whenever one or a few words would accurately convey the meaning of the button ,otherwise graphic images were used.

b. Navigating in Winomms

All navigation in Winomms is initiated with the point and click technique using a standard mouse device. Allowable navigation in any given screen is context sensitive. This scheme reflects a philosophy that is task specific. Once the user has committed to a particular task area to work in, he should be allowed to navigate forward or backward if the task has multiple pages, exit the task area to a previous choice, return to the main menu, or exit the system. This decision led to the use of navigation buttons or graphic images. In multiple page documents the best design was to use "Next Page" and "Previous Page" buttons that allow instant navigation forward or backwards. This particular implementation was chosen because TOOLBOOK includes pre-defined scripts that allow navigation to the next or previous page in the book currently being used.

In the case of the index options choice cards described in the Implementation Overview section of this thesis, allowing the user to go back to a previous choice required designing a method to allow the user to know what the last choice he made was, and a method of navigating easily back to that last choice. This was implemented by using the TOOLBOOK text field abilities to read in a line of text that has been mouse device selected. When the

user clicks on a choice from one of the index cards, that choice is written into one of five "Previous Choices" boxes aligned on the left side of the screen. This allows a history of past choices available for viewing. The design of the "Past Choices" boxes required navigation scripts that would clear the current index option card being displayed and show the card that contains the past choice the user selected. The purpose of this feature, driven by the design goal of ease of use, is to allow easy reversal of errant or unwanted choices and to give the user a feeling of control over the system he is using.

V. APPLICATION SUMMARY

A. FACTORS AFFECTING PERFORMANCE

The performance of this application depends largely on the hardware configuration of the system it is run on. This is due to the graphics based nature of the Windows environment. Toolbook, the development software used in this research, can only run in the Windows environment. Toolbook also makes extensive use of graphics. These features are very CPU intensive since all of the graphics used are bit-mapped. Additionally, graphics images require large amounts of storage space which increases the number of secondary storage accesses.

B. TYPICAL APPLICATION PERFORMANCE

This application requires a minimum of an INTEL 286 CPU, 20 Megabyte harddrive and two Megabytes of system RAM. With this configuration, loading in the "Winomms" application from the Windows Program Manager will take 3 minutes. Selecting buttons that cause navigation to other screens typically takes 4 to 6 seconds and data retrievals from the database take 12 to 45 seconds. As this configuration is the minimum requirement, these typical performance times are the worst case situations.

Use of the INTEL 386 CPU with 4 Megabytes of RAM and a fast 40 Megabyte hard drive significantly increases performance of this application. Loading "Winomms" from the Program Manager will take 9 to 18 seconds, screen navigation 1 second, and typical database retrieval 3 to 4 seconds. The TOOLBOOK application software has recently been update and the new version 1.5 promises significant increases in system application performance.

Performance of this application running on a standard configured 386 pc compared to a standard DOS based database running on a Zenith 248 pc is superior. Although this application will run on a Zenith 248, it would be unacceptably slow.

VI. CONCLUSION

A. SUMMARY OF RESEARCH

Winomms was designed to provide the typical database user with a more efficient method of accomplishing daily tasks. These tasks include searching, retrieval and update functions on a relational database. The cluttered menu-driven screens have been replaced with a graphical front-end. This research produced a prototype Windows application which runs in the Windows environment.

The Windows environment provides many benefits, especially to the novice user. Complicated command line syntax is eliminated. The user executes the application by clicking on the appropriate icon with a mouse device. This provides a far easier and more efficient method for shipboard personnel to perform the maintenance data tasks required on modern Naval vessels.

B. REVIEW OF RESULTS

The TOOLBOOK development software and the Windows environment provided a powerful set of tools that enhanced rapid prototyping of this application. Some limitations that exist, however, will require addressing before a completely developed system is feasible.

- The version of TOOLBOOK used for this research, 1.0 is annoyingly slow. A new version has been released and promises greatly increased speed.
- TOOLBOOK scripts are all interpreted with a resulting performance decrease. A compiled script would be significantly faster.
- TOOLBOOK is still a relatively new product. There is little to no information from third party sources on using TOOLBOOK.

C. FUTURE DEVELOPMENTS AND UPGRADES

Improvements to this application could be realized in two primary ways. The new version of TOOLBOOK is now available and offers greatly increased speed of development and execution. This application could be ported into the new TOOLBOOK and fine tuned for a performance increase. There are currently no facilities to print out maintenance documents on the existing application that this research addressed. Finished documents are laboriously formatted onto a paper tape of 5 bit codes that are then printed out on special printers at maintenance facilities. This could be implemented via TOOLBOOK, providing more flexibility than the current paper-tape print-out used in the Fleet.

LIST OF REFERENCES

1. **Shneiderman, Ben "Designing the User Interface" Addison Wesley Publishing Company 1987**
2. **Giannotti, G. CDR and Duffy, Kevin LT. "ARGOS: Design and Development of Object-oriented, Event-driven, Multimedia Database Technology in support of the Paperless Ship", Masters Thesis, Naval Postgraduate School, Monterey, CA, December 1988**
3. **Pierce, Joseph R. " Toolbook Companion" Microsoft Press 1990**

BIBLIOGRAPHY

Swain, Roy E., *The User-System Interface for Computerized Products*, Virginia Polytechnic Institute and State University.

Barth, Paul S., *An Object-Oriented Approach to Graphical Interfaces*, ACM Transactions on Graphics, July 1986, Volume 5 Number 3.

Smith, Joseph R., *The Windows Companion*, Microsoft Press 1990.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
Cameron Station
Alexandria, Virginia 22304-6145
2. Dudley Knox Library 2
Code 52
Naval Postgraduate School
Monterey, California 93943-5002
3. Office of Research Administration 1
Code 012
Naval Postgraduate School
Monterey, California 93943-5002
4. Chairman, Computer Science Dept. 1
Computer Science Department
Naval Postgraduate School
Monterey, California 93943-5002
5. Chief of Naval Research 1
800 N. Quincy Street
Arlington, Virginia 22217-5000
6. Center for Naval Analyses 1
4401 Ford Avenue
Arlington, Virginia 22302-0268
7. Naval Ocean Systems Center 1
271 Catalina Boulevard
San Diego, California 92152
8. Curriculum Officer 1
Computer Technology Program, Code 37
Monterey, California 93943-5000
9. C. Thomas Wu (Code CS Wq) 2
Computer Science Department
Naval Postgraduate School
Monterey, California 93943-5000

- | | | |
|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
| 10. | Division Head
MDS Division
Data Systems Department
Naval Weapons Station
Concord, California 94520-5000 | 2 |
| 11. | Robert Calogero
Director SEA CEL-PA
Logistics Policy and Appraisal Division
Naval Sea Systems Command
Washington, D.C. 20362-5101 | 2 |
| 12. | Clifford G. Geiger
Deputy Chief Engineer - Logistics
Naval Sea Systems Command
Washington, D.C. 20362-5101 | 1 |
| 13. | Peter Davenport
SNAP III Project Manager
COMPSPAWASYSCOM
Code PMW 164-1
Washington, D.C. 20363-5100 | 1 |
| 14. | Philip B. Stiles
Naval Sea Systems Command
Technical Data Division of the
Chief Engineer for Logistics Directorate
Washington, D.C. 20362-5101 | 2 |