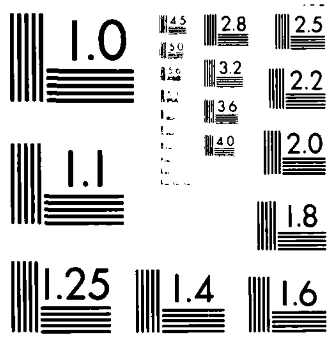
AD-A156 827 A DESIGN METHODOLOGY AND PROTOTYPE FOR THE RESERVE TRAINING SUPPORT SYSTEM(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA M J WINSLOW ET AL. MAR 85 1/2

UNCLASSIFIED F/G 5/9 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

A DESIGN METHODOLOGY AND PROTOTYPE
FOR THE
RESERVE TRAINING SUPPORT SYSTEM

by

Michael J. Winslow
and
Howard C. Seeger

March 1985

Thesis Advisors: Ronald B. Kurth
Michael Spencer
Thomas Swenson

Approved for public release; distribution is unlimited

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 Design Methodology and Prototype for the Reserve Training Support System

5. TYPE OF REPORT & PERIOD COVERED
Master's Thesis
March 1985

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Michael J. Winslow
Howard C. Seeger

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Naval Postgraduate School
Monterey, CA 93943

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Naval Postgraduate School
Monterey, CA 93943

12. REPORT DATE
March 1985

13. NUMBER OF PAGES
123

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 is 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)
Reserve Training Support System
Database Management System
ORACLE
prototyping
fourth generation language

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
There is much to be done to update the Naval Reserve Automated Information Systems to 1980's technology. This thesis analyzes the Naval Reserve's automated information system - the Reserve Training Support System (RTSS). It presents the systems background and specifications, its problems and the Reserve's own plan to resolve these problems. After a discussion on the characteristics of an effective automated information system, RTSS is critiqued. The key issue is the obsolescence of (Continued)

ABSTRACT (Continued)
of the hardware and software being used. To alleviate their information problems, the Reserves must develop a plan to implement new hardware and software technology in a coordinated fashion. An alternative is presented to the traditional system development process: prototyping. Relational database theory is briefly discussed. ORACLE, a relational database management system, is used to implement the database prototype that serves as an example of how current technology can be used to eliminate many of the Naval Reserve's information problems.

Accession For
NTIS GRA&I X
By
Dist
A/1

A Design Methodology and Prototype for the Reserve Training Support System

by

Michael J. Winslow
Lieutenant Commander, United States Navy
B.A., Holy Cross College, 1974

and

Howard C. Seeger
Lieutenant, United States Navy
B.S., United States Naval Academy, 1978

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
March 1985

Authors: Michael J. Winslow

Howard C. Seeger

Approved by: Ronald B. Kurth, Thesis Advisor

Michael Spencer, Co-advisor

Thomas Swenson, Co-Advisor

Willis R. Greer, Chairman,
Department of Administrative Sciences

Kneale T. Marshall,
Dean of Information and Policy Sciences

ABSTRACT

There is much to be done to update the Naval Reserve automated information systems to 1980's technology. This thesis analyzes the Naval Reserve's automated information system- the Reserve Training Support System(RTSS). It presents the system's background and specifications, its problems and the Reserve's own plan to resolve these problems. After a discussion on the characteristics of an effective automated information system, RTSS is critiqued. The key issue is the obsolescence of the hardware and software being used. To alleviate their information problems, the Reserve's must develop a plan to implement new hardware and software technology in a coordinated fashion. An alternative is presented to the traditional system development process: prototyping. Relational database theory is briefly discussed. ORACLE, a relational database management system, is used to implement the database prototype that serves as an example of how current technology can be used to eliminate many of the Naval Reserve's information problems.

