A MICROCOMPUTER SYSTEM FOR TARGET INFORMATION IN THE FIRE SUPPORT COORDINATION CENTER: A DATA BASE APPROACH

Ronald J. Coulter

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

A MICROCOMPUTER SYSTEM FOR TARGET INFORMATION
IN THE FIRE SUPPORT COORDINATION CENTER:
A DATA BASE APPROACH

by

Ronald J. Coulter

June, 1981

Thesis Advisor: Lyle A. Cox, Jr

Approved for public release; distribution unlimited.

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Microcomputer System for Target Information in the Fire Support Coordination Center: A Data Base Approach.		5. TYPE OF REPORT & PERIOD COVERED Masters Thesis June, 1981
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Ronald J. Coulter		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE June 1981
		13. NUMBER OF PAGES 145
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)
Microcomputer, user interface, target information, data base, FSCC (fire support coordination center), fire support coordination, Marine Corps, UCSD Pascal, amphibious operations

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
This thesis presents the specification, design and implementation of a prototype microcomputer system for the target information section of the Marine Corps fire support coordination center. Currently, the target information section uses a series of index cards, handwritten list, acetate covered battle maps and grease pencils to perform the target information functions.

The thesis examines and analyzes these functions in detail and

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601 |

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

proposes a solution in the form of a system, data base and interactive user design. The resultant Microcomputer System for Traget Information (MISTI) employs an ALTOS Z-80 microcomputer, the UCSD Pascal operating system, a user friendly interface and data base technology. It is proposed as an interim system until the Marine Integrated Fire and Air Support System (MIFASS) becomes operational.

Approved for public release; distribution unlimited.

A Microcomputer System for Target Information in the Fire Support Coordination Center: A Data Base Approach

by

Ronald J. Coulter
Lieutenant Colonel, United States Marine Corps
B.S., College of the Holy Cross, 1964
M.A., Pepperdine University, 1975

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 1981

ABSTRACT

This thesis presents the specification, design and implementation of a prototype microcomputer system for the target information section of the Marine Corps fire support coordination center. Currently, the target information section uses a series of index cards, handwritten lists, acetate covered battle maps and grease pencils to perform the target information functions.

The thesis examines and analyzes these functions in detail and proposes a solution in the form of a system, data base and interactive user design. The resultant Microcomputer System for Target Information (MISTI) employs an ALTOS Z-8Z microcomputer, the UCSD Pascal operating system, a user friendly interface and data base technology. It is proposed as an interim system until the Marine Integrated Fire and Air Support System (MIFASS) becomes operational.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

AAA----- Anti-Aircraft Artillery Target

ARTY---- Artillery Target

ASCII--- American Standard Code for Information Interchange

ASIS---- Amphibious Support Information System

BDA----- Battle Damage Assessment

CBAT---- Counter Battery Target

CODASYL- Conference on Data Systems Languages

CP/M---- Command Program/Monitor; operating system built by Digital Research for Microcomputers

CRT----- Cathode Ray Tube

DASC---- Direct Air Support Center

DBMS---- Data Base Management System

DTG----- Date-Time Group

FASC---- Fire and Air Support Center

FDC----- Fire Direction Center

FMFM---- Fleet Marine Force Manual

FORT---- Fortification, Bunker, Hardened Site Target

FSC----- Fire Support Coordinator

FSCC---- Fire Support Coordination Center

IBM----- International Business Machines Corporation

INST---- Installation, Buildings Target

K------- Kilobyte (1024 bytes)

MAGTF--- Marine Air-Ground Task Force

MIFASS-- Marine Integrated Fire and Air Support System

MISC---- Miscellaneous Target

MTACCS-- Marine Tactical Command and Control Systems

NGF----- Naval Gunfire

NWP----- Naval Warfare Publication

OP------ Observation Post Target

SACC---- Supporting Arms Coordination Center

SAM----- Surface-to-Air Missile Target

SEAD---- Suppression of Enemy Air Defense

TARBUL-- Target Bulletin

TERR---- Terrain Target (Hilltop, Road Junction, Field, etc.)

TGT----- Target

TIC----- Target Information Center (Navy)

TIO----- Target Information Officer

TIS----- Target Information Section (Marine)

UCSD---- University of California at San Diego

VEH----- Vehicular Target (Truck, Jeep, etc.)

ACKNOWLEDGEMENTS

The author would like to acknowledge the assistance and support of LtCol Don Sortino, USMC, Assistant Fire Support Coordinator of the Second Marine Division, Camp Lejeune, North Carolina in providing current information on target information procedures and Capt C. M. Miller, USMC, of the Marine Corps Tactical Software Support Activity (MCTSSA), Camp Pendleton, California for providing timely and detailed information on the Marine Integrated Fire and Air Support System (MIFASS).

The guidance and suggestions provided by thesis advisor, Professor Lyle A. Cox, Jr. and first reader, Professor Dusan Z. Badal are deeply appreciated and their encouragement and confidence were a constant source of optimism. Capt Jeffrey A. Neufeld, USMC and Cpt A. Ross Strickler, U.S. Army, provided technical assistance in the finer points of the UCSD system and their assistance proved timely and valuable.

I. INTRODUCTION

A. THE PROBLEM

More and more of the applications of modern amphibious warfare, from real-time combat systems to the data bases that control the men, materiel and resources needed to wage war, have turned to computerized solutions. The products of the technological explosion have enabled the Navy-Marine Corps amphibious team to do more, to do it faster and to do it with a degree of efficiency and accuracy previously unobtainable.

This evolution of modern technology has not yet reached the Marine Corps tactical command posts established on the beachhead. The target information section of the landing force fire support coordination center (FSCC) plays a signficant role in the conduct of effective coordination of tactical air, artillery and naval gunfire support on targets of high priority. Yet the target information officer and his staff accomplish their important task by the use of index card files, cross-reference files, hand written lists of targets and colored grease pencils on acetate-covered tactical maps. This method is time consuming, slow in response to inquires about target information, tedious and difficult to maintain in a current status and does not provide information in a sufficiently timely and accurate

manner. It is 40 year old technology in the age of computers.

The requirement to automate many of the functions of the tactical command post has been identified and the command post of the future is being planned for and developed now. Until it arrives, there is a need to provide an interim capability to the landing force. An automated solution to the target information function will simplify the task of the target information section considerably, will provide rapid, accurate and timely target information to the members of the FSCC, and can be made operational now, five full years before the planned introduction of the computerized command post.

This thesis contends that the automation of the target information function is necessary to improve the operational capability of the landing force FSCC and that implementation of a suitable and effective target information system is possible. This thesis will prove this contention by implementing and designing a working prototype which will increase operational effectiveness immediately as well as provide a testbed and learning model for the future automated command post. The prototype will be designed to perform all the duties and functions of the target information section as currently stated in doctrinal publications. The interim system will hopefully contribute to the development of the future system and identify areas

of concern and improvement before the future Marine Corps system becomes operational.

B. BACKGROUND

An important aspect of amphibious fire support coordination (the planning and execution of tactical air, artillery and naval gunfire support so that targets are adequately covered by a suitable weapon or group of weapons) is the function of target information. One of the major duties of the fire support coordinator, that member of the landing force staff responsible for coordination of fire support, is to ensure that the fire support coordination center receives and disseminates available target information to all staff sections and commands requiring the information. He also must work closely with the target information officer and the commander and his staff in the selection of targets and assignment of classification and attack priorities.

Target information is the direct application of combat intelligence to fire support and is a key to the proper employment of supporting arms in conjunction with each of the plans of the amphibious operation. Effective fire support coordination and the planning of amphibious operations generate a continuing requirement for target acquisition, dissemination, evaluation and recommendation for attack.

To accomplish this important task, the commander of the amphibious task force assigns a target intelligence officer to the supporting arms coordination center (SACC). This officer operates the target information center (TIC) and works closely with the air intelligence officer, the landing force targeting representatives and the supporting arms coordinator. The commander of the landing force has a target information officer (TIO) who operates the target information section (TIS) as an integral part of the landing force fire support coordination center and a target intelligence officer who functions in the landing force intelligence center.

The Navy staff uses a computerized target information system which is part of the shipboard Amphibious Support Information System (ASIS) and maintains the list of targets as part of a data base. Target information operations in the SACC are thus computerized and, while the ASIS target system is not the most modern of data base systems, it is efficient, effective and fast. When the functional responsibility for maintaining targets is passed ashore to the landing force TIO, the computer system is replaced by an index card filing system, which, while effective, is neither fast nor efficient by comparison. Additionally, the index card system lends itself to inaccuracies and omissions in target data, particularly when the information must be maintained in a timely manner. The tactical requirement for

accurate and timely target information is no less critical or important when the landing force is on the beach, yet the system to accomplish this task is antiquated and cumbersome.

The staff of the TIS manually transfers the target information data contained in the ASIS data base to 5 by 8 inch target cards. After duplicating the entire target file, the TIS must construct a cross reference file to list the target by grid location and a cross-index file to keep track of certain types of targets. In addition to the target cards, the TIS also makes up lists of particular categories of targets which may be of interest or value to members of the FSCC.

The TIS obtains intelligence information from landing force and supporting arms agencies, converts this to target information and enters the information into the target card files. The information is made available to the supporting arms representatives in the FSCC and, based on the TIO's recommendations, a decision is made when and how to attack a particular target. Results of attacks on targets, front line reports and intelligence information are used to refine the target list card and delete or deprioritize those targets that present a diminished threat to the landing force.

Access to specific information from the target list (for example, more than one category of the cross-index files) requires physically searching through each list and constructing sub-lists to determine the appropriate

information. The constant availability of timely and accurate target information is required for the effective employment of supporting arms and planning of fire support. The TIS plays a key role in providing this information and the constant process of adding to the target list, selecting targets for attack and deleting targets once neutralized is performed by the TIS staff using the target card file.

C. INTEGRATED FIRE AND AIR SUPPORT SYSTEM

One of the most complex aspects of modern amphibious warfare is the control and coordination of supporting arms particularly in the transition of responsibility from the Navy in amphibious ships to the Marine Corps combat units ashore. The grease pencils, map boards and field radios that have served Marines so well since the days of Guadalcanal will, in the future, be eclipsed by the automated system called the Marine Integrated Fire and Air Support System (MIFASS).

MIFASS is part of the Marine Corps integrated command and control system called MTACCS (Marine Tactical Command and Control Systems), a collection of eight major systems which will give the Marines a capability of exercising real-time command and control of combat forces in the post-1980 time frame. MIFASS is designed to perform the functions of the fire support coordination center, (FSCC) the direct air support center (DASC) and, to a degree, the

artillery fire direction center (FDC) at one central location called the Fire and Air Support Center (FASC).

It is a distributed processing system in which microcomputers control interactive display devices, manage data bases, perform computational tasks and drive printers to provide hard-copy records of messages and operator decisions. It is currently in full scale engineering development with an initial operational capability planned for the 1986-1987 time frame. MIFASS addresses the requirement for target information by providing the TIO with a digital display device which will have both a graphical representation of the target on a battle map and a video screen for alphanumeric display of target information.

D. NATURE OF THE PROBLEM

An automated solution to the target information function will not be realized until the introduction of the MIFASS computers into the Fleet Marine Forces. Until such time as the system is delivered, the target information function of the FSCC is tied to the current doctrine and the target card filing system.

In this report, an interim systems solution to the problem of automating the target information function of the FSCC is presented. It computerizes those basic functions of the TIS in a simple, inexpensive and effective manner. It simplifies the tasks of the TIS, provides a mechanism for

rapid and accurate retrieval of target information and could improve the operational capability of the FSCC.

E. NATURE OF THE SOLUTION

The amount of target information that needs to be processed is sufficiently small that a microcomputer is the most suitable piece of hardware for implementation. The current versions of microcomputers are very versatile with efficient operating systems, various input/output media including video terminals, inexpensive and relatively portable secondary storage media (floppy diskettes and cassettes), high level language programming capabilities and even scaled down versions of data base management systems. Thus, the technology in hardware as well as software currently exists in the commercial marketplace and it is possible that a practical system can result from efficient and careful design and implementation.

The design task is broken down into three distinct parts, each of which is influenced by the overall design characteristics and is individually addressed in separate chapters.

The design of the physical and logical data base is influenced by the desire to have a simple yet sufficiently informative data model, a rapid, real-time response and a restricted, single application system. The system design is influenced by the microcomputer environment which restricts

the user both in main memory space and the speed of access to secondary storage and the requirement for an effective interactive system for a non-sophisticated user.

The design of the software to implement both the data base and the system is overwhelmingly influenced by the requirement that the system support real-time, interactive processing of a casual, non-programmer. Termed "Marine proof" in the vernacular, it requires a sophisticated interface employing user friendly dialogue techniques to ensure that the operation is simple and efficient. For this reason, and to facilitate system portability, a microcomputer compatable high level programming language is employed in implementation.

In order to better identify the user environment and to obtain an understanding of the functions of target information, the next chapter describes the mission and the current procedures of the target information section. It is from this information that the system characteristics were developed and the design based. The information was obtained from Navy and Marine Corps doctrinal publications as well as current operating procedures of a Marine Division target information section. Chapters III through VI develop in detail, the reasons for the parameters selected and the decisions made in the design of the overall system, the logical and physical data base and the applications software.

Chapter VII addresses the implementation of the system and further implications of system application in the Marine Corps, as well as tactical employment and interface with current and future systems. Conclusions and recommendations are included in the last chapter.

The source code listing, which has been developed as a result of this thesis, has been published as a Naval Postgraduate School technical report entitled A Prototype Program for Target Information (NPS52-81-007). A data dictionary and an example of the system interface are included in the appendices. A bibliography of applicable references and a list of abbreviations used are also included.

II. TARGET INFORMATION PROCEDURES AND EMPLOYMENT

A. GENERAL

A precise understanding of the duties of the target information officer and procedures used by the target information section is required before detailed requirements for an automated target information system can be stated. This chapter is devoted to that purpose. It discusses and examines in detail the doctrinal duties and functions of the target information officer and the current procedures for executing these functions.

The target information officer is a member of the fire support coordination center (FSCC). He and his staff provide target information to the fire support coordinator so that effective employment of supporting arms is driven by timely and accurate target intelligence. He works directly with the artillery representatives, the air officers and the naval gunfire support officers in diseminating appropriate target information and obtaining surveillance information. He assigns battle damage assessments for attacked targets and further refines the target list.

His relationship with both the amphibious task force target intelligence officer and the landing force target intelligence officer is extremely important since it is from these sources that he obtains the target intelligence which generates the target information.

B. DUTIES OF THE TARGET INFORMATION OFFICER

The TIO is a Marine Corps officer who performs his duties under the staff cognizance of the fire support coordinator (FSC) and works closely with the landing force operations and intelligence sections. The primary doctrinal publication for the Marine Corps is Fleet Marine Force Manual (FMFM) 7-1 (Fire Support Coordination) which outlines his duties as follows:

1. Keeping the FSC and the other fire support representatives in the FSCC informed of the status of targets.

2. Ensuring that pertinent target intelligence is posted on the FSCC target and/or situation maps.

3. Preparing and maintaining target file cards.

4. Entering target attack evaluations and surveillances on the target cards.

5. Supervising the operation of the target information section (TIS) of the FSCC.

6. Preparing the landing force list of targets or the Marine air-ground task force (MAGTF) target list for promulgation by the operations officer. The FSCC will provide targets, to include their classification and priorities, which are to be included in the target list, target bulletins and/or lists of targets.

7. Preparing and releasing target bulletins when control of the target list has been passed to the commander landing force or when the MAGTF is engaged in land warfare.

8. Keeping the target intelligence officer advised of target information available through supporting arms sources.

C. FUNCTIONS OF THE TARGET INFORMATION SECTION

The functions of the TIS are oriented to the requirements of the supporting arms (air, naval gunfire and artillery) in the preparation of fire support plans and the command requirements for target information. The TIS uses all of the available intelligence gathered by the agencies of the amphibious task force and the landing force. These agencies include landing force and amphibious task force target intelligence sections and intelligence agencies of the supporting arms.

The TIS is responsible for recording all target information, analyzing this target information, maintaining records and making recommendations of targets which are appropriate for attack. FMFM 7-1 lists the following functions of the TIS:

1. Maintaining required target and situation maps.

2. Maintaining target cards and target files, including cross-indexed files of target information.

3. Consolidating, evaluating and displaying target information.

4. Recommending classification and attack priorities to the FSC.

5. Collecting from all agencies and sources, any information pertaining to the results of attack on targets by the supporting arms.

6. Consolidating and evaluating results of attacks by the individual supporting arms and the methods of attack, and recommending additional measures that appear necessary from the overall results and analyses.

7. Coordinating on all matters with the landing force target intelligence officer and the artillery unit intelligence officer for target and counterfire information and correlation of records and files.

8. Maintaining current counterfire target lists to include counter-mortar, counter-battery and SEAD (suppression of enemy air defense) lists and providing this information to the supporting arms representatives as well as ensuring dissemination to the landing force as a whole.

9. Preparing and disseminating target bulletins (TARBUL's) after control of the target list has been passed ashore.

10. Maintaining a nuclear and chemical target folder to assist in the selection, evaluation and planning of attack by supporting arms utilizing nuclear and chemical munitions.

The composition and organization of the TIS varies with the FSCC level but typically at the landing force level it consists of one officer (TIO) and from one to three enlisted personnel. Personnel are usually trained in target intelligence, supporting arms capabilities and limitations, organization, fire support coordination principles and communications.

While the functions and duties of target information personnel are determined by the doctrinal publications, the actual procedures to accomplish these functions will differ slightly from one organization to another, however these variations are minor.

D. TARGET INFORMATION RECORDS AND FILES

The records and files of target information consist primarily of situation maps, target file cards, target lists and cross-indexing files at the landing force level. They are the tools used to catalogue, analyze and disseminate target information.

The target map provides a visual reference of targets appropriate for attack by supporting arms. The friendly situation map contains all information pertinent to supporting arms operations and typically includes objectives, front lines, fire support control measures, unit boundaries and unit locations.