TABLE OF CONTENTS

I. INTRODUCTION 9

II. BACKGROUND/EVOLUTION OF RTSS 12

A. HISTORY 12

B. RTSS 16

1. RTSS (Air) 17

2. RTSS (TM) 19

C. APPLICATIONS SOFTWARE 21

1. RTSS (Air) 21

2. RTSS (TM) 22

D. SYSTEM SOFTWARE ENVIRONMENT 24

E. SYSTEM HARDWARE ENVIRONMENT 25

F. PROGRESS TO DATE 25

III. AUTOMATED INFORMATION SYSTEMS 30

A. INTRODUCTION: THE VALUE OF INFORMATION 30

B. PROBLEMS 32

1. Naval Reserve 32

2. Navy 35

C. STRATEGIES 36

D. CURRENT TECHNOLOGY 39

E. SYSTEMS DEVELOPMENT 42

IV. THE PROTOTYPE 49

A. INTRODUCTION 49

B. NORMALIZED FORMS 50

1. Modification Anomalies 50

2. Data Duplication 53

3. First Normal Form 54

4. Second and Third Normal Form 55

5. Fourth and Fifth Normal Forms 57
6. Tradeoffs 58
C. ORACLE DATABASE MANAGEMENT SYSTEM 61
1. The System 61
2. System Features 61
3. System Components 62
4. SQL 64
D. DATA SECURITY 69
1. Unauthorized Access 69
2. Navy ADP Security Program 70
3. Database Access 70
4. Data Access 72
E. DATA INTEGRITY 77
1. Transaction Validation 77
2. Format Checks 78
3. Reasonableness Checks 79
F. IMPROVING SYSTEM PERFORMANCE 80
1. Indexing 80
2. Clustering 81
G. RECOVERY 83
1. After Image Journal 83
H. REPORT WRITING 84
I. ORACLE'S GRADE 85

V. CONCLUSIONS AND RECOMMENDATIONS 87
A. CONCLUSIONS 87
B. RECOMMENDATIONS 89

APPENDIX A: DATA ELEMENTS DICTIONARY 92

APPENDIX B: INTERACTIVE APPLICATIONS GENERATOR OUTPUT 95

APPENDIX C: PERSONAL INFORMATION SCREEN 100

APPENDIX D: PROTOTYPE INDICES 120

LIST OF REFERENCES 121

INITIAL DISTRIBUTION LIST 123

LIST OF FIGURES

2.1	RISS Geographical Distribution	26
3.1	Traditional Development Process	44
4.1	Modification Anomalies(uncorrected)	51
4.2	Modification Anomalies (corrected)	52
4.3	Data Duplication (uncorrected)	53
4.4	Data Duplication (corrected)	54
4.5	First Normal Form	56
4.6	Second Normal Form (uncorrected)	57
4.7	Second Normal Form (corrected)	57
4.8	Multivalued Relationships	59
4.9	ORACLE Component Block Diagram	63
4.10	Non-Procedural Language	65
4.11	ORACLE Query Function	66
4.12	Data Definition Language	66
4.13	Data Manipulation Language	67
4.14	Data Control Language	69
4.15	User Authorization(Log on) Procedures	71
4.16	Three-level Privilege Structure	75
4.17	Alternative View of Data	76
4.18	Indexing	81
4.19	Clustering	82

I. INTRODUCTION

In this thesis we present an approach for solving a portion of the ADP problems of the Naval Reserve. We have a two-fold purpose in this effort.

First, we have a strong desire to produce a usable product. We want to do something functional, while satisfying the academic requirement for a thesis. To that end, we called on the experiences of our advisors with Automated Information Systems (AIS) in the Naval Reserve; specifically, their experience with the Reserve Training Support System (RTSS). They believed that much remained to be done in updating the Reserve Information Systems to 1980's technology. Second, we both have a strong desire to use a fourth generation relational language. One of these commercial off the shelf products, ORACLE, is installed on the Computer Science Departments VAX 11/780 computer at the Naval Postgraduate School and is available for our research. The factor bringing the two together became evident early in our research when we discovered that ORACLE could run on the equipment being installed for RTSS, given an operating system conversion or change.

In gathering the appropriate background information for our topic we were immediately presented with larger issues. The concepts of strategic policy and organizational philosophy in Information Systems management must first be addressed and resolved by an organization before a new computer system will be effective. To have discussed just the one system in question without discussing the issue of the present and future status of Information Management for the Reserves would have been myopic; too many questions would have been left unanswered.