The bulk of the record keeping involves the target file card. The file of 5 by 8 inch cards contains a separate target card for each known or suspected target both by target number and by grid coordinates. Figure 1 is an example of a target card. Information appearing on the target card includes the following:

- target symbol (conventional map symbol)
- target number
- target classification
- attack priority
- target location (grid coordinates)
- target elevation in meters
- map reference
- target description
- assignment of supporting arms attack means
- source and date of target information
- photograph number and grid location
- remarks of additional significance

Security Classification

TARGET CARD

Target Number NA8157	Map Symbol ? ┤├ 12"	Grid Coordinates 3915 - 7650
Target Classification A		Elevation 350 M
Target Priority I		Map Number

Description 12 INCH ENEMY COAST DEFENSE ARTILLERY INSTALLATION OF REINFORCED CONCRETE CASEMATE FACING SW
Assigned to: ( ) Air (✓) Naval Gunfire ( ) Artillery ( ) Other Support Weapons:
Source and Date of Information API DTG 131452 JUL 63
Photo Number VP 17-1314152 Photo Coordinates 3915 - 7650
Remarks: ATTACK WITH HEAVY CALIBER NGF USING ARMOR PIERCING OR SHELL HE W/DELAY FUSE, DIRECT FIRE

Security Classification

FRONT OF TARGET CARD

Security Classification

RECORD OF MISSIONS

Date/Time Group	Firing Unit	No. Rounds and Type	Damage Reported	Damage Assessed
141415Z	CLG-4	12 6"	DESTROYED	NEUTRALIZED
150930	CLG-4	24 6"	DESTROYED	DESTROYED

Security Classification

BACK OF TARGET CARD

Figure 1. Example of a Target Card.

The target cross-index file consists of one card or list for each type of target (e.g., counter-battery, armor, SEAD, fortification, etc.). Each card or list typically includes only the target number of each target, it´s priority, the recommended method of attack and the final disposition of the target.

In a typical amphibious operation, the landing force usually operates with a maximum of approximately 200-300 targets. With a separate target card for each target by target number as well as by grid coordinate and a cross-index card for the 10 to 15 target types, the target file can easily exceed 500 cards. An example of a Marine division target card file organization is illustrated in figure 2.

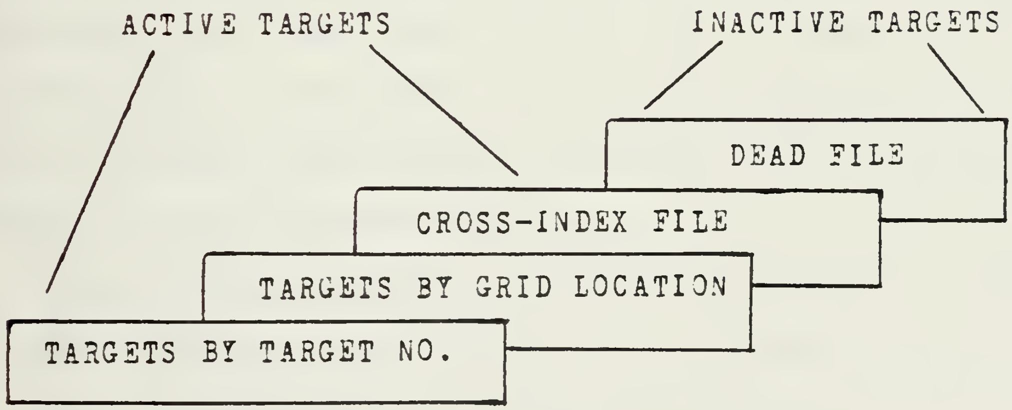


Figure 2. Target Card File Organization.

E. THE TARGET LIST

A semantic distinction must be made between the "target list" and the "list of targets". The "target list" is a collection of targets which is maintained and promulgated by the senior echelon of command. There is only one "target list". It contains targets which are pertinent to the landing force as a whole and which are to be taken under attack by supporting arms. A "list of targets" is maintained at any echelon of command and includes those confirmed, suspected or possible targets for information and planning purposes as well as for possible attack by supporting arms. The "target list" is a subset of the "list of targets".

Subordinate units use the target list as their basic source of targets and also include targets that have a significant but specific or "short-life" value to their operations in their unit list of targets. As an illustration, a battalion would only include those targets from the landing force target list which were located in or adjacent to their zone of action.

Targets can be further described as active or inactive. An active target is one which is on the target list or list of targets and presents a bonafide current or future enemy capability to interfere with operations. An inactive target is one which has been overrun by friendly forces or destroyed by supporting arms or has shown no activity for 72 hours and no damage assessment has been made, although these

latter targets are inactivated with caution. The inactive targets are placed in a deadfile and reactivated if necessary. Figure 3 depicts the target list terminology.

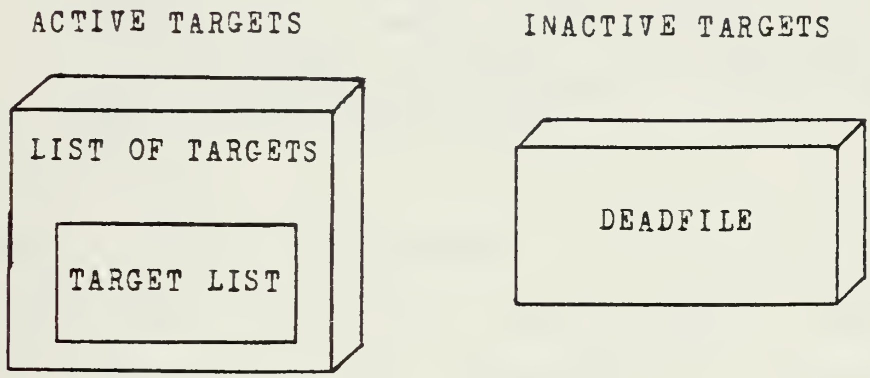


Figure 3. Target List Terminology.

F. TARGET CLASSIFICATION

Targets are classified by the effect which their existance or elimination may have on the amphibious task force and by restrictions imposed by the commander on the attack of certain targets.

The primary doctrinal publication for amphibious warfare, NWP 22-2 (Supporting Arms in Amphibious Operations) list the following target classifications:

Class A...Targets that threaten ships, aircraft, minesweeping and underwater demolitions operations.

Class B...Targets that threaten assault forces in the ship-to-shore movement and assault of the beach.

Class C...Targets that threaten or oppose landing force operations afterlanding or affect the ability of the enemy to continue resistance.

Class D...Targets that will not be fired on prior to D-Day.

Class E...Targets that must not be destroyed (unless specific orders for such destruction are issued by the amphibious task force or landing force commander) either because of probable future use by our own forces or for humanitarian reasons. These installations may be neutralized, harassed or interdicted if prior approval is obtained from the commander imposing the restrictions.

G. TARGET PRIORITY

The target information officer, in coordination with the target intelligence officer, the fire support coordinator and the supporting arms representatives reviews and recommends the assignment of attack priority. The target priority is established to determine the sequence of attack and/or the effort to be allocated to a given target. The TIO establishes the priority based on the target´s effect on the accomplishment of the landing force mission and its relative importance as compared to other targets.

FMFM 7-1 lists the following target priorities:

Priority I.....Targets capable of preventing the execution of the plan of action by the landing force and its elements.

Priority II....Targets capable of immediate serious interference with the plan of action of the landing force and its elements.

Priority III...Targets capable of ultimate serious interference with the plan of action of the landing force and its elements.

Priority IV....Targets capable of limited interference with the plan of action of the landing force and its elements.

H. THE TARGET BULLETIN

In order to maintain up-to-date target information records, it is essential that reports of the discovery of new targets and the analysis of supporting arms attacks on existing targets be reported to the appropriate units. The TIO evaluates and consolidates reports of target information and supporting arms battle damage assessment (BDA) and prepares a target bulletin (TARBUL). Upon approval, it is released to interested commanders of higher, lower and adjacent elements of the amphibious task force.

The TARBUL is normally transmitted over existing teletype or radio circuits and typically adds new targets to the target list (giving the target number, location, elevation, priority, classification and description), gives damage assessment to existing targets which have been attacked by supporting arms, cancells targets from the target list (relegating them to the deadfile) and reactivates previously cancelled targets. TARBUL's are serialized and issued on an as-needed basis.

I. OPERATIONS OF THE TARGET INFORMATION SECTION

While the target information section is heavily involved in the early phases of the operation, the most important with respect to this thesis occurs during the preparation of the objective, ship-to-shore movement and operations ashore.

The target list is initally maintained by the SACC target intelligence officer. The target information is stored in a data base of the ASIS system and a navy computer operator working in the SACC operational spaces uses the QUEST data base query language to access targets and target information from the data base. Requests for a target listing and for special purpose reports must be composed in the query language each time. Response to the query is displayed on a video display unit in the SACC. The report printouts are available from a printer located in the main computer spaces.

During this period, the TIO is monitoring and duplicating the target list with the target card files. It typically is an opportunity for the TIS staff to become very familiar with the target card file procedures, although it requires almost a complete duplication of effort between the TIC and the TIS.

When the TIS goes ashore with the landing force FSCC, they obtain computer printed copies of the latest target list as a backup to their card file. Changes to the target list during the phasing of the TIS ashore are covered by a TARBUL issued by the commander amphibious task force.

Operations ashore are characterized by constant refinement of the target list, adding newly acquired targets and the employment of supporting arms on existing targets. When target information is received, the target is plotted

on the target map, a classification assigned, a target card prepared and all available information evaluated. A priority of attack is assigned and a recommendation regarding attack by supporting arms is made. The target card is then added to the target number file, the grid location file and the cross-index file if necessary.

As fire support missions are executed, the TIS attempts to expedite the surveillance reports from the available fire support sources. A damage assessment is made based on the reported surveillance. The information is added to the back of the target card and the target is updated as required. The primary sources of this information are artillery forward observers, naval gunfire spotters, forward air controllers and liaison officers.

New targets are reported to the landing force TIS from the target information sections of subordinate units who have uncovered targets of sufficient importance to be recommended for inclusion on the target list. Targets are also received from the target intelligence officer, the artillery target acquisition battery and acoustic and seismic sensors. Based on the accuracy of this information (confirmed, probable, possible or unknown), a determination is made whether to add the target to the target list, the list of targets or the inactive file.

J. OPERATIONAL CHARACTERISTICS

The operations of the TIS focus on two major functions; the maintenance of the target card file and the graphical representation of the target information on the target map. The former function appears to lend itself to an effective automated solution. The following items are the significant recurring requirements for maintenance of the target card file:

adding a target to the file

deleting a target from the file

changing information about a target in the file

changing the status of a target (active-inactive)

updating the cross-index file

The products of this maintenance are used by the TIO and the staff of the FSCC for effective fire support coordination and delivery of supporting arms. An analysis of these products indicate that the target card file provides the following specific capabilities:

provides all target information for a specific target

differentiates between active and inactive targets

sorts or catalogues targets by various parameters (which include target no., coordinates, classification, priority, target type, supporting arm assigned and target accuracy)

provides information upon which to base a TARBUL

provides information for production of the target list

Any automated solution which will be of value to the TIO must be able to perform the requirements for maintenance of the target file quickly and efficiently. It must provide the required end products (TARBUL, target lists, specific information about a particular target, etc.) as well as the capability of providing specific target information in a manner and format which can best be utilized in the FSCC.

The solution involves the manipulation and management of the information contained on each target card in such a way that the speed, efficiency and effectiveness of the TIO is enhanced. This must be done in a simple, easy and uncomplicated manner and must produce timely and accurate information.

K. SUMMARY

The organization examined in this chapter is for the landing force target information section (TIS) (typically a Marine division or a Marine amphibious brigade) which constitutes the most important and most heavily staffed section. The TIS exists at regimental and battalion level as well, but with less formality. The card file is not as extensive (due to the fewer number of targets in the zone of action of a smaller unit) and the target personnel usually perform their functions as an additional rather than a primary duty. The automated solution, however, is equally useful for subordinate units of the landing force in

assisting them in the effective and timely management of target information so that they may effectively employ their supporting arms on the most important targets.

This chapter has provided a review of the duties and functions of the target information section, the tools and doctrinal procedures of target information and the techniques of operation. Additionally, the characteristics of the target information function which can be automated have been identified and analyzed. The following chapter uses this analysis to develop a conceptual framework for the design of the target information system.

III. SYSTEM DESIGN CONSIDERATIONS

A. PRIMARY CONSIDERATIONS

1. Background

Having defined the current procedures for the target information function, the task now remains to provide a satisfactory system design for an automated solution. The design is influenced by two important considerations. The nature of the data base is both physically small in size and functionally restrictive in what information is required from it. This, combined with a requirement for a relatively lightweight, portable and versatile computer, makes the selection of a microcomputer an obvious and logical choice for hardware. This confines the solution, however, to the microcomputer environment which, while it has many desirable features, imposes a number of major restrictions on the design.

The second major influence on the design is the impact of human engineering on the user interface. The user is a Marine in the target information section of the FSCC and the functions he performs are a known entity. The system must conform both to his level of training and computer sophistication and to the functions and tasks he performs. This requires an interface which is user friendly, extremely easy to operate, sufficiently sophisticated to allow the

user to perform the required functions effectively and without error, and capable of operating in a real-time, interactive mode.

Thus, the solution is confined by two separate environments: the microcomputer environment and the one defined by the friendly, sophisticated user interface. They jointly determine the data structures, the control structures, memory allocation, interactive complexity and the system modular design. The system must be designed to operate effectively within the restrictions imposed by the microcomputer and the parameters required by the user interface. An abstraction of these environments is depicted in figure 4 below.

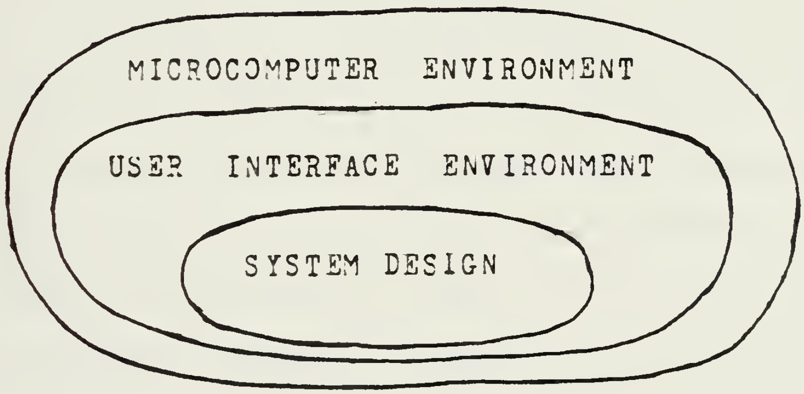


Figure 4. System Design Environment.

2. Tasks

A key task in the system design is the definition of the usage factor. This is the description of the system's processing requirement, i.e., how the data is utilized by

the system. This leads to a top-down design methodology and three important tasks which will determine the design of the data base as well as the applications program. These tasks are:

1. To identify all processing functions and subdivide these functions into modules (processes).
2. To determine all of the data that each process uses to perform its designated function.
3. To adequately describe the system retrieval requirements.

B. THE USER INTERFACE

While chapter VI will address in detail the human engineering aspects of the user interface, it is important to recognize at this point in the development of the system that the user is classified as a parametric user. Simply defined, the parametric user is one whose system input is in the form of parameters only. He is not a programmer although he may have programs available that he can use. He is transaction oriented, putting information into the system and retrieving it from the system, generally requiring a short response time. The parametric user requires current and timely data and rapid and easy recovery from errors.

In addition to designing the system to perform the tasks and functions of target information, it must be engineered for the parametric user in order for it to be used

effectively and with a degree of confidence because of its predictable behavior.

In the design of an interactive system, a very important consideration is the appearance of the system to the user. Use of the technique of interaction by anticipation, that is, anticipating the desires of the user and presenting him with a corresponding list of options, allows the user to simplify his input by selecting rather than specifying the data. The employment of menu selection techniques and computer initiated dialogue, important applications of interaction by anticipation, will be used to provide the friendly man-machine interface.

C. USER DESIGN CRITERIA

A particularly important aspect of the design is the nature of the constraints on the cognitive processes of the user. One constraint is the amount of information that a person can consider at one time and the length of time that the information can be retained in short term memory. Hence, the information available from the system should be simple enough to be quickly and easily assimilated.

The system should also be fast enough so that the user is not distracted by the loss of information in his short term memory due to a slow response time. The system should be able to reinforce user memory whenever required. This implies a user initiated request for help to which the

system must reply with the appropriate information. An important aspect in designing a help function is uniformity of the command as well as the expected reply.

A second consideration which is important in the design of interactive systems is the experience level of the user. The system should be able to cater to the novice user and effectively direct his input to perform the required tasks. It is also important for the system not to ignore the experienced user. The interface should be able to adapt to the needs and characteristics of its users based on the user's experience.

The interface should also be robust in nature. It should respond in an effective and unambiguous manner to any input and allow the user to recover from simple errors. It should discourage illegal input and guide the user to the proper inputs required. It should provide closure to the user, i.e., a logical completion to a specific action within an expected period of time. It should limit the user input to the necessary data and instructions sufficient to perform the required tasks.

This is best accomplished for the parametric user by interaction by anticipation and a restricted and unambiguous flow of man-machine communications. Thus, communications from the user to the computer is by discrete selection of semantically meaningful options, and from the computer to the user by the presentation of information contained in the

menu selection or dialogue frames. This will allow for rapid and easy operations for the user and a unity of design for the implementor.

D. THE MICROCOMPUTER ENVIRONMENT

Microcomputers impose a stringent set of restrictions on the resources available when implementing or executing a program. These restrictions include the small size of main memory, the lengthy access time and small capacity of secondary storage and the low processing rate.

Typically, microcomputers are constructed with 32 to 64K bytes of main memory. When consideration is made for the operating system, the applications program and the data base, it becomes obvious that they cannot all exist in main memory at the same time and the partitioning of memory and the arrangement of secondary storage will be a key consideration in the system design. Putting all the data into main memory is not feasible because of its size, yet putting all the data in secondary storage results in unacceptable response time.

System response time is important to the user. Thus, those operations to which he expects a quick answer must be performed quickly with minimal access time. For other operations which are logically time consuming to the user (for example, input of a new target into the target list), closure will have to be delayed (with a computer advisory

message) while the information is processed. The routines of the applications program must be designed to optimize the accesses to secondary storage, which is the bottleneck in microcomputer systems.

E. FUNCTIONS OF THE SYSTEM

From an analysis of the information provided by the target card file and the functions and duties of the target information section, a number of major functions of the system have been identified. From these functions, system output has been identified, both in the form of display on a video terminal and printed hard copy. These functions and outputs determine the design of the data base, the applications program and the overall system.

1. Primary Functions

The primary functions of the system involve the manipulation and input of target information into the proper storage formats. These functions include:

Add a target to the file
Delete a target from the file
Change information about a target
Change target status (active/inactive)
Copy data base to a backup file
Initialize the target file data base
Display certain target information
Print certain target information

These last two functions could become very extensive operations if desired. However, a carefully restrictive design of the data base model and a desire to limit the

semantic options of the parametric user to certain, systems defined operations, has reduced them to managable yet fully applicable functions.

2. Display Options

The CRT (cathode ray tube) device will be the primary user interface mechanism. Most of the information input and extracted from the system will be performed via the CRT. The interactive queries to the data base will result in the following display options:

Display a complete target card
Display a list of all the active targets
Display a list of all the inactive targets
Display the target list
Display the information for the next TARBUL
Display a list of targets by specific parameter(s)
Display parameter status for the active targets

The parameters indicated above are selected categories of target information obtained from the target card which are the typical parameters for special listings and the cross-index files. It represents a selection of those items of information which can be most effectively utilized by the FSC and the supporting arms representatives in the FSCC. These parameters include:

Target Priority
Target Classification
Target Number
Target Status
Target Type
Supporting Arm Assigned
Attacked Target
Target Information Accuracy
Grid Coordinates

3. Print Options

Hard copy of the target information is a definite requirement for operations at any level FSCC. The system will have the capability to print the target list and the list of targets. The production of a TARBUL based on the transactions with the data base since the last published TARBUL will provide a significant help to the TIO.

The target listings by specific parameter (for example, a list of all active targets, class C, priority II, of target type "SEAD" assigned to artillery) is a requirement that will be applicable to all members of the FSCC. The system will also have the capability to print a copy of the target card for dissemination to other agencies as well as to provide a manual backup in case of power or computer failure.

F. SUMMARY

This chapter along with the preceeding chapter has defined the doctrinal functions of target information, determined the environment for the automated solution of these functions and presented the system requirements for

this solution. These chapters form a necessary foundation for the subsequent chapters which address the specific details of the system and the data base design. The next chapter addresses the actual system design and includes the hardware and software selection and a top-down, modular approach. It contains important decisions concerning the data base which are developed in greater detail in chapter V.

IV. SYSTEM DESIGN

A. CONCEPTUAL SYSTEM DESIGN

1. Generality of Approach

In taking a top-down classic approach to the design of the target information system, the initial design does not consider the restrictions imposed by the operating environment. This is done for two reasons. First, the conceptual design presents a simple, traditional, straight forward solution which can, in concept, be readily implemented. Second, it provides the basis upon which modification and adjustment may be performed to fit the simple solution into the restrictive environment. The size of the system, the interface requirements, and the restrictive data base view will cause the conceptual design to be tailored and modified to operate in the selective environment.

2. Data Base Considerations

The target card data provides the entities (or records), attributes and relationships of a data base system. The controlling software, the data base management system (DBMS), would normally contain language facilities for defining the data base, for manipulating the data base information and for obtaining information from the data base. This last facility, the high level query language,

allows the user to manage the information of the data base and perform the required operational functions.

The data base concept enables the user to store the data in space saving and efficient ways. Redundancy of data can be eliminated and data items deleted which can be implicitly derived from other data items. The system allows construction of different views of the data so that different users can perform different functions on the same type of data. Applications programming is simplified since it only needs to specify parameters to the DBMS which locates and fetches the data.

Thus, the design of the data base portion of the conceptual system will require the construction of the logical and the physical view of the information, definition of the information in terms of the data base definition and manipulation languages and providing a DBMS with a facility for query language translation to operate on the data base.

3. Applications Program Considerations

The user environment remains as defined, a friendly, sophisticated interactive man-machine interface. The applications program must interact with the user and with the DBMS. The use of a query language for the parametric user would require the user to learn the data base query language. Alternatively,a collection of query language statements could be imbedded in the applications program and selected by the user utilizing the menu selection interface.

These statements would interact directly with the DBMS. Additionally, the host language could be extended to enable it to pass information to the DBMS in the form of a procedure call.

The requirement for menus, help functions and system explanations could be effectively solved by the use of user-oriented utility modules which could be accessed as needed. The basic conceptual system design derived from a top-down view of the target information system task is depicted in figure 5. This basic design will be refined to fit within the solution environment.

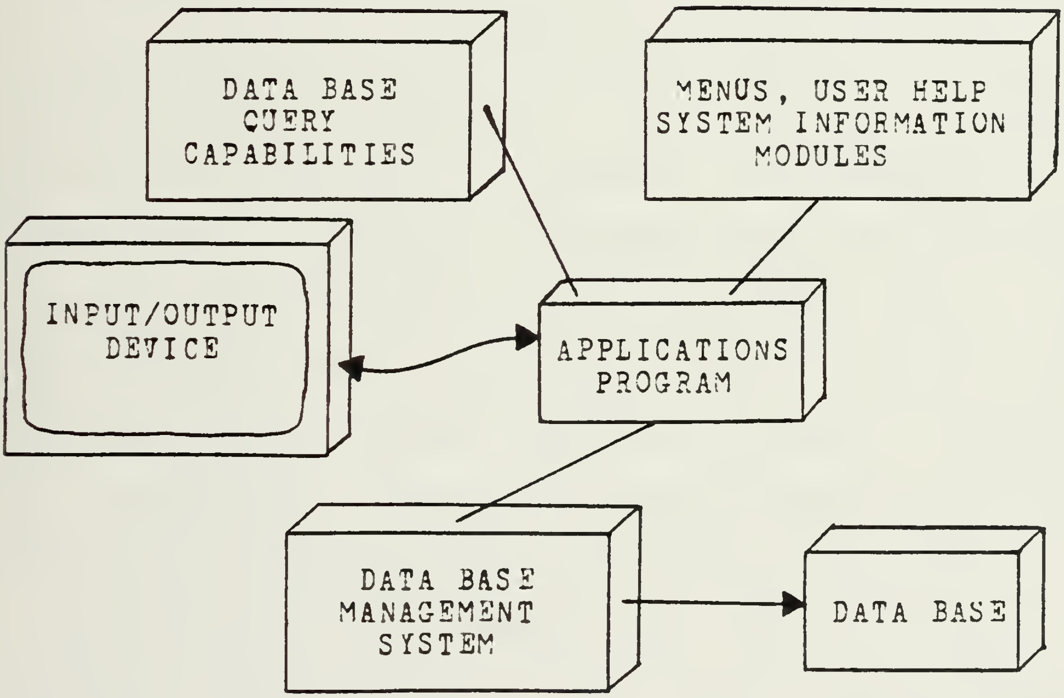


Figure 5. Design of the Conceptual Model.

B. PRELIMINARY CONSIDERATIONS FOR SYSTEM DESIGN

With the basic framework laid out by the conceptual model, the task now becomes one of attempting to insert this

"classic approach" design into the restricted environment of the microcomputer. This requires a high degree of specificity in order to identify the tools to be employed in the implementation and the methodology of employing those tools. The target machine must be identified to precisely define the microcomputer constraints. The data base model and its physical and logical organization must be defined, the applications program functions and task flow analysis must be determined and the target programming language must be identified.

C. HARDWARE SELECTION

The selection of the system hardware was driven by three considerations. First, it had to be a commercially available, typically configured microcomputer. Such generality is needed if the system was to be transportable to other microcomputers. In the search for a typical microcomputer, an effort was made to avoid the home or personal computers which, while small, easily transportable and inexpensive, possess neither the processing power nor the virtual memory capacity needed for the system.

The second consideration was for a computer that possessed acceptable size and weight characteristics for transportability, had a compact configuration, was generally rugged for a commercial product and had sufficient processing capacity.

The third consideration was availability. The ALTOS ACS-8000 is a representative of micro-systems commercially available, and was selected for use in this work. The ALTOS microcomputer conforms well to the desired computer characteristics. LCDR D. L. Smith in his thesis entitled <u>Method to Evaluate Microcomputers for Non-tactical Shipboard Use</u> cited the ALTOS as one of the top four microcomputer systems evaluated and found it suitable for use on U.S. Navy ships.

The ALTOS ACS-8000-1 is a single board Z-80A based microprocessor with 64K bytes of random access memory and two Shugart SA-800/801 eight inch, single side floppy diskette drives contained within the 16 by 7 by 17 inch compartment. It requires a CRT for input/output and supports 128 characters of upper and lower case ASCII with 80 characters per line on a 24 line video display. The computer weighs approximately 35 pounds, has a forced cooling system, utilizes standard 115 volt electric power with a battery backup and operates within a temperature range of 32-105 degrees farenheit and a humidity range of 10-90 percent.

The two floppy diskettes with the IBM 3740 single density format and the 64K of main memory gives a total memory space of 576K bytes. The high level language support for the ALTOS includes the CP/M operating system, Basic, FORTRAN, Pascal, PL/I-80, APL, LISP, COBOL and the Micro Data Base Systems DBMS for a microcomputer.

D. PROGRAMMING LANGUAGE SELECTION

The selection of a programming language was influenced by three major considerations. First, the hardware selected and the availability of assemblers, interpreters and compilers to support a programming project on this hardware narrowed the field considerably. Second, there was a desire to use a language which is relatively self-explanatory, self-documenting and transportable. And finally, the language would have to support a robust, user-oriented interactive program.

Of the available languages, Pascal was selected for a number of reasons. It has features which make it readily useable for systems and applications programming in that it is "strongly typed", requiring explicit data declaration. It forces the data base to be completely designed before the source program is written.

Pascal's structure encourages modularity as well as top-down design and implementation. It is a relatively simple language and is the basis for the proposed Department of Defense standard high order programming language, Ada. The most popular version of Pascal for microcomputer use is the University of California at San Diego (UCSD) version developed by the University's Institute for Information Systems.

The UCSD (Mini-Microcomputer) Pascal version 1.4b is a system intended to run on a stand-alone mini or

microcomputer. It is highly machine independent since it runs on a pseudo-machine interpreter. The system contains a compiler, linker, screen oriented editor and an operating system which are compatible with Z-80 microprocessors that operate under the Digital Research CP/M operating system.

Because of the microcomputer environment, there are a number of differences between the UCSD Pascal and the standard version of Pascal as defined by Jensen and Wirth. Particularly helpful are a number of string intrinsics, random access of files by a SEEK command, file handling commands and segment procedures. The segment procedure capability, for example, enables the user to segment the applications program into a main program and up to six procedure modules which are retrieved from secondary storage when called. This allows a large portion of the program object code to reside on disk when not needed, thus, increasing the size of main memory for computation and operating system functions.

E. DATA BASE CONSIDERATIONS

The design of the physical and logical data base is addressed in detail in the next chapter and, accordingly, this section will address only those items of importance to the system design. In that the user view of the schema and the conceptual view of the schema are identical, and because there is no requirement for an integrated data base, the

traditional data models (relational, hierarchial and network) will not be employed.

Consideration was given to using existing DBMS systems for the target information system but they were rejected for essentially two reasons. First, the target information system is a restricted, single application data base. It is sufficiently restricted that a general purpose, multi-faceted data base management system is not required. Second, the use of a DBMS query language was considered both time consuming and difficult to learn for the system user and unnecessarily complicated the interface. This is especially true because the system is designed to limit the type of queries allowed on the data base.

By extending the host language from the applications program to the data base, data independence is lost. However, since the system will not allow the user to access the data base in any way other than that specifically allowed by the system interface design and since there is only one view of the data, this does not present a problem.

The data base will consist of two files. The first is a flat, relational model representation with the target as a single record and the target information pertinent to that specific target as the attributes of that record. All of the active and inactive targets will be contained in this main target file. The second file will be the data base query file consisting of the primary and secondary keys for each

target. The standard system queries will return information in the target list format and will obtain all the necessary data from the data base query file.

This partition of data base files increases the data redundancy of the system since all the data in the query file is duplicated in the main file. However, this is done to improve the system response time to user queries by greatly reducing the disk accesses that would have been required to process the main file. This tradeoff is made in favor of the user interface and at the expense of additional storage space and increased program complexity.

The secondary keys used to process the target information queries (priority, classification, type, etc.) are contained in the in the data base query file. An index file containing the addresses of the target records is constructed in main memory at the beginning of the program, thus, eliminating the need for a separate record address file and reducing the number of disk seeks required to access a record from the main target file. Were this not done, the system would have to access an address index file in secondary storage to obtain the address and then access the record in the main target file in secondary storage to obtain the record. Having the index in main memory requires only one access to the disk, the access for the actual record.

The modules of the applications program directly interface to the data base files accessing, manipulating and rewriting the data as necessary. The system performs data base operations, not data base management and is an extension of the host language.

F. USER INTERFACE CONSIDERATIONS

In the preceding chapter, the user interface was defined and the general requirements determined. These requirements are now translated into specific design parameters for the target information system. Additional discussion of the user interface and details of the dialogue techniques used are presented in depth in chapter VI.

The user interface is characterized by four major attributes:

1. All communications between the user and the computer is through menus, if a simple command is adequate, or through interactive computer initiated dialogue for more extensive data entry.
2. Extensive help is available at all times. This help includes explanations of the options available, the format of the required input and examples of the correct input.
3. The display processing time is as short as possible to remain within the constraints of short term memory retention and logical closure.

4. The user is restricted to the system defined options for data input and data and information retrieval. Thus, only those procedures defined by the system through the menus and the dialogues can be used.

G. APPLICATIONS PROGRAM CONSIDERATIONS

The methodology of applications program design encompasses three separate but interrelated areas, each of which is continually influenced by the system design environment. These areas are semantic structure design, syntactic structure design and software design.

In the top-down semantic design stage, the system goals were translated into the applications program goals and the system functions and requirements were determined, categorized and prioritized. The task flow of the system was analyzed and alternatives developed and compared. The selection of the most effective solution to each of the problems posed by the system requirements was expressed as a functional module. This module was then further broken down into smaller modules which address particular parts of the functional requirement. The data structures and control structures were then determined which best enable these modules to perform the required functions.

The design of the syntactic structures paralleled that of the semantic structures and involved the determination of display formats from a comparison of different approaches to

the user interface. In addition to display formats, system response formats, error diagnostics, user aids and help facilities were also specified.

The software was designed in a top-down modular fashion and best use was made of the facilities of the UCSD Pascal system segment procedures as well as the structured approach provided by the Pascal language.

The target information system applications program consists of six major modules. The primary module is the Interface module. It acts as the executive of the program and controls the interaction of the user input/output, the data base operations and the segment procedures.

The remaining modules are segment procedures, that is, they reside in secondary storage until called into main memory by a procedure invocation. Upon invocation, they are read into main memory and computation continues. When control is returned to the calling procedure (in this case, the Interface module), the memory space is deallocated. The UCSD system allows up to six of these segment procedures and permits them to be nested in order to further reduce the amount of code necessary in memory.

The Initialize module is used when the data base system is initialized and a new target file is created. The Query module contains the menus and the semantics for the system queries to the data base query file. It is used only when target information by specific parameter is desired by the

TIO. A Utility module contains a number of housekeeping routines for constructing the TARBUL, determining target file status, copying the data base to another diskette, printing target listings and other functions.

The Target module is the major segment procedure and is used for adding, deleting and changing targets in the main target file. It also updates the data base query file and if necessary, the TARBUL file. The Inform module contains user-oriented information concerning doctrinal terminology, systems instructions, version information and tactical guidelines.

The target information system design is illustrated in figure 6. Because much of the design was influenced by the microcomputer constraint, the amount of object code resident in memory at one time has been minimized. The illustration in figure 7 shows the expected allocation of secondary and main memory for the system. (see pages 64 and 65)

H. SECURITY AND INTEGRITY

Security for the system is essentially non-discretionary. The system is secure because it is located in a secure area (the FSCC is usually a restricted, controlled access area within a secure perimeter). The targets contained in the list of targets are typically classified confidential and, therefore, the diskettes would be considered classified matter. Thus, the usual precautions

and practices for the security of classified material are sufficient for the system.

As a further safeguard and security feature, the system will have a user password which will allow only bonafide individuals to access the data base of targets. Input of an improper or erroneous password will keep the user in the outer edge of the Interface module and prevent opening the data base index file without which, the data cannot be accessed. The Utility module has a subroutine which allows the user to specify his own passwords.

The target information system will reside on an eight inch floppy diskette which will contain the Pascal operating system and the object code of the applications program. The source code, editor and compiler will be removed from the diskette to prevent any user from modifying or changing any part of the system. The user can only change the password and the target information.

Upon activation of the system, a user advisory message will be printed on the CRT screen informing the user of his responsibility to safeguard the classified information. All of the printed output of the system will contain confidential markings on each page as required by current security regulations.

Nuclear and chemical target information and analysis will be excluded from the target information system and processed in accordance with current procedures. This is

done primarily because these targets require special techniques for analysis, are typically of a higher classification than confidential and are of such a degree of sensitivity that special handling is usually required.

I. TRANSITION

A key element of the system is the physical and logical design of the data base. Since the system is functionally restrictive, the data base has been designed to provide optimal performance to the user. This has resulted in design parameters which are explained in depth in the next chapter. The chapter develops some of the important considerations of data base technology and the methodology of data base selection and file determination. These techniques were used to design the logical target information record.

System and environmental requirements impact significantly on the physical data base design and a number of alternatives are presented and evaluated. The physical record design as well as the inverted file indices are described in detail and provide a justification for the system design presented in this chapter.