Moreover, RTSS must not be seen merely in the short sighted light of separate Reserve organizational issues. Certainly the Reserves have unique mission requirements; they are presently developing their own unique architectures to meet advancing Information System Management issues head on. But Reserve issues are Navy issues, Navy problems are Reserve problems, and so on. Indeed, we believe our research will show that many of the problems the Reserves face today have arisen from the lack of a clear delineation of where the Reserves and the Navy's AIS needs differ.

Thus our treatise progresses from background and specifics on RTSS, to a wider scope of AIS problems and issues facing the Reserves, and back to more RTSS specifics. We present kindred problems and issues for the parent Navy using the vehicle of a CNO ordered blue ribbon panel of experts recruited by the National Science Foundation. We then address present and academic design/redesign strategies for the Naval Reserve, and introduce our choice for attacking this problem of considerable scope. Finally, we invested a large amount of time learning ORACLE, and offer this prototype as an attempt to starting afresh. Our prototype is only an example; it is illustrative of what can be accomplished with a fourth generation relational system. We think our problem solving approach fits soundly with the recommendations of independent studies of those larger issues.

Chapter One is an overview of the thesis, with brief summaries of the chapters and the thoughts behind our overall approach. It knits the far-reaching research efforts towards a meaningful conclusion.

Chapter Two is a background investigation of RTSS(Air), with a brief look at a concurrent and partially redundant system in the RTSS family, RTSS(Training Management). We look at the intended purposes and surrounding policies of the system as it has grown. We present the hardware and software environments for the system, and then address the progress and problems in the Reserves' implementation efforts.

Chapter Three is the transition in which we focus on the concept of information as a resource; the ways the Navy, the Reserves, and specifically RTSS, have dealt with this issue are presented. We note some exciting and ambitious plans for the future as put forth in the latest information systems strategic plans for the Navy and the Reserves. We then transition through brief discussions of implementing current technology and systems development methodologies to the prototyping concept.

Chapter Four is the Prototype. In an effort to bring the reader who is unfamiliar with relational databases onboard, we have included a brief discussion of some germane terms and concepts at the start. We then present a discussion of ORACLE and how we used its features to implement our prototype design. We have taken some rather large manuals for using ORACLE and condensed them into a simplified approach to what the system is capable of doing, and what we did with it.

Chapter Five is our final conclusions and recommendations. Included are recommendations for the Naval Air Reserve and the path it should take. Finally, we discuss the next steps to be taken in the development of the prototype.

E. SYSTEM HARDWARE ENVIRONMENT

The DEC family of PDP-11 computers is used at RTSS sites in New Orleans and other non-ATSS sites. The PDP-11/70 is at New Orleans, and either PDP-11/23 or 11/50 CPUs are at the smaller sites; specific details of each site configuration are available in the Functional Descriptions of the RTSS systems. The follow-on generation to the PDP-11, namely the VAX-11/780, is being requested in the POM 87 for the New Orleans RTSS(TM) site. Code 46 (ADP Plans) at CNAVRES is guiding the purchase under the umbrella of Navy ADP purchasing requirements.

F. PROGRESS TO DATE

RTSS(Air) presently consists of four Commander Naval Air Reserve Forces (COMNAVAIRESFOR) central sites, each of which support a number of satellite sites. The central (host) sites are geographically spaced to serve the largest number of aviation activities. In addition to the central and satellite sites, Reserve aviation activities are co-resident on seven Regular ATSS sites. 95% of all participating units were to be supported by the end of calendar year 1984 by either RTSS(Air) or ATSS. [Ref. 6: p.1]

Owing to its borrowed and aged nature, RTSS(Air) has several drawbacks to effective implementation in the field. Since it is tied so closely to ATSS, RTSS implementation suffers any problems that ATSS is susceptible to. NAVAIR support of the ATSS system has finally increased to a staff of three after many years of requests. Moreover, the initial premise in the RTSS (Air) Functional Description was that no software modification costs would be incurred by the Reserves because of the link to ATSS. This is faulty on two counts. First, while the Reserves wait for updated software instead of updating it themselves, they incur the