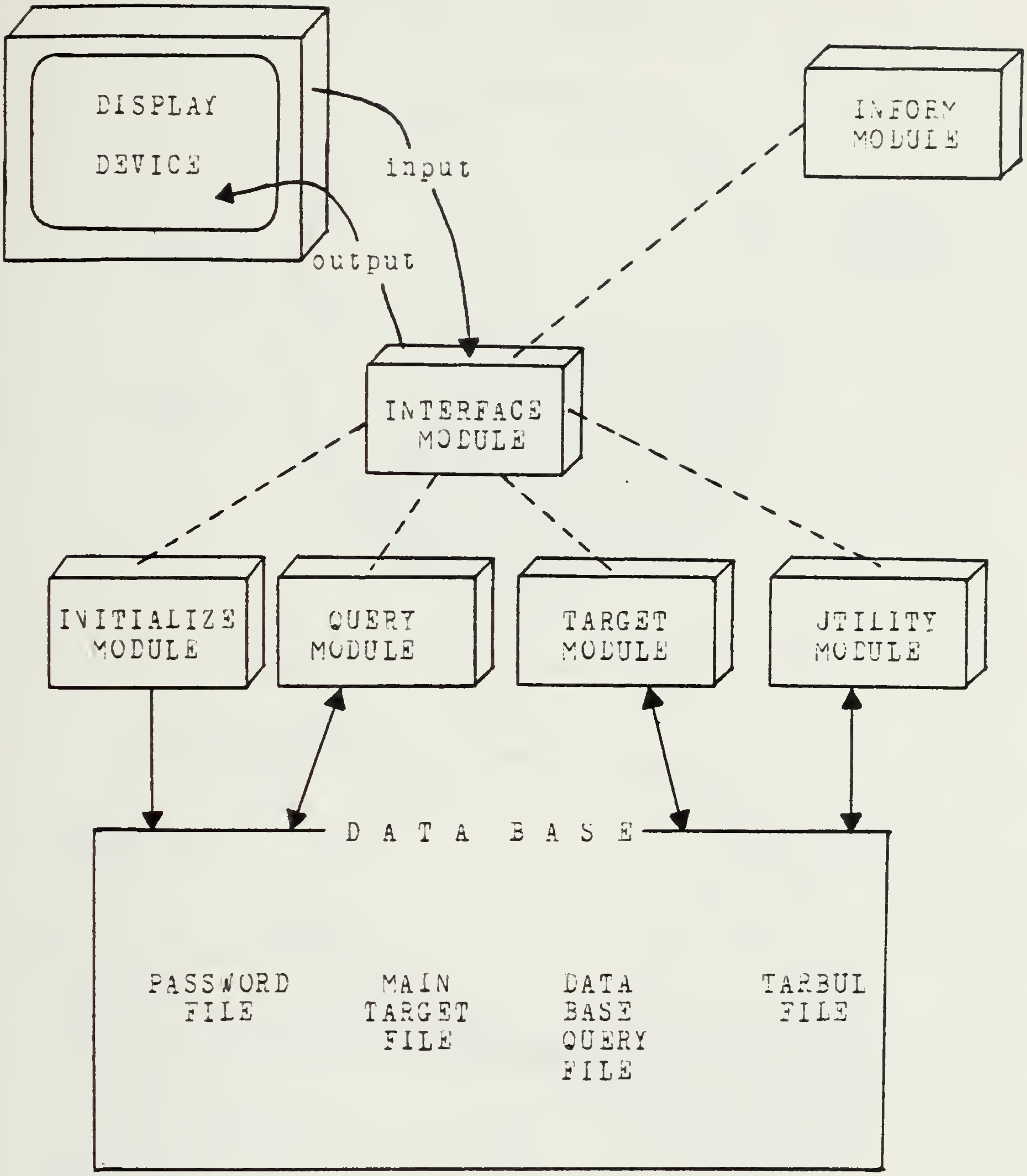


Figure 6. Target Information System Design.

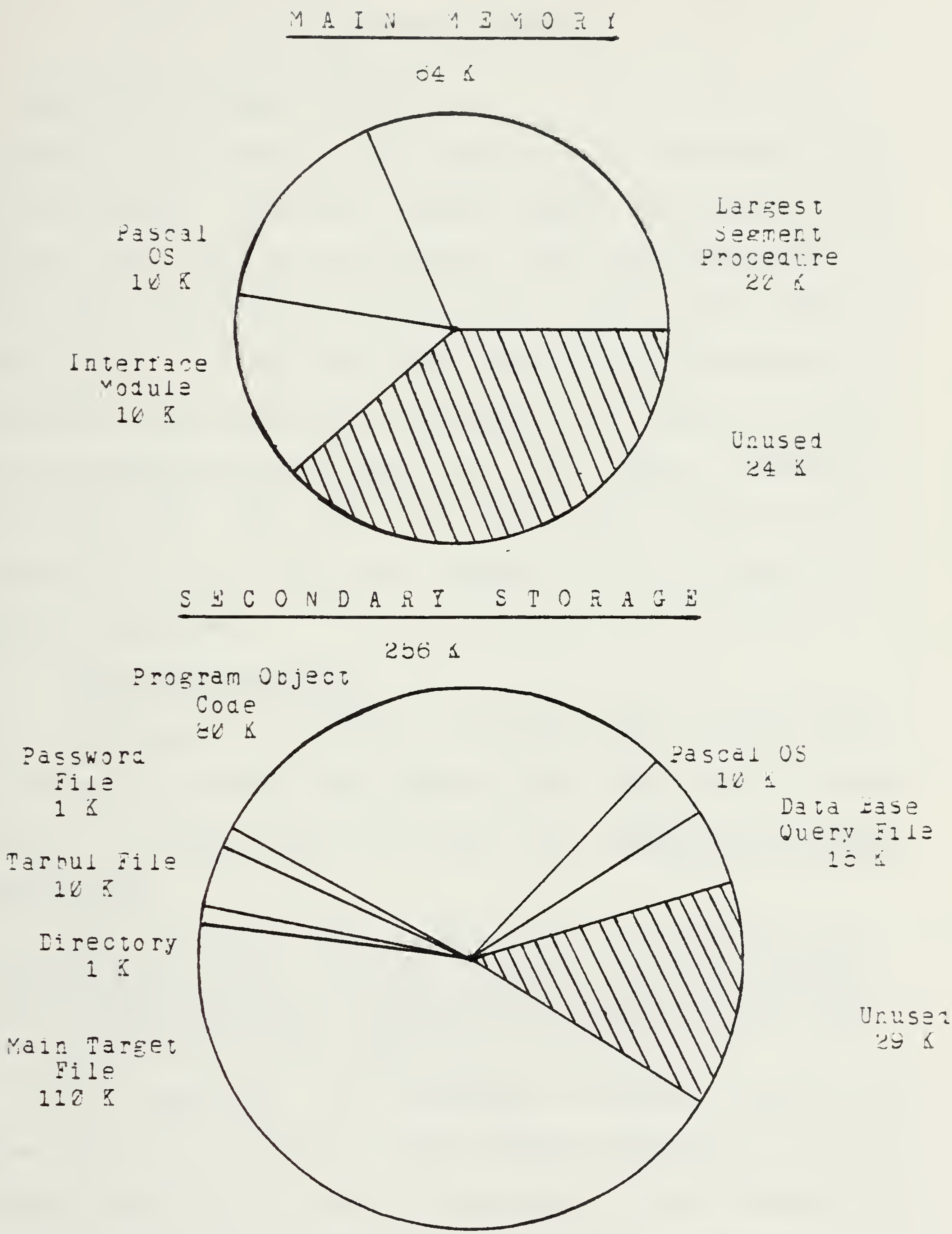


Figure 7. Memory Allocation.

V. DATA BASE DESIGN

A. PRELIMINARY DESIGN PROCESS

Before the design of the physical and the logical data base can proceed, there are certain ground rules and design criteria that must be established. Data base technology has become more formalized in the past ten years with general acceptance of three major data models, the relational, the hierarchial and the network or CODASYL. The task now becomes one of determining the content of the target information data base and which of the major data models is most appropriate for the detailed design of the logical and physical data base.

1. Data Base Concepts

Perhaps the initial starting point should be a definition of the data base. One of the most often quoted sources is James Martin's from his Computer Data-base Organization:

> "A data base may be defined as a collection of interrelated data stored together with as little redundancy as possible to serve one or more applications in an optimal fashion; the data are stored so that they are independent of programs which use the data."

It is important to distinguish between a data base system and a file system. A file system organizes the data storage capability which is provided by the hardware or operating system software. The hardware is partitioned into

files which are associated with a particular user or for a specific purpose. Operations on one file are done in isolation from the other files (or other users). Thus, to access the same information from many similar files may require as many separate operations as there are files.

A data base system, on the other hand, organizes the file storage capability which is provided by the file system. The relationship between elements or entities of the file are made accessible to the system. The user gains access to all of the data because it is now available through relationships to other data. Additionally, different users can access the same data and share it.

Access to the data base is provided by a data language, a set of operations which permit access to the data that has been organized by a data model. Data base management systems are generally classified by the way they provide access to the data. A self-contained system provides all the capabilities and required services by itself, typically, through a query language. Host-based systems carry out the retrieval and update functions only and deliver the data on request to programs written in the host system language. Occassionally, the host language is extended to operate directly with the data base, but usually with a loss of data independence.

2. Data Base Terminology

Each of the major data models refers to concepts of data in slightly different terminology. For example, the physical record type "target" consisting of target number, grid location, altitude, priority, classification and description is called an entity by one model, a segment or a logical record by another and a tuple by the third. To avoid confusion and misunderstanding, a set of terms which are partially intuitive in nature and generally from the relational model will be used. These terms, their definitions and an example from the target information system are as follows:

Record.....a group of one or more data items or attributes which corresponds to a simple record or entity [a target]

Attribute.....the smallest unit of data, a data field with a certain value [target AA0001 with the "priority" attribute has value III]

Relationship.....the connector between individual records of the same type or groups of records of different types [the list of targets]

Relation.....the set of all records of a given type [the list of targets]

Degree.....the number of attributes in a record [for a record with target number, location, priority, classification and description, the degree of the record would be 5]

Cardinality.....the number of records in a relation [the number of targets in the system]

Domain.....the set of all possible values for an attribute [priority has domain I, II, III,IV]

Primary key.....one or more attributes of a record whose value uniquely identifies the record [target number AA0046]

Secondary key.....an attribute which may or may not uniquely identify the record but which defines a set on the record [all priority I targets]

Schema.....the structure of the entire data base

Subschema.....that portion of the schema viewed by a particular user or group of users

Flat file.....a relation in normal form: a single level record array with only one record type

3. <u>File Determination</u>

The total volume of data in the target information system must be viewed with the objective of splitting it into smaller units that may be considered the basis for organizing the data base file. Having already determined the system functions from chapters III and IV, the data objects

and the relationships to perform these functions must be determined and organized. The results of this organization will become the criteria for the modular design of the applications program.

William House in Data Base Management provides an excellent methodology for file determination. Data splitting separates the system information into subsets which can be dealt with more or less independently and perhaps made an independent file of the data base. Record design determines the format of the content to appear in each record and the modes of indexing in order to establish the index data that must be present.

Volume analysis estimates the size of the individual record and the size of the record's relation (cardinality). The physical distribution of the number of records in each file must also be taken into account to determine the space management requirements. Activity analysis determines the frequency of reference and estimates the total activity for the records of each file. It is this analysis that is essential to the question of file design and one of the key considerations in the microcomputer environment, particularly the access bottleneck to secondary storage.

File design is dependent upon the record structure, physical distribution of the records in the storage device and the indexing method employed in referencing the record. The critical issue for file design is its performance.

4. File Performance

There are a number of criteria which must be considered when estimating the performance of the file design. There will inevitably be a trade off between storage space and processing speed. There are instances when the rapidity of access to information in the data base is more important than saving or optimally utilizing the secondary storage space. This may mean redundancy of data.

Gio Wiederhold in Database Design outlines seven measures of file performance. These parameters were considered when designing the physical data base. These measures of file performance include:

1. Storage required for a record
2. Time to fetch an arbitrary record from the file
3. Time to get the next record within the file
4. Time to update by inserting a record into the file
5. Time to update by changing a record in the file
6. Time for exhaustive reading of the file
7. Time for reorganization of the file

5. Architectural Perspective

The data base system architecture is also an important consideration. It depicts the natural, conceptual and physical views of the data in the data base. It is the key to data independence and gives the DBMS much of its power and flexibility.

At the most abstract level, there is the external data base. This is the way in which the user views the data base. It consists of any number of different perspectives of individual users, which are considered subschemas of the

data base. Typically, the subschemas are more important than an overall view of the external data since they define the information environment for a single user or specific application.

The external data base maps into the conceptual data base which is the logical view of the information contained in the data base. It is the schema, or combination of all the subschema expressed in the logical format of a data model. It consists of the records, relations and relationships of the data as well as the primary and secondary keys used for processing the data base.

The conceptual level maps into the internal data base, which, as the physical view of the data, is the least abstract level of the architecture. The physical data base contains the records, files, indices, inverted files and record sequences of the data base. An illustration of the different levels of the data base system architecture is shown in figure 8.

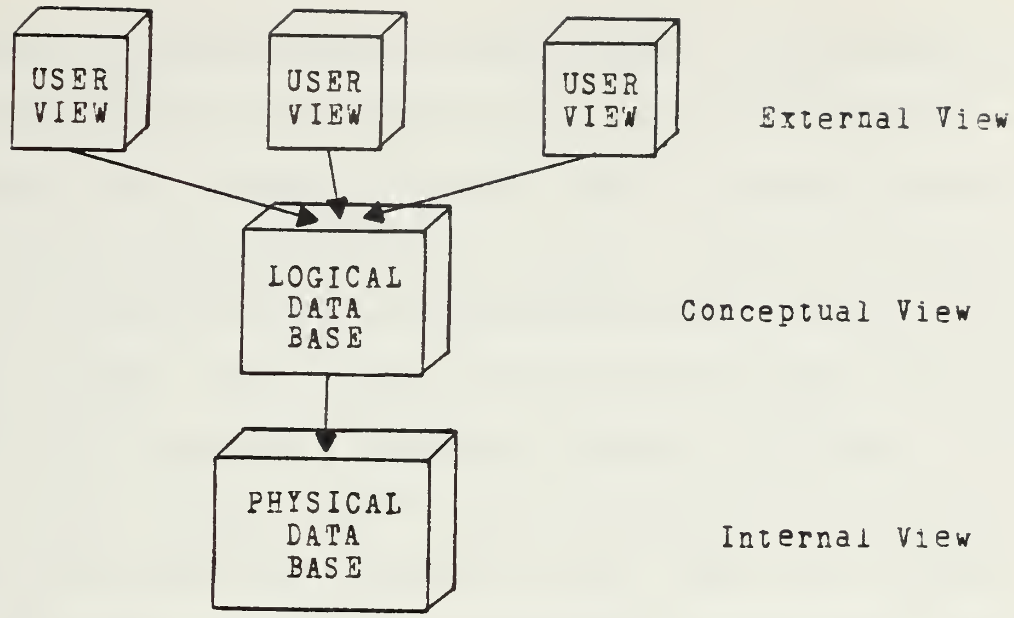


Figure 8. Data Base System Architecture.

6. Types of Systems

The discussion so far has mainly considered the data base management system concepts for the target information system. Given the nature of the target information file and the system environment, other types of systems bear consideration. An appropriate alternative to a general purpose DBMS might be a single application system.

A single application data base system establishes an operation using the available file system facilities and designs applications programs which interface to the data base. A system for the routine processing of data and the answering of a prespecified and limited class of queries is sometimes referred to as an operations system. This type of system is designated for a precisely defined and limited set of operations.

In a data base management system (DBMS), an information system, the nature of the queries will not be pre-defined by the system and lengthy searches may be necessary when a query is made. The capability to process generally stated queries is characteristic of the multi-purpose design of the DBMS and often accounts for its relatively large size and cost. In an operations system, lengthy searches can generally be avoided because the information is typically stored in the form it is needed. The two types of systems use data bases which are differently structured both logically and physically.

B. LOGICAL DATA BASE DESIGN

1. Data Splitting

Given the total information in the target information system, or more precisely, the set of data that represents this information, it is necessary to separate it into subsets which can be dealt with more or less independently.

The information for the system comes from the target card. All of the information pertinent to the data base is on that card or can be implied from it. A block record of all of this data pertinent to the system can be visualized in figure 9.

TARGET NUMBER	GRID LOCATION	ALTITUDE
TARGET TYPE	ARM ASSIGNED	ACCURACY
ATTACKED?	FIRING UNIT	PRIORITY
CLASSIFICATION	DESCRIPTION	DTG ACTIVE
PHOTO NUMBER	PHOTO COORDINATES	BDA
MAP REFERENCE	TARGET LIST?	REMARKS
DTG ATTACKED	NO.TYPE ROUNDS	STATUS
DAMAGE REPORTED	DAMAGE ASSESSED	SOURCE

Figure 9. Target Information Conceptual Record.

The data can logically be split into different segments, for example, description information, surveillance information, status information and source information, but a consideration of the user and the conceptual view of the system is necessary first.

There is only one user, the target information officer and he has only one view of the data, that of the target card. He may use that data differently depending upon the tactical situation or internal operating procedures but his logical view of it has not changed. An integrated data base will have many users and many different views of the data (one schema with many subschema). The target information data base is not an integrated data base and it has only one user and one view of the data, thus, the schema and the subschema are the same. The need for data independence

(logical data in this case) is no longer required for the system because the external view is equal to the conceptual view.

Splitting the target data does not achieve any added flexibility, simplicity, independence or efficiency. Thus, the target record can consist of the above 24 attributes, no other relationships and be organized as a flat file.

2. Record Design

For this large set of data on target information, a determination must be made of the format of the record and the modes of indexing in order to establish the index data that must be present. Two alternatives were considered, one with a flat file and a second with multiple records. The multiple record version merely added more complexity and more data to the files with little benefit to the system other than it "looked" more like a data base.

The flat file appeared to be the simplest conceptually and the easiest to implement. The data within the record was ordered in a functional manner for semantic purposes and the primary and secondary keys were determined.

There is only one way to uniquely identify a target and that is by the target number. This is primarily dictated by doctrinal procedures since the target alpha/numeric combination determines the originating unit as well as a specific target. Target grid coordinates may be considered as an additional unique key, however, a single map location

can be targeted for multiple purposes. Therefore, the target number was selected as the primary key for the record.

There are a number of secondary keys for each target but only a few of these have a real meaning to the TIO. Those keys which will be needed to access certain types of target information have been selected as secondary keys. It is for these keys that the queries to the data base will be designed. Figure 10 illustrates the primary and secondary keys of the target record.

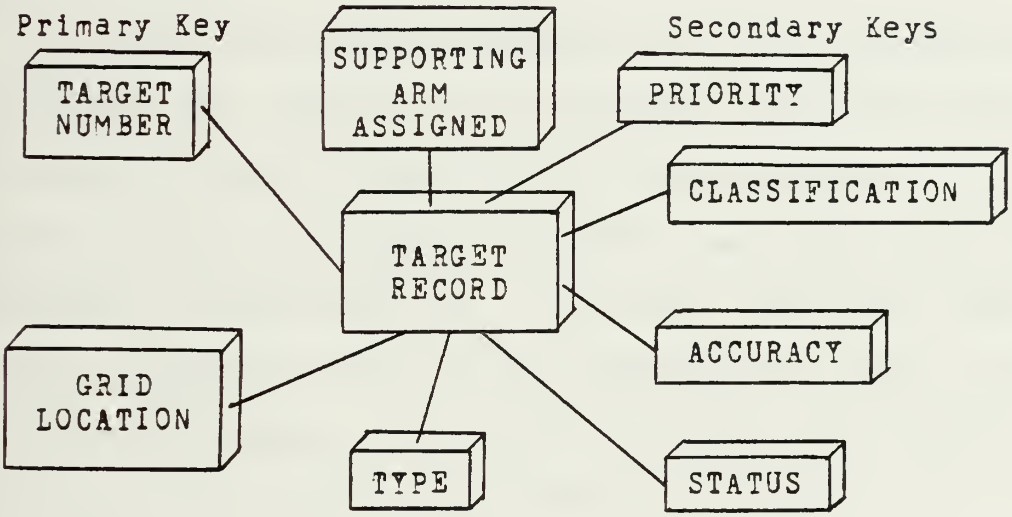


Figure 10. Primary and Secondary Keys.

3. <u>Volume and Activity Analysis</u>

Estimates must be made of individual record sizes and then of the expected file sizes as well as determining the frequency of reference of information in the data base. In determining record size, a number of considerations came into play. Data items which could be derived or implied from

other data items were deleted. For example, if a target had a BDA in the record, it had been attacked by supporting arms. If there were no BDA, the target had not been attacked.

Additional attributes were identified whose requirements were implied by other data items. For example, target type was made a record attribute (and a secondary key) since the target description, being text, would have to be semantically analyzed in order to reveal all targets of enemy "artillery".

Attempts were made to reduce the size of stored data by encoding the domains of attributes. The domain for target priority is [I, II, III, IV]. To place priority III in the target record file would take three characters; reduced to a numerical representation, it takes only one number (3). Target damage is described by a maximum of eight different words, the largest of which is 11 characters. This has been reduced to a single number from one to eight.

A data dictionary was developed which listed each attribute, its domain, data item size and data type (see appendix A). From this document, the size of the record was determined to be approximately 240 bytes. Since adequate secondary storage appeared to be available, a fixed length record was selected. In addition, since more than one BDA was expected for a given target, the record size was increased to three BDA per target bringing the record size

to approximately 370 bytes. With a maximum of 300 targets in the system, the file would occupy appproximately 110K bytes of secondary storage.

An estimate was made of the number and type of references to the data based on known and anticipated tactical requirements. Again, the design restrictions on what the user could ask of the data base played an important consideration. The majority of the information for retrieval was either an individual target card or a list of specific targets. Return of the target card to the user was a simple matter for file design, merely retrieve the record from the data base and display all of the information. The retrieval of specific information is more complicated.

The specific information about targets is best displayed as a target list since that is the most useful format for the user. Figure 11 is an example of the format required from the data base.

LIST OF TARGETS

TGT NO	CL	PRI	LOCATION	ALT	SA	DESCRIPTION
AA0021*	C	II	34566543	90	AIR	2 T-62 TANKS
AA0020	C	IV	23455565	40	NGF	5 INCH COASTAL GUN
AA0056*	E	IV	67665466	50	NONE	4 SCHOOL BUILDINGS
AA0013*	A	III	65677412	15	NGF	BUNKERED TRENCHLINE
AZ1022	D	II	76885454	110	ARTY	BN ASSEMBLY AREA
AZ1005*	C	I	34345656	20	ARTY	PLT ZSU 23-4
AA0012	B	III	56445456	10	NGF	CONCRETE BUNKER

NOTE: * indicates target list

Figure 11. Example of a Target List.

4. Design Conclusions

Because the data base is a single application data base and is an operations system rather than an information system, the use of a specific data model was rejected. The flat file format lends itself to the relational model and the resultant system approximates the relational methodology. The system does qualify as a data base. However, it is a very restricted one due to its specific purpose.

Access to the data base, given the limited choice imposed upon the user by the system design, was by extending the host language rather than the use of a query language or imbedded data language. The record design incorporates the target information as one record with one primary and seven secondary keys for a total of 22 attributes. This monolithic record is depicted in figure 12.

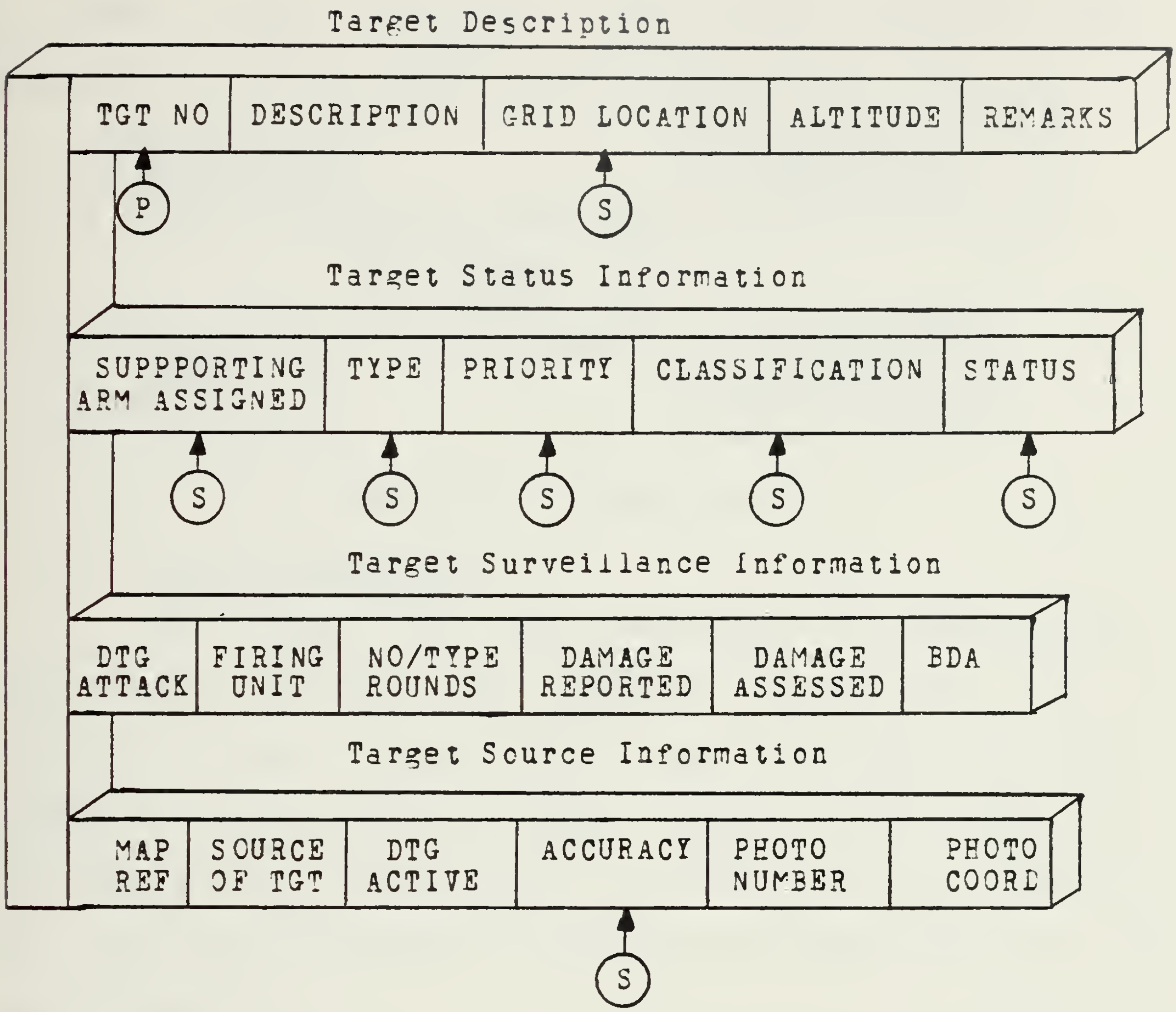


Figure 12. Logical Record Design.

C. PHYSICAL DATA BASE DESIGN

1. System Output

The system output must be considered before discussing the physical data base design. These end-products require a certain content, format and response time. Retrieval of information for the TIO must be rapid and the physical design of the data must facilitate speed, even at the cost of storage efficiency.

These items of system output have previously been identified and are depicted schematically in figure 13:

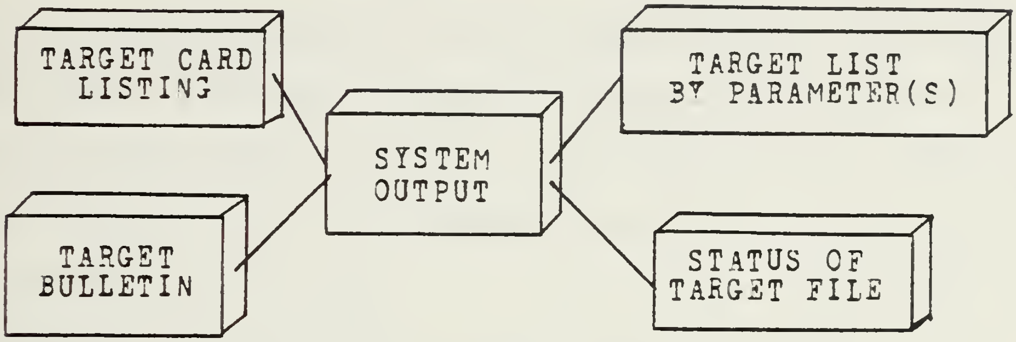


Figure 13. System Output.

2. Index File Design

The choice of file organization is dependent upon the record structure, the physical distribution of the records in the storage device and the indexing method employed to reference the record. To some degree, the amount of storage space available will influence the file design as well. The critical issue is, however, the efficiency of its performance.

The record is accessed by its primary key. The target number, unfortunately, is not always assigned in sequence. Thus, there is no logical order inherent in the target number although they could be ordered in numerical sequence for the sake of order. But there is no consistent order to warrant the use of sequential, indexed sequential, hashed or binary tree storage schemes. The use of the dense index allows us to access the required target efficiently (with only 300 targets) as well as insert new targets easily at

the end of the file. While deletion of targets would leave holes in the storage file, an unused target space record could be maintained which would keep track of the holes and assign newly inserted records in the available space.

The dense index would have to be made on both the target number and the grid coordinates, since it is a doctrinal requirement to be able to sort on both. The UCSD Pascal implementation makes this a much easier operation with its string intrinsics and random access capability of relative records. This also allows the index to be stored as a subscripted array and enables the system software features to do most of the manipulation. The index design is illustrated in figure 14. It is essentially two arrays each consisting of the target number or the grid location for each of the allowable 300 targets.

Tgt no	Tgt no	....	Tgt no	Tgt no
Rec no 0	Rec no 1	....	Rec no N	Rec no 300

Grid	Grid	....	Grid	Grid
Rec no 0	Rec no 1	....	Rec no N	Rec no 300

Figure 14. Target File Index Design.

The ease of the UCSD Pascal random access capability is illustrated in figure 16. The SEEK command on the file name and the record number provide for a base address and an offset to the required target number. This allows for quick and easy access to any target requested by the user.

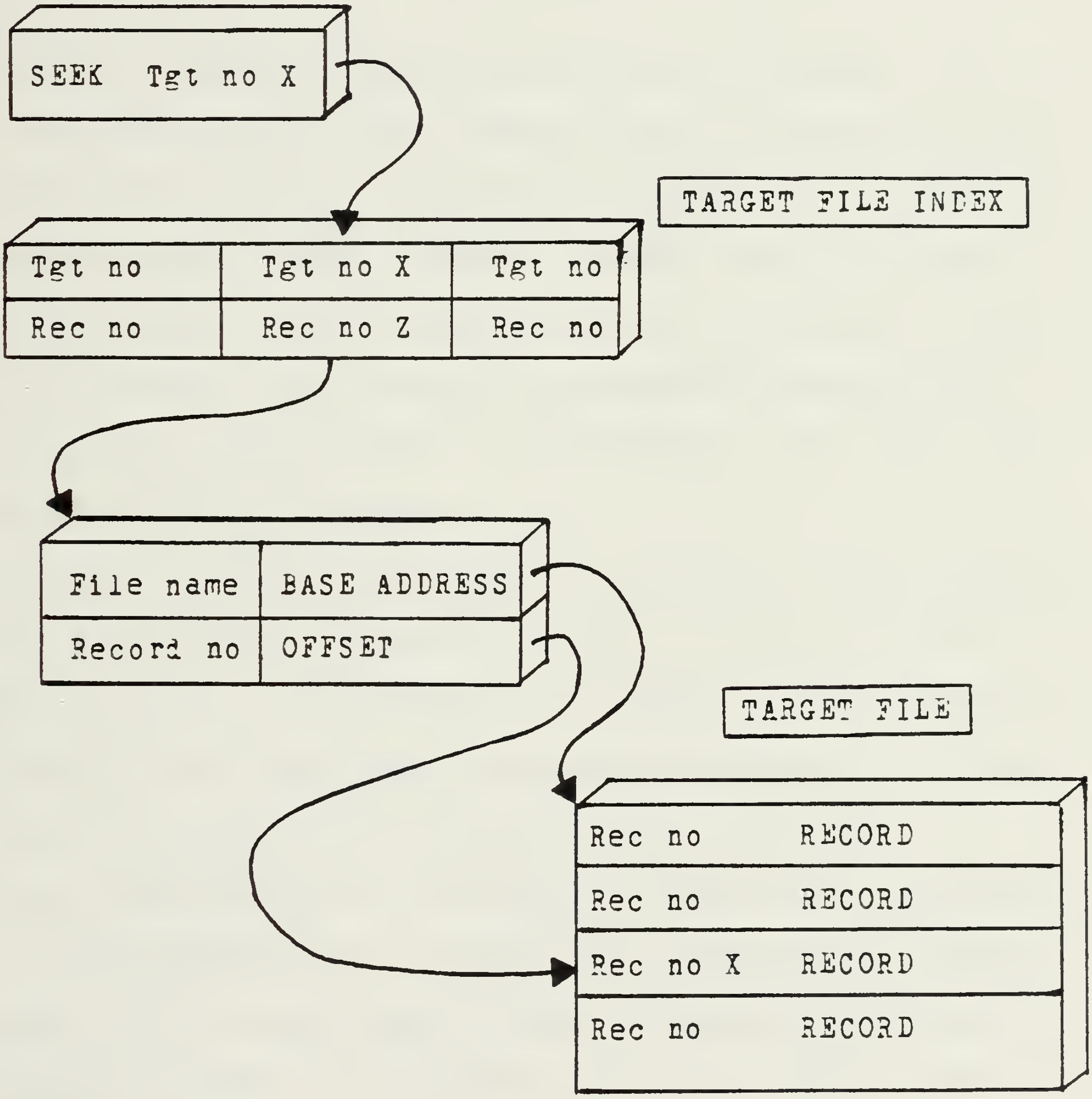


Figure 15. UCSD Pascal Random Access Capability.

3. Physical Design Alternatives

The file design will permit easy and efficient access to the target record. The display of the target card is essentially solved by this design. What remains is the accessing of the important attributes for the target listings by specific parameter, i.e., the queries from the TIO.

The most straight-forward approach is to access all of the required data from the target file. An efficient method of doing this would be to use multi-linked lists through the appropriate data items. Header records would provide a pointer into the file and links would provide access to each item of specific data. There are overhead considerations in this approach, particularly in rearranging the links when adding or deleting a target.

A major disadvantage to this approach is the estimated time it would take to process a query. If all priority I targets are to be retrieved, the program module must find the header index and follow the pointer through the target file until it found each of the priority I targets.

The disk accessing to secondary storage is a bottleneck in a microcomputer and should be minimized whenever possible. The access time to find all priority I, class C, previously attacked, tank targets could be quite lengthy. Arranging the file into blocks of five to ten target records per block would decrease the amount of disk accesses.

A second approach would be to use an inverted file structure (often used in data base systems to improve direct access to certain data) for each of the secondary keys with a pointer (actual or symbolic) to each of the specific records. An access to the entire inverted file index would identify by name or by location, each of the applicable records. Once determined, the records could be retrieved.

To process a target list with multiple parameters, only the target numbers from the indices need to be read into memory and the appropriate intersection made of the common record attributes. This would entail one seek per index and then one seek for each appropriate record. Additional efficiency is obtained if all the index files are read into main memory when the user is going to make accesses to the data base. This will reduce the number of disk seeks required to access records and is particularly effective for multi-parameter queries.

There is a bit more efficiency in the second method, particularly in accessing data with multiple parameters but the cost is in use of more secondary storage for the inverted files. With the amount of secondary storage available (see the estimate in figue 7), the trade-off between storage space and processing speed is considered acceptable.

A third alternative is even more expensive in terms of secondary storage since it calls for redundancy of target

data. A subset of the main target file can be formed to provide the data in a more accessible form. The file would be a separate record extracted from the main target file and consist of only those attributes or items which will be needed for the target query and the resulting output listing. This would eliminate the need to access the main target file for queries since all the system queries would be confined to the data base query file. Once the target number was identified by the query mechanism, the appropriate target listing information would be obtained from the file and displayed on the CRT screen.

This data base query file would have records of 45 bytes in length with a maximum file size (for 300 targets) of about 14K bytes. The logical record is illustrated in figure 16. To access the data in this file, the inverted index files could be used.

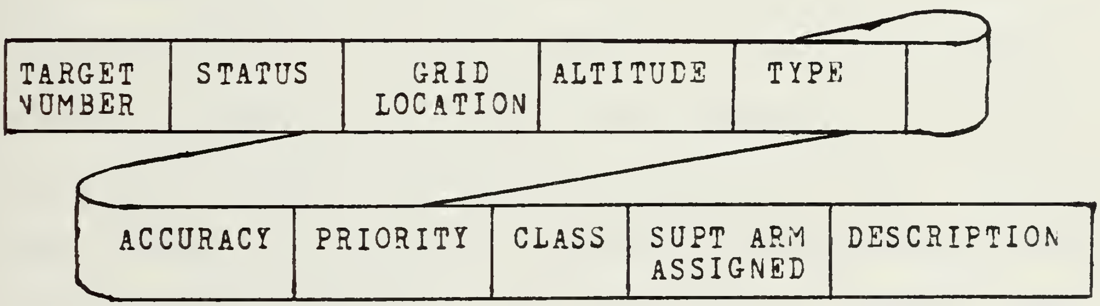


Figure 16. Data Base Query File Logical Design.

The data base query file would be loaded into main memory each time the Query module is activated. There is no requirement to write the file to secondary storage when the Query module is deactivated since there will be no changes

made to the file. In that the Query module and the data base query file would be simultaneously located in main memory, queries could be quickly and efficiently processed.