CANTRAC and MCRF courses, and create, delete and review capability for non-standard courses.

j) Readiness--allows for IRAD billet input, update, and review by unit, Reserve Program Number (RPN), rating, or designator, and generates diary entries for posting to IMAPMIS. It is linked to the Billet Maintenance subsystem.

k) Mobilization--the primary communications vehicle for reporting quantitative unit mobilization data and unit status information in the event of an actual mobilization or mobilization exercise. It is a subset of the files created by the Billet and Active Activity files, and is supposed to be updated after every billet update.

As of December 1984, the only subsystems fully operational were MATTS, ARTS, Readiness, and Ready Mariner.

D. SYSTEM SOFTWARE ENVIRONMENT

Computer programs in RTSS systems must interact with the DEC Resources Sharing Timesharing System/Extended (RSTS/E) operating system. RSTS/E is designed to allow up to 64 simultaneous processes[2] to interactively access large amounts of data. It dynamically allocates processor time, memory space, file space, and peripherals to suit the changing demands of the system. Development to date has been in the BASIC-PLUS and BASIC-PLUS-TWO programming languages; RSTS/E can also support COBOL, FORTRAN IV, APL, and RPG II language processors.

---

[2]In a single processor environment such as this, the term "simultaneous processes" means that the system can handle separate user processes with no appreciable delay such that the user thinks he is the only one on the system. In actuality, if a large number of users were to attempt to run simultaneous processes, there would be a delay on the order of several seconds.

order writing systems for accurate data. It is presently independent of RTSS(Air) Personnel software.

b) Billet File Maintenance--allows for review of individual or unit billets, and updates of Individual Readiness Assessment Designator codes for either individuals or units. It is supposed to be updated via IMAPMIS, and maintained via MATTS.

c) Activity File Maintenance--allows for review of reserve and active activities by different codes (RUIC,APC,AUIC), and update by RUIC of reserve activity files. It is to be updated monthly by IMAPMIS.

d) Mobilization Assignment Trainee Tracking System (MATTS)--records gains, losses, transfers, and reassignments via the three previous subsystems.

e) Ready Mariner--tracks the flow of Ready Mariner personnel through the Naval Reserves.

f) Variable Report Generator(Query)--enables users to develop and generate reports or requests for information from the database on demand.

g) Fixed Reports--enables the user to generate certain preprogrammed fixed reports, such as readiness and mobilization reports which cannot be met through Query.

h) Utilities--a collection of software modules for automating labor intensive manual operations, such as producing mailing labels.

i) ACDUTRA Request Tracking System (ARTS)--provides authorized users access to the RTSS(TM) ACDUTRA application files to create, update, review and delete applications and to generate approval, disapproval, cancellation, and quota request letters concerning these applications. It is also to provide access to the Course Files with review capability for standard

This sub-system was not operational as of June 1984. It will utilize two separate software components:

i) Enlisted Training Software--individual training records and schedules for training syllabi.

ii) Aircrew Software--individual aircrew training records, schedules for student training with start/stop dates, and long term trend analyses to detect deficiencies in the training program itself.

g) Resource Scheduling Software--types, quantities, and status of training resources. It matches training resources to training needs.

h) Resource and Configuration Accounting Software--monitors status and configuration of training aircraft resources. It provides the capability for individual aircraft maintenance records, and generates all required periodic maintenance reports.

At present, only the Personnel and Resource & Configuration Accounting Software are operating. The Testing software is close to implementation. Another subsystem not directly included in the original Functional Description, Reserve Training Tracks, is presently being developed to map billet training requirements to an individuals background and then schedule him/her for available schools in an individual training track.