This eliminates the need to access the disk to perform queries, therefore greatly reducing the processing time. The cost, however, is in an additional file in secondary storage, an array to hold the data base query file in main memory, redundancy of data and tailoring of the data base query file and the Query module to fit into main memory simultaneously. There is also the added complexity to the program when additions, deletions and changes are made to the main file in that these changes must also be reflected in the data base query file.

Consideration was given to doing all updates to the data base query file while it was located in main memory since the array which holds the records is a static data structure. While it would improve system efficiency and preclude loading the data base query file each time the module was called, the chances of loss of data through a power surge or a system failure are sufficiently great to eliminate this approach. The differences between the two data files after a long period of uninterrupted operation would render the query capabilities invalid because of inconsistent data. The expected configuration of the allocation of main memory during data base query operations is shown in figure 17.

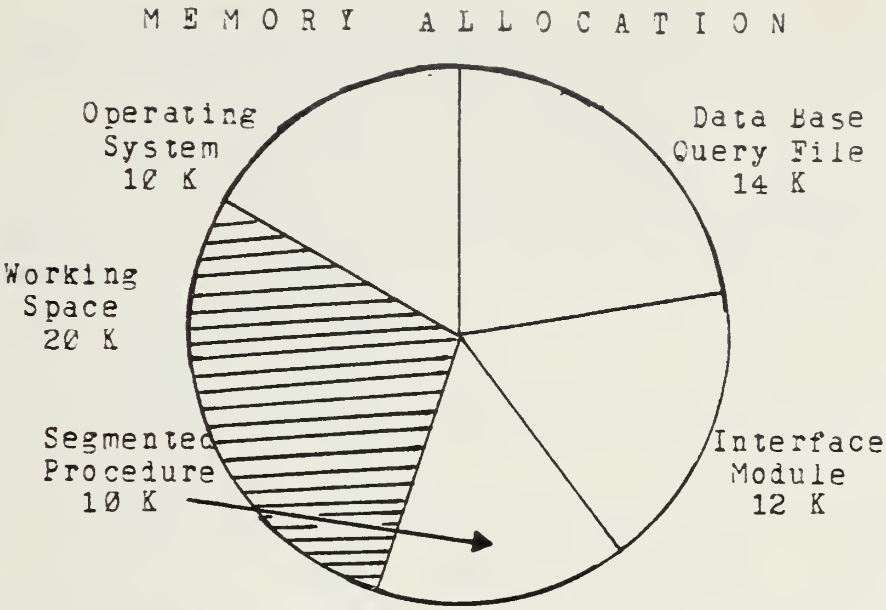


Figure 17. Main Memory Map For Data Base Queries.

While each of the above alternatives supports the logical design of the data base, the third alternative is the fastest and was selected because of the user's need for timely access to the data base information. The expensive trade-off in added complexity and redundancy is made in favor of the user.

4. Inverted File Design Considerations

An inversion on the secondary keys would allow the system to conduct multiple-key processing of queries. Each of the secondary keys would have a separate inverted file for each of the values of their domain. The index would contain the actual pointer to the appropriate record in the data base query file. An example of an inverted file for the target priority attribute is shown in figure 18.

Target Priority Index

I	1	3	7	14				
II	2	5	9	11	15			
III	4	16	10	8	14	13	6	17
IV	12	18						

Figure 18. Example of an Inverted File Logical Structure.

The implementation of the inverted files could employ a linked list of the target location pointers by specific domain of the attribute. However, this file must be updated each time the user adds or deletes a target from the system as well as makes a specific change which will affect the index. For example, a priority I target could be changed to a priority IV target after successful attack by supporting arms. This would necessitate a change in the main target file, the data base query file and the inverted index for target priority (both for priority I and IV) as well as an input for the transaction log of the TARBUL.

This disadvantage combined with the complexity of implementing linked lists, maintaining multiple inverted index files and the overhead involved detracts significantly from the elegance of using inverted files. A simple, practical and straight-forward solution is required.

6. Flat File Array Processing

The data base query file is implemented as a single dimension array of records so that it can be loaded into main memory. The UCSD Pascal system performs effective array processing and this feature can be used to perform the data base query functions. The method selected for the target information system is array processing of the flat file.

The Query module prompts the user to select the special criteria for the target listing. It does this by presenting the user a series of menus from which the secondary keys and their respective domain attributes can be selected. Once the attribute is selected, the module processes the array to determine if the targets in the array possess this attribute. Each target with the attribute is flagged and the system returns to the menus for further key selection. A domain can be selected only once per list. Thus, the system permits only the logical "ANDING" of one attribute of the domains of the secondary keys.

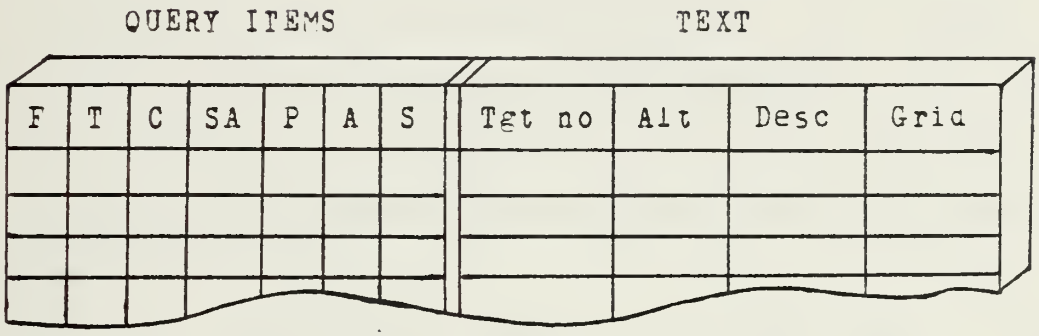
Upon selection of another attribute, the system will process only the targets which were flagged by the previous array processing. This will significantly reduce the search time and result in increasingly greater refinement of the list for each subsequent part of the query. Since there are only six secondary keys, the maximum query size is six items although the user could stop short of that number at any time in the processing. When the user stops the query and

requests the listing, those targets which are flagged are accessed and written to the console screen. When the next query is initiated, all the flags are reset.

Two additional design features have been incorporated to speed the array processing. The first feature is the storage characteristics of the secondary keys in the record. Each is stored as a single character requiring no type conversion thus, enabling quick and easy comparisons. The second feature is in the design of the query menus.

The menus have been arranged so that the most discriminating indices are presented to the user first. Target type has a domain of nine values and is presented first to the user. Thus, the first pass at the target list will probably result in the smallest list of flagged targets. This reduces the amount of array processing for the remaining portions of the query. For example, if the system has 100 targets and 82 of these are "active" status and 14 are of type "tank", then the first pass in either case will be for all 100 targets. If status was the first part of the query then the next search would be through 82 "active" targets until the "tank" targets were found. However, if type was the first part of the query, only 14 records of type "tank" would have to be searched to find the "active" targets. The first search processed 182 targets, the second, using the least-list principle, processed only 114 targets.

The physical design of the data base query file is shown in figure 19.



F....Flag
T....Type
C....Class
P....Priority
A....Accuracy
S....Status
SA...Supporting Arm

Figure 19. Data Base Query File Physical Design.

5. Data Base Partitioning

In addition to the main target file and the data base query file addressed above, the functions of the system require additional file considerations. In particular, there is the requirement to produce a TARBUL when requested by the TIO. The system must retain in a separate file, all the information that is appropriate for the TARBUL. Typically, this information consists of targets added to or deleted from the target list, changes to targets on the target list and significant BDA on attacked targets. The Target module of the applications program extracts and formats the appropriate information for the TARBUL file in conjunction with normal processing.

A security feature of the system is a user defined password which allows entry into the program only when the proper password has been received. In order to provide for the retention of passwords between uses of the system, a small file was constructed which contained the user password. A procedure of the system allows this password to be written to the diskette and retrieved when the system is activated.

There is a requirement to provide the user with a file status report on a periodic basis. It is essentially a statistical breakdown giving the number of active targets, inactive targets and targets on the target list as well as a count of the targets in each attribute by domain. A separate file could be kept for this information, but to decrease complexity and storage requirements for the system, a routine from the Utility module is used to accumulate statistics from the data base query file and display the information to the CRT screen when requested by the user. Once again, having the data base query file in main memory will reduce the computation time needed for this process.

The data base is, thus, partitioned into two data base files (one a subset of the other) and two utility files, the TARBUL file and the password file. They must share secondary storage space with routines of the operating system and the applications program object code. The partitioning of the data base is depicted in figure 20.

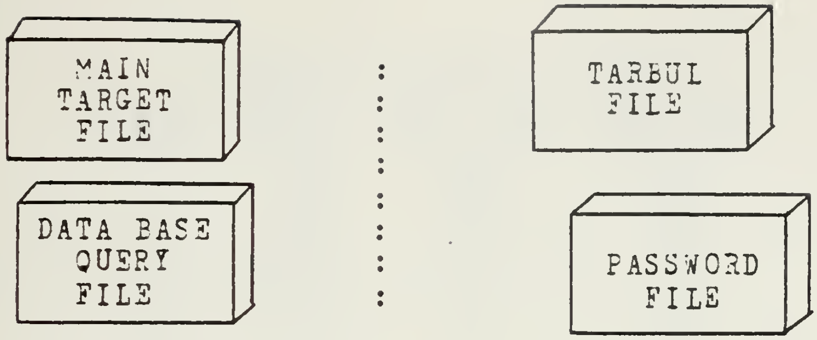


Figure 20. Data Base Partitioning.

D. OTHER CONSIDERATIONS

In considering inverted files, it was determined that some record attributes did not yield a sufficiently discriminating index. For example, the data item "status" yielded a poor index since only two values are possible, active and inactive. In an effort to establish more discriminatory indices and at the same time reduce the physical size of the data items in a record, index items were combined and compressed in a coded form. Target status was enlarged to encompass the target list index and the target attacked index. The combining of these three indices, each with domains of value two, yields one index with a valid domain of six values. The newly formed index is as follows:

CODE	ACTIVE TARGET	TARGET ON TGT LIST?	TARGET ATTACKED?
1	yes	yes	yes
2	yes	yes	no
3	yes	no	yes
4	yes	no	no
5	no	--	yes
6	no	--	no

The data dictionary, which was developed in response to the volume analysis of the data base record, addresses each attribute separately by name, data type, physical size, logical size and lists the domain where appropriate. Specifics about the target record and the query record are listed as well as a determination of the physical record size. All the data in the record is in ASCII character format; even items such as the grid location and altitude, which are actually integer values, are stored as characters. Conversion, when necessary, is performed by the applications program.

E. SUMMARY

The primary consideration in the design of the data base was ease of use and speed for the user in the microcomputer environment. This consideration overrides the inefficiencies of a dual data base record. Moreover, the complexity of the processing requirements is invisible to the user. He is concerned only with fast retrieval of certain types of information. As the casual user, he is not concerned with high level query languages and their use. Rather, he

requires a machine that will serve his needs and not the other way around.

E. F. Codd, in his 1974 article Seven Steps to Rendezvous with the Casual User stated that "It is projected that by the turn of the century, the majority of data base management systems will be oriented toward the casual user". A system which will be able to perform its function quickly, easily and accurately during the intensity of combat operations will be the one that is specifically designed to conform to the user's environment. This system was designed with this principle in mind.

The following chapter describes the important considerations of the user interface and the methodology employed in making this interface an effective one. It considers the psychological issues which affect the man-machine interface as well as the modes of user input and computer initiated dialogue. These are the techniques which enable the target information system to conform to the user's environment.

VI. INTERACTIVE INTERFACE DESIGN

A. GENERAL

One of the major design features of the target information system is that it provide a friendly yet sophisticated user interface. It should be sufficiently sophisticated to perform all of the required functions simply and efficiently with only a minimum of interaction from the user. The environment must be a friendly one, allowing the user to recover gracefully and with minimum effort from error, guiding the correct input and providing the user with assistance or information when needed.

The user is a Marine, trained in the conduct of supporting arms operations in a combat environment. He is a parametric user, a casual operator of a computing machine, with no computer training and a limited capability of operating a computer. The system must be sufficiently simple for this user to learn to operate it effectively in a minimum of time and with a minimum of effort. It must inspire his confidence, simplify his task, increase his effectiveness, reduce or eliminate any computer "anxiety" and, most importantly, enable him to accurately and quickly carry out his mission.

This chapter outlines the design criteria and techniques used in determining the quality of the man-machine

interface. These criteria were employed in the design of the applications program and constitute its basic framework. The system effectiveness can only be measured by how well the interface between the man and the machine has succeeded. James Martin, in his <u>Design of Man-Computer Dialogue</u> described the psychological impact of the interactive interface on the user as follows:

> "It has become increasingly realized that many information processing operations are best carried out not by machine alone, nor by man alone, but by a judicious combination of man and machine... A key to success in many real time operations lies in the recognition of machine limitation and the building into the system of appropriate human capabilities."

B. DESIGN PRINCIPLES

There are four basic principles used in the design of the user interface. First, it must be self-explanatory. The user must be able to use the system without reference to an external source. This implies that the system guide and direct the user in the execution of his tasks regardless of his level of expertise. This requires simplicity, ease of use, and elimination of system failure. Second, the system must be self-helping. Whenever the user wants or requires help or assistance, the system must respond. It should identify the improper input, guide the user to the proper input required and provide an example of the correct input when appropriate. Accordingly, error messages must be explanatory and the system must respond to every input.

The third principle is a requirement for a simple interface with the user. The system must respond in a timely fashion to input which is short, simple and obvious to the user. The processing complexity must be invisible to the user for every procedure of the system. They should all appear to be a straight-forward and simple task.

The fourth principle, previously mentioned in Chapter III, is interaction by anticipation, that is, anticipating the desires of the user and presenting him with a corresponding list of options. Thus, the system can avoid the problems of employing error diagnostic and advisory messages. Only those actions that are legitimate are presented for user selection. Input of any of the displayed actions will result in a syntactically correct command and allows further processing. Input of an action other than the legitimate one results in a simple user advisory message and obviates the need for elaborate diagnostics. The most common type of dialogue that uses interaction by anticipation is menu selection and to a lesser degree, form filling. Menu selection allows the user to select the desired option rather than requiring him to specify that option.

C. PSYCHOLOGICAL ISSUES

1. Short-Term Memory Considerations

Our short-term memory holds interpreted units of information for up to 30 seconds before it fades away. With

continued exposure to the same type of information, short-term memory retention can be improved but essentially, we are able to retain only a limited amount of information at one time. George Miller's classic paper in 1956, _The Magical Number Seven--Plus or Minus Two_, described experiments which suggested that the short-term memory was limited to a perception of about seven units. For terminal interaction, this implies that the processing capacity of an individual is limited to only a few items and that it should be taken into consideration when designing menu formats. They should be simple, semantically meaningful, arranged in a logical progression (to the user...not the programmer) and brief.

2. _Closure_

There is great psychological relief to short-term memory when information no longer needs to be retained. This produces a powerful desire to complete a task within the short-term memory span, reduce the memory load and gain the psychological relief. Closure is the completion of a task leading to this relief. The user expects to experience closure after completing an activity. Any delay in achieving closure or the interruption of this process is frustrating.

The pressure for closure implies that the user (particularly, the novice or parametric user) will prefer multiple small operations rather than one large, complex one. In system design, this suggests that interactions be

defined in sections or logical segments so that completion can be obtained and information released. All actions of the user should be responded to in a positive manner by the system.

3. User Anxiety

The user attitude toward the computer can impact upon his learning and performance with the system. Computer "anxiety", generated by fear of failure, may reduce the user's short-trem memory capacity and inhibit his performance. The system should put the user at ease but without being patronizing, obvious or cute. The user will respond better if the instructions are clear, unambiguous, expressed in familiar terms and easy to follow. Constructive advisory messages and positive reinforcement are prefered to threatening, condemning or meaningless error messages. "Please reenter your choice" is more user friendly, less intimidating and more effective than "Bad entry-error 21". The target information system has been designed to provide a comfortable, helpful and friendly environment.

4. Control

A driving force in human nature is the desire to control. In using computers, the novice is perfectly willing to follow the computer's instructions and accept the computer as the controlling agent in the interaction. As his level of experience increases, the user may resent the computer's dominance and may look at the computer only as a

tool. Thus, the system should be designed to enhance the user control or at least the appearance of user control. Properly formatted menus, advisory messages and error diagnostics can give the user the impression that he is in complete control of the situation. The menu allows him to make decisions on input parameters as well as selecting different functions.

D. RESPONSE TIME

A simple limit on response time, the time it takes for the system to respond to a command, is desirable for effective man-machine interface. An acceptable response time is a function of the type of command and the user's expectation of what a reasonable response time is. For some operations, he is content to let the machine crunch away but for others, he expects an immediate response. The timeliness of response to target information queries was the primary factor for the design of the dual data base.

In normal conversation, the user's expectancy of a response is within about two seconds. A lack of response within four seconds would be an unnatural break in the conversation. In studies of man-machine interface, a response within two seconds has been shown to constitute an important and reasonable boundary in the effectiveness of feedback. Errors must be responded to within two to four seconds so that the closure period is forced at the

appropriate time. While system initalization may be acceptable to the user within 30 seconds, he expects to access the next menu or to receive input help almost instantly. When the delay is expected to exceed the two second parameter, the system should acknowledge the command and indicate that processing is underway (and periodically reinforce this until the process is complete). This ensures that the user knows his command has been accepted and the machine is processing the request rather than observing a blank screen and wondering what to do next.

E. INPUT MODES

1. Mode Selection

Among the different types of interactive dialogues considered, computer initiated, form filling and menu selection were determined to be the most appropriate for the parametric user and the specific application of target information. A combination of these three methods provides flexibility, ease of use, simplicity and covers a complete range of the system functions.

The computer initiated dialogue, where the user responds to the computer, has the advantage of requiring very little training for the user to operate the system. However, the dialogue can be rather lengthy, the system can be rather slow to respond and there is a loss of flexibility in the sequence of the dialogue. The form filling technique, where

the user fills out form on a visual display device, is straight forward for the operator in that all he needs to do is provide the appropriate information. Error diagnosis must be immediate to be effective and cursor manipulation must be considered.