2. RTSS (TM)

The RTSS (TM) software consists of eleven system functions, [Ref. 3: pp.3-5 to 3-18], in various stages of development at this time:

a) Personnel File Maintenance--allows for review and update of records by individual or by unit. This subsystem is linked to IMAPMIS, MATTS, and the ACDUTRA

C. APPLICATIONS SOFTWARE

1. RTSS (Air)

The applications software for RTSS (Air), as derived from the original ATSS programs, is divided into the following categories: Personnel Qualifications, Report Generation, Test and Evaluations (TEVS), Flight Data, Personnel Scheduling, Enlisted Training, Aircrew Training, and Resource Configuration and Scheduling (RCAS) [Ref. 2: p. 1]. A brief explanation of each follows:

a) Personnel Software--the cornerstone of the RTSS (Air) database, with personal identification and training history data for individuals.

b) Personnel Qualification Standards (PQS)/Qualification Software--outputs listings of personnel qualification for specific tasks and the expirations of those qualifications; not operational at this writing.

c) Report Generation Software--a variable report generator to provide users with the capability to retrieve records based on a specified selection criterion, sort or arrange these records in any desired sequence, and generate reports with selected data items from the record.

d) Test and Evaluation Software--provides the capability to create, update, review, and delete individual test items and generate printed tests, and provide optical scoring and statistics generation. This sub-system is not operational at this writing.

e) Flight Data Software--allows the entry of yellow sheet data for statistical reports on aircraft and aircrew flight times.

f) Personnel Scheduling Subsystem--matches a student's education and experience with the needs of the service and the available training paths to meet those needs.

with the CNAVRES RTSS (TM) centralized data base and other RTSS (Surface) units.

Seven years ago, the long term goal was for total integration of RTSS (Surface), RTSS (Air), and RTSS (TM) into a consolidated and centralized database to provide real-time information for personnel, mobilization, recruiting, readiness and training management. At present, the RTSS (Air) and RTSS (TM) are two separate systems running on two separate PDP 11/70 computers at New Orleans. Personnel data from the field must be entered twice in order to keep data on both systems. Much of the data, and resultant inaccuracy, for RTSS (TM) has come from its dependence on the Inactive Manpower And Personnel Management Information System (IMAPMIS) for billet and mobilization queries. IMAPMIS is acknowledged to be inadequate, and is presently under extensive redesign[1] [Ref. 4: p.1-14].

Informal correspondence with the RTSS coordinator at New Orleans indicated a hopeful union of the three RTSS systems within a year by the new contractor, Martin-Marietta Corp. At the same time, however, he acknowledged that once the DEC/VAX 780 equipment arrives, all the software would need to be designed away from the RSTS/E operating system, and a database more powerful than the locally designed but obsolete one in use would need to be implemented. [Ref. 5]

---

[1] See also Chapter 3: PROBLEMS, Naval Reserve, under the Logistics/Shore Support Programs discussion, and STRATEGIES, under the SDS discussion.

h) Reduction of administrative and clerical workload of field, staff, and operating units.

2. RTSS (TM)

The RTSS (TM) was designed to support training management, mobilization assignment, readiness measurement, and mobilization and readiness reporting. Like RTSS (Air), it was to have the capability to create an individual's record when initially affiliated with the Naval Reserve and then to follow him/her throughout the reservist's entire service with the Naval Reserve. The long term objectives of the system, [Ref. 3: pp. 2-2,2-3], overlap those of RTSS (Air) considerably:

a) Increasing the quantity and quality of Selected Reserve mobilization billet assignment capability at all command levels.
b) Integrating personnel and training record data under a single system accessible from remote locations.
c) Providing a methodology for the real-time measurement and reporting of personnel training readiness.
d) Increasing the efficiency and effectiveness of scheduling training for the drilling Reservist.
e) Providing more timely and accurate information for mobilization reporting.
f) Improving the reliability of training information at all command levels.
g) Reducing the administrative and clerical workload of operating units.
h) "Capturing" input data at the source, thereby eliminating intermediate error-inducing steps.
i) Providing limited stand-alone local processing capabilities for the Naval Reserve Center.
j) Providing an integrated communications capability enabling the localized units to exchange/update data