Menu selection, where the user selects an appropriate response to a number of choices, requires little or no user training and has the advantage that the user may be informed about the full range of the system features. A simple exit from the menu sequence and the opportunity to return to previous menus enables the user to achieve flexibility to navigate through the system. The limited number of choices on any particular frame and the information about the sequence of frames which leads to the current one provide a narrow context within which it is easy to design effective user aids and error messages.

Menu selection is a form of computer initiated dialogue since it is asking the user a question and providing a limited set of valid answers. The user determines the appropriate input and the system responds with the answer or another menu. This contributes to the ease of use of the system and the untrained user can become proficient in a very short time. The technique does run the risk of being too slow and tedious but it can be speeded up by a high speed terminal and fast access to menus. Figure 21 is an example of a menu from the target information system.

Appendix B contains a more detailed example of the type of interactive interface that the user of the system would use. It shows the menus and the advisory messages that the user would encounter while using the Query module to perform target queries on the data base query file.

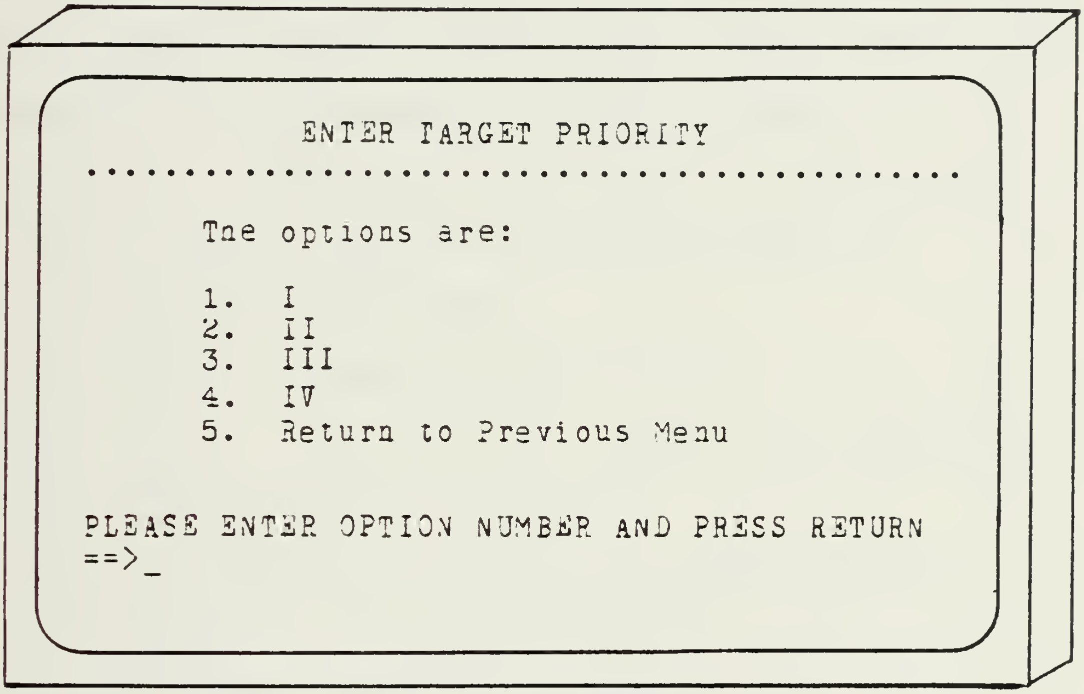


Figure 21. Example of a Menu.

2. <u>Menu/Form Filling Formatting</u>

The computer initiated dialogue will always require an action from the user. The options presented to the user generally include continuing processing, exiting, obtaining help and achieving closure. Even improper input will be responded to in an effective, immediate manner. Design of the formats for the dialogue will ensure that the range of options provided to the user will meet these prerequisites.

The manner in which the data is formatted can affect the efficiency of the operator by influencing both his speed and his error rate. A poorly formatted dialogue can cause bewilderment, anxiety and improper input. James Martin in his book, <u>Design of Man-Computer Dialogue</u>, lists twelve criteria for the design of menu and form filling screen formats. These criteria were used in the design and implementation of the target information system:

1. Display a small amount of information at one time
2. Do not include unwanted/unneeded information
3. Have one idea per display
4. The operator response should be short
5. The computer should always respond to the operator
6. Use formats designed for clarity
7. Strive for similarity (position, format, terms)
8. Avoid difficult words or characters
9. Provide an easy means for correction
10. Make instructions to the operator stand out
11. Clean up the screen when possible
12. Make it easy for the operator to ask for help

S. E. Engle and R. E. Granda in their work, <u>Guidelines for Man/Display Interface</u>, mention many different and useful techniques for improving the interactive dialogue and the user input. Some of the more important points used in the design of the target information system are paraphased in

the following list:

- Avoid abbreviations and contractions
- Be consistent in use and meaning of technical words
- Use examples to supplement instructions
- Be consistent in presenting identical/similar data
- Use numbers when listing selectable items
- Place most probable items at the top of the menu
- Standardize screen organization and format
- Give user directions before the list of choices
- User input should be kept to a minimum
- Present data in a recognizable order
- Avoid verbosity and wordiness

F. ERROR HANDLING

Well designed diagnostics and error messages can guide the user to enter the correct commands. When the system prompts the user that an error has occured, it should allow for error correction immediately. In the menu selection dialogues, the range of options is predetermined by the system and only a valid input will result in the appropriate closure. Invalid inputs can be easily determined and appropriate guidance provided to the user in the error message to obtain the proper input.

Form filling dialogue can check the field length and field type (for example, a grid location would consist of eight numbers and any other input would be invalid). Error

messages must indicate the nature of the error and how to recover from the error. When appropriate, the system should respond with an example of the proper input. Figure 22 below is an example of an error message from the target information system:

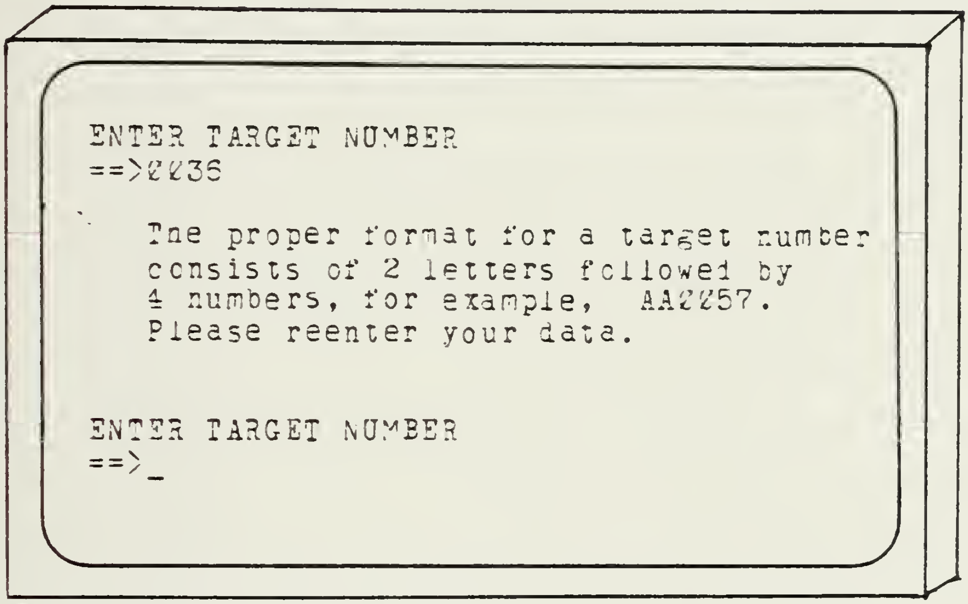


Figure 22. Example of an Error Message.

A central problem in handling errors is in providing the user with the right kind of information. Even experienced users occassionally require assistance in some portion of the system. Accordingly, a user help function has been designed to complement the dialogue and provide the user with additional detailed but concise information on the appropriate input. In some cases, this function is built right into the range of options of the menu, however, at any time the user can receive help or more information by simply

typing a "?". When the system is not sufficiently self-explanatory, it is designed to be self-helping with appropriate and meaningful instructions, advisory messages and error messages.

VII APPLICATIONS PROGRAM IMPLEMENTATION

A. SYSTEM IMPLEMENTATION

Based on the design criteria outlined in chapters IV, V and VI, the target information system was coded, compiled and tested. Each module was coded independently of the other and tested in its own environment. Similarity of the interface was maintained between each module and was the first part of each module coded. After the interface was in place and working, the module was filled out to perform the system functions. After each module was tested and debugged independently, it was incorporated into the system where additional testing and debugging took place. The next module was then coded after the system was functioning properly.

The Interface module is the main system program and contains the global data structures and the system primitive routines such as clearing the screen, skipping lines, and printing error messages. It makes the calls to the segment procedures from a master menu. Another primary function of this module is to build the address map of the records at the start of each session and to open the system data files. This module compiled to 6122 bytes of object code which is four K less than originally expected.

The first of the segment routines is the Inform module. This module contains system operating instructions,

doctrinal explanations of target information terms, target analysis guidelines, security requirements and examples of formats used in the system. It is basically all text and serves to inform the user of the capabilities of the system. The module compiles to 17,300 bytes of object code.

The Initialize module is designed to erase the current data files and completely reinitialize the system. It performs no other function for the system. It builds the data files to the required size and fills the file with empty records. It compiles to 2500 bytes of object code.

The third segment procedure is the Target module. It contains sub-modules for adding a target, deleting a target, changing current target information, displaying a target and adding a BDA to the target record. Major difficulties were encountered in loading this very large module and its involved user interface into main memory with the operating system code and the system Interface module. Initial compilation of the module was to 36,400 bytes. The main problem was the amount of text that the user interface was consuming. One screen frame of user information (advisories, explanations, etc.) usually took 1070 bytes of object code. The expense of so much text in the code was much too great for this module.

Consequently, a decision was made to reduce the size of the module by putting most of the text in separate text files on the diskette and retrieving these files from

secondary storage when called by the program. This produced two unpleasant side effects. First, the interface was slowed considerably since it now took longer to place a user message on the screen (although the user could not read as fast as the output) and second, the diskette now contained numerous text files in addition to the code and data files shown in figure 7.

Since the Pascal system used four blocks of 512 bytes each for a text file no matter how small the file was, the 54 text files took approximately 110 K bytes of secondary storage. This caused the system to employ the second ALTOS disk drive, which had not been used previously. The resultant reconfiguration of the secondary storage allocation is shown in figure 23. Certain menus and recurring messages were retained in the target module to speed the processing and decrease user wait time.

The use of the text files and the reorganization of the BDA routine into a separate segment procedure (called by the Target segment procedure), reduced the module to 17,000 bytes of object code (19,300 bytes when the BDA procedure is called). This proved a satisfactory solution without a significant design change. The direct access capability of the Pascal system proved both accurate and fast with no apparent wait in the process time between a request for a record and a reply to the CRT.

SECONDARY STORAGE

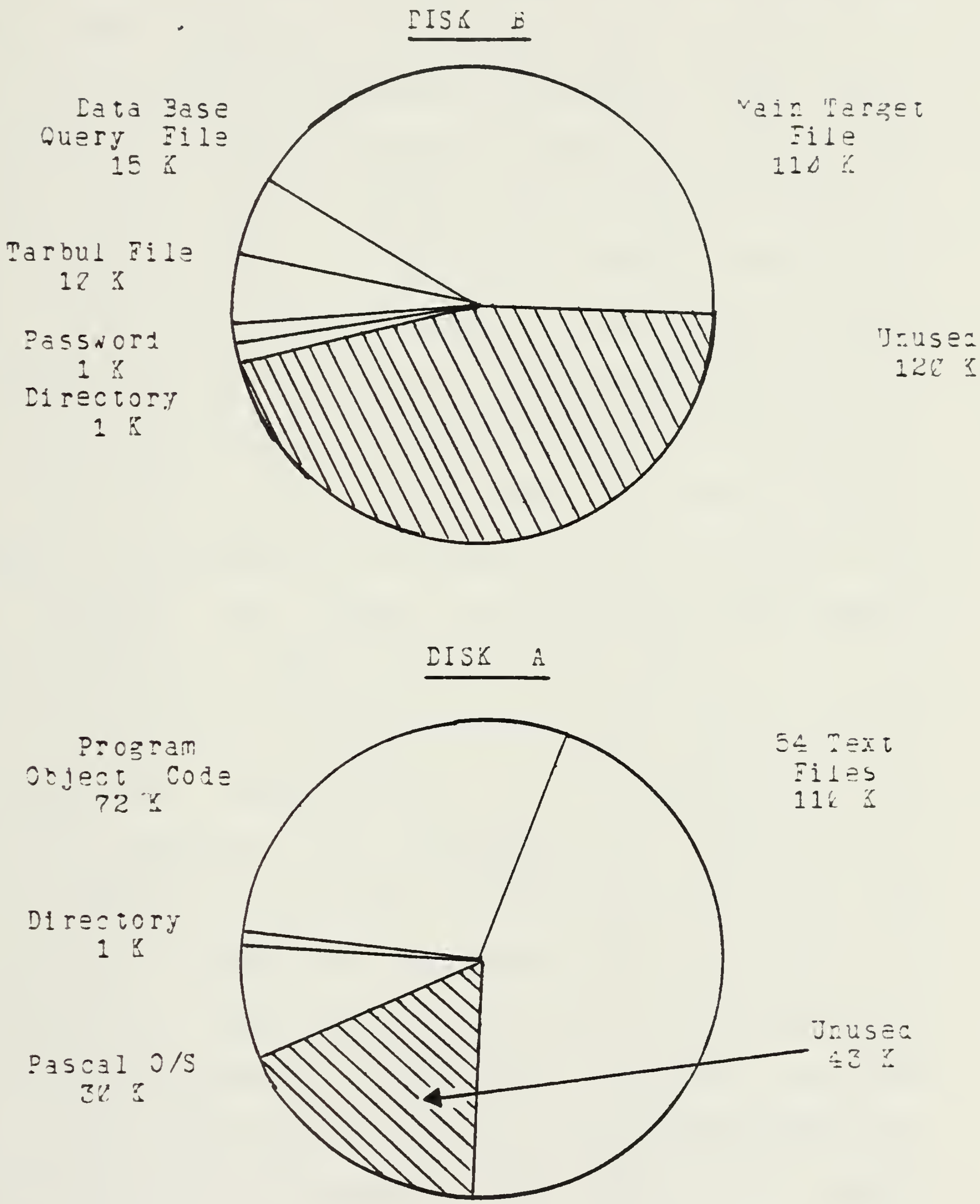


Figure 23. Actual Secondary Storage Usage.

The last module implemented was the Query module. It employed some of the menu text files used by the Target module, thus decreasing its code size, and proved easier to implement than expected. While the query selection is limited to the logical "ANDING" of elements of the domains of the record, the array processing proved to be very rapid and well within the two second time response parameters. Loading of the data base query file was straight forward and changes in the main record, performed in the target module, were being correctly reflected in the data base query file. While initial evaluation of the module proved satisfactory, it is felt that system improvement could be obtained by designing an interface and an algoritnm which allowed both logical "ANDING" and "ORING" of attributes. This segment procedure compiles to 8220 bytes of object code.

The final segment procedure, the Utility module, has not been completely implemented due to programming time constraints. However, the complete user interface is in place and operational and gives the user the impression that the system is operating. The erase file function is operational and works effectively. The following functions have not been implemented: changing the password (requires design of the password record as well as implementation), copying the data base query and target files to a backup diskette, operations on the TARBUL--displaying, renumbering, printing and reinitiating (this also requires design of the

TARBUL record), printing target cards and lists and the computation and display of target file statistics. The current size of the module is 18,300 bytes. As functions are added and the size increases, much of the user interface can be transfered to text files to keep the size of the module at an acceptable level.

The entire program compiles to 72 K of object code, 66 of which is contained in segment procedures. The system source code, which is contained in the Naval Postgraduate School technical report entitled A Prototype Program for Target Information (NPS52-81-007). is over 5200 lines long. Initial testing and implementation was done with a target list size of 100 targets and later expanded to the required 300 target maximum. It has been debugged for execution and tested for operational accuracy. While initial results are very satisfying and the system proves fast and accurate, extensive testing to include field testing would be required before the system could become operational. Additionally, the Utility module would have to be completed.

There are two main concerns in testing the system. First, that it meets the requirements of the target information section and effectively accomplishes its purpose...the automation of the target information functions, and second, that the user interface is as effective as it proports to be. This will require extensive testing and validation as well as operational testing and

evaluation in appropriate tactical command post exercises which employ the division or the MAB fire support coordination center.

B. TECHNICAL CONSIDERATIONS

The Pascal program listed in the technical report is transportable to other UCSD Pascal systems and with modification (random access, segment procedures, strings) to any Pascal system. Certain aspects of the program were implemented to conform to the Datamedia Elite 2500 video terminal. This terminal has 80 characters per line with 24 lines of display with full upper and lower case ASCII operating at a data rate of 9600 baud. It has a 1920 character screen capacity, an alphanumeric keyboard and display and both synchronous and asynchronous interface. Some of the special characters used in the program include:

ASCII	Decimal	Function
SO	14	blink field on
CAN	24	blink field off
BEL	7	bell/beeper
GS	29	roll field on
US	31	clear screen

It is recognized that the design and implementation is based on the ALTOS ACS 8000-1 computer. Advances in microcomputer technology will invariably modify the environment for which the prototype was designed. The addition of hard disk capabilities to microcomputer systems

will increase the secondary storage space as well as the processing speed now available with floppy diskettes. The current system can certainly function effectively in such a new environment but the possibilities of redesign should be considered if the increase in efficiency is warranted and the time span between a new implementation and the introduction of MIFASS is sufficiently long.

Since the UCSD Pascal is available on so many systems, with the proper setup routines the system is sufficiently portable provided the secondary storage capability is available. The source code is compilable on personal computers such as the Apple II when the Pascal system card is included with a 64 K memory although the size of secondary storage (the Apple uses a 5 1/4 inch mini-floppy diskette) will dictate a realignment of files and a further partitioning of system software.

C. TACTICAL CONSIDERATIONS

There are a number of considerations which involve the tactical and operational aspects of target information which bear consideration but which are beyond the scope of this thesis. The first of these is the survivability of the ALTOS and related equipment in the field. Special handling and care is required of each item as well as the system and target diskettes. Specific instructions for the user in the maintenance and handling of the system must be determined

and promulgated to the user. Additionally, power requirements and the resultant equipment modifications or adjustments needed to operate in a field environment must be considered.