designed to be capable of creating a trainee's training record or acquiring an existing record from another ATSS site when the trainee initially enters the Naval Reserve, maintaining an individual's training record, and monitoring the training history/requirements throughout the individual's entire service in the Naval Reserve. RTSS(Air) must also provide communications capabilities to support activities nationwide. This includes aviation activities at any of eight Reserve NASs, seven NARs, and active NASs. It is supposed to have the flexibility to support both remote and local users via several methods: dial-up, dedicated communications and batch updating through exchange of information with a remote minicomputer, word processor, or intelligent terminal. The long term objectives of RTSS(Air), [Ref. 2: p. 1], are:

a) An increase in the quantity and quality of Selected Reserve (SELRES) mobilization billet assignments at all command levels.
b) Integration of personnel and training record data under a single system accessible from remote locations.
c) Reduction of time and training resource requirements through individual SELRES diagnostic testing; training scheduling, tracking, and reporting; individualized instruction; and maintenance and administration of syllabi and courseware.
d) Reduction of time required for, and increase in accuracy of, development of data for mobilization.
e) Monitoring personnel readiness status.
f) Improvement in the effectiveness and efficiency of tracking trainee progress.
g) Improvement in the reliability of training information at all command levels.

ongoing USN training initiatives and provide compatibility with a standard active duty computer system. In addition, some cost avoidance would be realized from the sharing of resources when active USN and Reserve aviation activities were co-located at existing ATSS sites. Further savings would be realized by adopting an already existing system and avoiding the time-consuming and expensive systems development process.

For funding and control purposes the system was renamed the Reserve Training Support System (RTSS). It consisted of three major component systems: RTSS(TM) for training management, RTSS(Surface) for surface/ashore, and the RTSS(Air) for aviation. CNAVRESINST 5230, [Ref. 2: p. 1], established policy for the development as follows:

> The Reserve Training Support System (RTSS) is an automated training management support system. The purpose of the system is to provide training management support to field Naval Reserve training administrators and to increase the quality of training readiness information reporting at all Naval Reserve Command levels. A dual system approach will provide for a field training system in support of Naval Reserve field administrators and a Regional Training Management System in support of staff functions.

1. RTSS (Air)

The component of RTSS known as RTSS(Air) is designed to support personnel training and training management at aviation activities--NASs, Naval Air Reserves (NARs), and Reserve Forces Squadrons (RESFORONS). This includes active duty as well as drilling Reserve personnel. The system was designed to support multiple training needs--formal schools programs, On-the-Job-Training, initial training for Ready Mariner (RM) personnel, refresher training for veterans gained to the Naval Reserve after leaving active military service, and combat-ready training for RESFORONs. It is be more cost effective than a brand new system could initially provide. This is the first of many instances where outside agencies recommended that the ATSS/RTSS family be expanded to include more than just its limited training application. These agencies have recognized that training is only one step towards achieving the organization's goal and that many other functions should also be automated. CNO concurred with the findings of the study, [Ref. 1: p.19], with a lesser degree of urgency than implied in the study:

> ATSS, as it exists today, meets the intent and scope of an automated information system and should be managed, developed, acquired, and funded under ADP rules and regulations. However, one point must be emphasized. Regardless of ATSS status, it remains to be determined, through detailed feasibility studies and economic analyses, to what extent other systems can be accommodated and savings achieved without degrading the primary function of providing highly trained and qualified fleet replacement personnel for the aviation community.
>
> CNO will take the necessary action to remove the ATSS exemption from ADPE regulations consistent with an orderly transition scheme so as not to disrupt ongoing ATSS efforts.

ATSS is now finally under the ADP umbrella, and is using OPN dollars for funding. The transition of the ATSS Program from the Naval Weapons Center to the Navy Management Systems Support Office (NAVMASSO) should start in FY 85, with NAVMASSO picking up responsibilities for contracting and software development.

B. RTSS

The Chief of Naval Reserve became interested in ATSS as a viable training and administration tool for its activities as early as 1977. ATSS was chosen as the most cost-effective method to achieve its required personnel training and training management support. Justification for its selection was that it would align the Naval Reserve with route because all of the ATSS (and RTSS) software is tied to RSTS/E.