The acquisition and use of this equipment may be cause to examine the tables of organization to deteremine the proper staffing of the target information section since a reduction of personnel seems rather easy to achieve. Additionally, a method of transfering data from the Navy ASIS computer to the system microcomputer should be investigated and addressed since transfering the data electronically between the two systems is preferable to manually transfering the data into the microcomputer system. Concept of employment of the system aboard ship (in SACC or in landing force spaces) must also be determined and promulgated.

Once adopted and functional, long range plans must be determined for a transition from the system to the MIFASS system. These, of course, are locally generated requirements and can be addressed in the future. It is anticipated the MIFASS will require extensive operator training before it becomes operational whereas the microcomputer prototype system requires only user familiarization. It is estimated that the user can become proficient with this system after a one hour familiarization period.

The requirements for maintaining a graphical display of targets, while currently solvable with computer graphics, will continue to be accomplished by the use of tactical battle maps covered with acetate overlays where the situation dictates. This system makes no attempt to address the graphic display of targets. It is felt that the display capabilities of the MIFASS system will adequately satisfy the requirements when the system is introduced into the fleet.

D. SYSTEM REFINEMENTS

While the system meets all the requirements identified for the target information section, there are a number of refinements to the program which could be implemented in a later version of the prototype which would enhance the system performance and provide additional capabilities to the TIO. These include the following:

1. Expanding the size of the target file beyond the current 300 target maximum if the tactical considerations of future battle scenarios dictate.
2. Logical "ORING" and "ANDING" of record attribute domains in the Query module to provide a more refined selectivity of the special target lists.
3. Inclusion of a utility routine in the query module to print the target list obtained as a result of the query.

4. Allow the user to specify altitude in feet, yards or meters and have the system convert the data to meters.

5. Find the number and designations of targets within a certain radius or distance of another target (for example, a class E target) or a known grid reference point.

6. Provide a routine for keeping track of the next available target number from the FSCC block of target numbers.

7. Modification of the UCSD Pascal operating system command prompt to allow the user only one choice of option, i. e., to run the system program and remove the filer subroutine from his environment.

8. Plotting of artillery firing units and naval gunfire support stations to determine automatically, which targets are within the effective range of specific supporting arms.

9. Reduction of coding by improved algorithms and subroutines.

10. Provide operator training and system maintenance manuals for the system.

VIII CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

This thesis has contended that the operational effectiveness of the fire support coordination center would be improved by the automation of the target information function. Further, it appears that the design and implementation of a suitable and effective system is possible now, five full years before the introduction of the MIFASS computer system into the Fleet Marine Forces.

The thesis has presented one such design using a data base approach on a typically configured, commercially available microcomputer with a user interface specifically designed for the Marine performing the target information functions in an operational environment. An evaluation of the implemented prototype MIcrocomputer System for Target Information (MISTI) has determined that the requirements and specifications for the system as described in chapters II and III have been met and that the program operates effectively and efficiently.

The basic soundness of the design is reflected in both the operational effectiveness of the prototype and the lack of significant changes or modifications needed to meet the stated system requirements. The working prototype, if employed in an FSCC in its present state, would immediately

increase the operational effectiveness of the target information section.

B. RECOMENDATIONS

The design and implementation has been shown to be sound and, based on the observed effectiveness of the prototype program, the following recommendations are made:

1. That the implementation of the prototype MIcrocomputer System for Target Information (MISTI) be continued in accordance with the design criteria outlined herein and the system refinements discussed in chapter VII.

2. That the resultant system be tested and evaluated at selected Marine Corps commands to determine its effectiveness in actual tactical operations.

3. That the Marine Corps adopt the MIcrocomputer System for Target Information (MISTI) on an interim basis until the introduction of the MIFASS system.

4. That appropriate hardware and software be provided to the three Marine Division fire support coordination centers in order to employ this system.

5. That the Marine Corps Tactical Software Support Activity (MCTSSA) evaluate the MIcrocomputer System for Target Information (MISTI) as a testbed model for the software interface criteria of the target information portion of the MIFASS system.

APPENDIX A - DATA DICTIONARY

Records: 2

Relationships: 1

Record: Target

Primary key: target number

Secondary keys: classification, priority, accuracy, grid location, supporting arm assigned, type, status

Retrieval keys: target number, grid location

Repeating groups: BDA (DTG surveillance, firing unit, no/type rounds, damage assessed, damage reported, BDA) 3

Cardinality: 300

Degree: 22

Attributes length: 239 bytes in basic record
66 bytes for each extra BDA
371 bytes full record count

Value set size: 150

Length: fixed

Record: Target Query

Primary key: target number

Secondary keys: classification, priority, accuracy, status, type, supporting arm assigned

Retrieval key: target number

Repeating groups: none

Cardinality: 300

Degree: 9

Attributes length: 43 bytes

Value set size: 132

Length: fixed

Attribute list: Target Record and Target Query Record
(Target Query Record attributes indicated by *)

ATTRIBUTE	DESCRIPTION	STORED AS	DOMAIN
*Target number	2 char, 4 int	6 char	AA0001-ZZ9999
*Grid location	8 int	8 char	numerical
*Altitude	4 int	4 char	0002-9999
*Description	40 char	40 char *20 char	string
Remarks	40 char	40 char	string
*Classification	1 char	1 char	A, B, C, D, E
*Type	4 char	1 char	TANK, MISC, SEAD, INST, OP, TERR, VEH, FORT, CBAT
*Priority	3 char	1 char	I, II, III, IV
*Status	3 char	1 char	ACTIVE/INACTIVE
*Supporting arm assigned	4 char	1 char	AIR, ARTY, NGF OTHR, NONE
DTG surveillance	6 int, 1 char	7 char	010001A-312359Z
Firing unit	6 char	6 char	string
No./type rounds	10 char	10 char	string
Damage assessed	11 char	1 char	
Damage reported	11 char	1 char	Interdicted, neutralized, illuminated, damaged, destroyed, unknown, unobserved,

		harassed	
BDA	40 char	40 char	string
Map reference	20 char	20 char	string
Source of tgt	20 char	20 char	string
Photo number	15 char	15 char	string
Photo coordinates	8 int	8 char	numerical
DTG activated	6 int, 1 char	7 char	010001A-312359Z
Target accuracy	9 char	1 char	Possible, probable, confirmed, unknown

APPENDIX B - EXAMPLE OF SYSTEM USER INTERFACE

This appendix illustrates the menu selection format of the user interface explained in detail in the preceeding chapters. It simulates the user operating the query module and forming a list of targets for a special listing. The fire support coordinator has asked the target information officer for a list of targets which are to be considered in the formulation of a flak suppression fire plan prior to the calling of close air support aircraft on an important landing force target.

Essentially, the list must include all targets from the following categories:

Type..............SEAD targets

Classification....all classes

Priority..........I

Status............active targets

Accuracy..........all categories

Supporting arm....artillery

The example simulates the user interface of actual system operation to complete this query as specified above. Each page represents a separate frame observed by the user on the CRT screen.

```
        SPECIAL TARGET LISTINGS
..........................................................

    The options are:

        1.  Form a special target listing
        2.  Continue to process
        3.  Write the special list to the screen
        4.  Information about this procedure
        5.  Return to previous menu

      PLEASE ENTER OPTION NUMBER
      ==>_
```

The user enters option 1

```
Categories for Special Listing
.......................................................

The listing can contain 6 items from the below menu:

        1.  Target type
        2.  Classification
        3.  Supporting arm assigned
        4.  Priority
        5.  Accuracy
        6.  Status
      * P.  Process information

  Special list currently contains Ø targets.
        Please start a new listing.

PLEASE ENTER OPTION NUMBER
==>_
```

User enters option 1 to select the target type

```
ENTER TARGET TYPE

 The options are:

    1.  Tank
    2.  SEAD target
    3.  Installation
    4.  Counter Battery
    5.  Observation Post
    6.  Terrain
    7.  Vehicles
    8.  Fortifications
    9.  Miscellaneous

PLEASE ENTER OPTION NUMBER
==>_
```

User enters a 2 for the SEAD targets

```
               Categories for Special Listing
.....................................................

    The listing can contain 5 items from the below menu:

            1.  Target type                Already Taken
            2.  Classification
            3.  Supporting arm assigned
            4.  Priority
            5.  Accuracy
            6.  Status
          * P.  Process information

       Special list currently contains 42 targets.

     PLEASE ENTER OPTION NUMBER
     ==>_
```

User enters option 3 to select the supporting arm

```
ENTER SUPPORTING ARM ASSIGNED TO TARGET

    The options are:

            1.  ARTY
            2.  NGF
            3.  AIR
            4.  AIR, ARTY
            5.  AIR, NGF
            6.  ARTY, NGF
            7.  AIR, ARTY, NGF
            8.  Other
            9.  None

 PLEASE ENTER OPTION NUMBER
 ==>_
```

User enters option 1 to select artillery

```
              Categories for Special Listing
..............................................................

  The listing can contain 4 items from the below menu:

              1.  Target type              Already Taken
              2.  Classification
              3.  Supporting arm assigned  Already Taken
              4.  Priority
              5.  Accuracy
              6.  Status
            * P.  Process information

     Special list currently contains 29 targets.

     PLEASE ENTER OPTION NUMBER
     ==>_
```

User enters option 4 to select the target priority

```
  ENTER TARGET PRIORITY

  The options are:

     1.  I
     2.  II
     3.  III
     4.  IV

PLEASE ENTER OPTION NUMBER
==>_
```

User selects option 1 for priority I targets

```
            Categories for Special Listing
.........................................................

The listing can contain 3 items from the below menu:

            1.  Target type                  Already Taken
            2.  Classification
            3.  Supporting arm assigned      Already Taken
            4.  Priority                     Already Taken
            5.  Accuracy
            6.  Status
          * P.  Process information

     Special list currently contains 16 targets.

    PLEASE ENTER OPTION NUMBER
    ==>_
```

User enters option 6 to select the target status

```
ENTER TARGET STATUS --ACTIVITY

 The options are:

    1.  Active
    2.  Inactive

  PLEASE ENTER OPTION NUMBER AND PRESS RETURN
  ==>_
```

User selects option 1 for active targets

```
               Categories for Special Listing
.............................................................

   The listing can contain 2 items from the below menu:

           1.  Target type                 Already Taken
           2.  Classification
           3.  Supporting arm assigned     Already Taken
           4.  Priority                    Already Taken
           5.  Accuracy
           6.  Status                      Already Taken
         * P.  Process information

      Special list currently contains 10 targets.

     PLEASE ENTER OPTION NUMBER
     ==>_
```

User has completed the query and now elects to
process the list of 10 targets

```
SPECIAL TARGET LISTINGS
.................................................................

The options are:

    1.  Form a special target listing
    2.  Continue to process
    3.  Write the special list to the screen
    4.  Information about this procedure
    5.  Return to previous menu

  PLEASE ENTER OPTION NUMBER
  ==>_
```

The user selects option 3 to display the listing

SPECIAL TARGET LISTING

Categories: SEAD ACTIVE Pri I ARTY

TGT NO	CL	PRI	LOCATION	ALT	SA	DESCRIPTION
AA0046*	A	I	35647582	100	ARTY	2 ZSU-23 PLT
AA0057*	C	I	35452353	60	ARTY	SA-6 CLUSTER
AA0078	A	I	35467787	50	ARTY	12.5 AAA SITE
AA0156*	A	I	35667746	120	ARTY	S-60 PLT IN OPEN
AA0122*	D	I	35334563	25	ARTY	SA-9 PLT IN TREES
AA0144*	B	I	35674664	50	ARTY	14.5 AAA SITE
AA0167	A	I	35455234	100	ARTY	ZU-23 AAA CLUSTER
NA0023*	D	I	34556867	20	ARTY	120 MM AAA CANNON
AA0188*	C	I	34567890	150	ARTY	SA-8 IN BUNKERS
AA0194*	A	I	36087646	45	ARTY	S-60 AAA CLUSTER

NOTE: * indicates target list

PLEASE PRESS RETURN TO CONTINUE
==>_

The user presses RETURN to continue

```
SPECIAL TARGET LISTINGS
...............................................................

The options are:

      1.  Form a special target listing
      2.  Continue to process
      3.  Write the special list to the screen
      4.  Information about this procedure
      5.  Return to previous menu

   PLEASE ENTER OPTION NUMBER
   ==>_
```

The user begins a new query or returns to the main menu

BIBLIOGRAPHY

Bowles, K. L., Microcomputer Problem Solving Using Pascal, Springer-Varlag, 1977.

Chamberlain, R. G., Conventions for Interactive Computer Programs, Interfaces, Nov 1975.

Chapman, J. A., Logical Data Base Design for Relational Data Base Systems, Masters Thesis, Naval Postgraduate School, 1978.

Clautero, R. M., Generalized Approach for Evaluating Data Base Organization and Indexing Methods, Masters Thesis, Naval Postgraduate School, 1975.

Codd, E. F., Seven Steps to Rendezvous with the Casual User, Proceedings of the IFIP TC-2 Working conference on DBMS, North Holland-Amsterdam, 1974.

Conrad, M., Pascal—A High-level Language for Micros and Minis, Datamation, July 1979.

Coulter, R. J., A Prototype Program for Target Information, Technical Report NPS52-81-007, Naval Postgraduate School, June 1981.

Date, C. J., An Introduction to Database Systems, 2nd Edition, Addison-Wesley Publishing Company, 1977.

Dahmke, M., The Altos ACS 8000 Single Board Computer, Byte Magazine, McGraw Hill, v. 5, no. 11, Nov 1980.

Dobie, R. B., Information Storage and Retrival System: A DBMS for a Microcomputer, Masters Thesis, Naval Postgraduate School, 1976.

Engle, S. E. and Granda, R. E., Guidelines for Man/Display Interface, IBM Technical Report, 1975.

Fleet Combat Direction System Support Activity San Diego, Quest System: Program Operator Manual ASIS, 1974.

Fleet Marine Force Manual (FMFM) 7-1, United States Marine Corps, Fire Support Coordination, United States Government Printing Office, 1978.

Gagle, M., Koehler, G., and Whinston, A., Data Base Systems and Micro-Computers: An Overview, Purdue University, August, 1979.

Grogono, P., Programming in Pascal, Addison-Wesley, 1978.

House, W. C., Data Base Management, Petrocelli Books, 1974.

Institute for Information Systems, University of Southern California at San Diego, UCSD (Mini-Micro Computer) Pascal, Release Version 1.4, 1978.

Jensen, K. and Wirth, N., Pascal: User Manual and Report, 2nd ed., Springer-Verlag, 1974.

Kahn, B. K., A Method for Describing Information Required by the Database Design Process, Proceedings of SIGMOND, ACM, 1976.

Kennedy, T. C. S., The Design of Interactive Procedures for Man-Machine Communication, International Journal of Man-Machine Studies, 1974.

Kroenke, D., Database Processing, Science Research Associates, Inc., 1977.

Lewis, T. G. and Smith, M. Z., Applying Data Structures, Houghton Mifflin Company, 1976.

Headquarters, Unites States Marine Corps, Marine Tactical Command and Control Systems (MTACCS) Master Plan, Washington, D. C., October 1979.

Marine Corps Tactical Systems Support Activity (MCTSSA), Marine Integrated Fire and Air Support System (MIFASS) Specifications, ELEX-M-296C, 1980.

Martin, J., Design of Man-Computer Dialogue, Prentice Hall, 1973.

Martin, J., Computer Data-Base Organization, Prentice Hall, 1975.

Naval Warfare Publication (NWP 22-2), Department of the Navy, Supporting Arms in Amphibious Operations, Naval Warfare Publications, 1973.

Ramsey, H. R., Atwood, M.E., and Kirshbaum, P.J., A Critically Annotated Bibliography of the Literature on Human Factors in Computer Science, Scientific Applications Inc., 1978.

Ross, R.G., Data Base Systems: Design, Implementation and Management, AMACOM, 1978.

Senan, A., and Sihombing, T. M., Database Management System for Microcomputers, Masters Thesis, Naval Postgraduate School, 1979.

Shneiderman, B., Improving the Human Factors Aspect of Database Interactions, ACM Transactions on Data Base Systems, 1978.

Shneiderman, B., Software Psychology: Human Factors in Computer and Information Systems, Winthrop Publications Inc., 1980.

Slesser, R. M., One County's Experience with Data Base, Department of Defense Computer Institute Selected Computer Articles, 1980.

Smith, D. L., Method to Evaluate Microcomputers for Non-tactical Shipboard Use, Masters Thesis, Naval Postgraduate School, 1979.

Smith, L. B., The Use of Interactive Graphics to Solve Numerical Problems, Communications of the ACM, 1970.

Snodgrass, R., A Sophisticated Microcomputer User Interface, Procedings of the Third Symposium on Small Systems, 1980.

Tsichritzis, D. C. and Lochovsky, F. H., Data Base Management Systems, Academic Press, Inc., 1977.

Ullman, J. D., Principles of Data Base Systems, Computer Science Press, 1980.

Weiderhold, G., Database Design, McGraw-Hill, 1977.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943	2
3.	Department Chairman, Code 52 Department of Computer Science Naval Postgraduate School Monterey, California 93943	1
4.	Professor Lyle A. Cox, Jr., Code 52Cl Department of Computer Science Naval Postgraduate School Monterey, California 93943	1
5.	Commanding General Attn: Fire Support Coordinator 1st Marine Division, FMF Camp Pendleton, California 92055	1
6.	Commanding General Attn: Fire Support Coordinator 2nd Marine Division, FMF Camp Lejeune, North Carolina 28542	1
6.	Commanding General Attn: Fire Support Coordinator 3rd Marine Division, FMF FPO San Francisco 96602	1
7.	Commanding General Education Center Attn: Supporting Arms Branch MCDEC Quantico, Virginia 22134	1

8. Commanding General 1
Development Center
MCDEC
Quantico, Virginia 22134

9. Commanding General 1
Attn: Fire Support Section (TEC)
MCAGCC
Twentynine Palms, California 92227

10. Marine Corps Representative 1
U. S. Army Artillery School
Fort Sill, Oklahoma 73503

11. TRADOC Research Element Monterey, Code TREM 1
Naval Postgraduate School
Monterey, California 93940

12. Commanding Officer 1
MCTSSA
Attn: MIFASS Team
Camp Pendleton, California 92055

13. LtCol Ronald J. Coulter, USMC 1
8033 Burnside Landing Drive
Burke, Virginia 22015

Thesis
C75667
c.1

192523

Coulter

A microcomputer system for target information in the fire support coordination center: a data base approach.