During Fiscal Year 1978, the Office of the Comptroller of the U. S. Navy (NAVCOMPT) reviewed the ATSS exemption status and ruled that ATSS was not a training device as defined by Department of Defense/Department of the Navy budget policy. NAVCOMPT and CNO reclassified ATSS as a computer-assisted training system within the generic category of equipment configured solely for training applications. NAVCOMPT authorized continued appropriation of ATSS hardware and software via Naval Aviation Procurement funds (APN), predicated on ATSS being used solely for aviation training applications with no other expansion capabilities in the future.

A Naval Audit Service study completed in 1980, [Ref. 1], found ATSS development and acquisition generally satisfactory. Two areas were mentioned where improvement could be made:

1. Opportunities exist for reducing costs by beneficially employing ATSS hardware and software for requirements of other related information systems and by manning operational sites with government vice contract employees.
2. Improvements can be made in fund administration and in the integrated logistic support areas of planning and configuration control.

The study determined that CNO's refusal to expand ATSS was based on the perceived need to maintain ATSS exemption status under the ADPE acquisition regulations. It found also that this exemption is no longer needed, because the final 10 ATSS software systems to be purchased were ordered under a FY 1980 contract. The general conclusions stated that the benefits of expanding the inherent capabilities of the ATSS software to allow for additional field uses would

By 1975, the system was designated ATSS for management control, and the Weapons Center still holds development control over the ATSS software.

A hardware upgrade to the PDP-11/70 followed, allowing expansion of the 101 data items to 509, and signalling the advent of Version Two of the software. The original file structure became inadequate. Version Three software design began in 1976 at five operational sites. The intent was to eliminate some redundancies and clearly differentiate separate functional areas into subsystems. A phase-in approach was used for Version Three design and development. This substantially increased the capabilities of the system, but failed to achieve the total integration required. It lacked the flexibility needed as the uses of the system gradually expanded. Thirteen subsystems have since evolved from these beginning efforts; Version Four of the software was designed to solidify and integrate these subsystems into a functional and expandable system incorporating a Data Elements Dictionary, easier program maintenance procedures, and a more accurate software configuration control. At this writing, version four software was still under development.

Large ATSS sites were upgraded to the VAX-11/780 CPU, a 32-bit machine capable of a 4-gigabyte program address space. The same RSTS/E operating system was used instead of the more versatile Virtual Memory System (VMS) operating system designed specifically for the VAX-11/780. RSTS/E burdened the normally efficient VAX computer and these ATSS sites did not realize a significant increase in computing performance over the old PDP 11/70's. All attempts to improve the RSTS/E software so it would not diminish the VAX's efficiency failed and NAVAIR finally returned the VAX computers for more PDP 11/70's. NAVAIR intends to keep the RSTS/E operating system and attempt to improve it until it is efficient enough for the VAX. They have decided on this

(VTS). The Naval Air community currently uses ATSS; the submarine community uses the VTS identification; the Reserve community named their adaptation RTSS; all systems have the same basic configuration.

The Digital Equipment Corporation (DEC) PDP-11/40 computer was selected as the initial mini-computer for CMP. Ordered in late 1971, it was procured through a competitive contract as a turnkey training device, i.e., the contractor was responsible for all installation, and only had to "turn the key" on before handing it over to the Navy. The system was exempted from the complex and lengthy Automatic Data Processing Equipment (ADPE) approval requirements by the Chief of Naval Material because it was designated solely a training device. The test and evaluation was done for VA-122 and two other operational A-7 squadrons. The computer was installed in mid-1972. The LTV software developed in Dallas was used, with software development continuing on-site. Data and operating files were generated for FRAMP and the operational squadrons. Evaluation of the prototype was successfully completed in 1973. Administrative personnel from other Naval Air activities visited the CMP at NAS Lemoore and were given a demonstration of the system.

CMP's name was changed to PMS during this time. In December 1973 the Naval Weapons Center at China Lake took control of development and implementation of the project. Its job included:

1. Anticipating the training needs of the various Fleet activities.
2. Observing and evaluating training methods and procedures both before and after installation.
3. Determining the expansion required to maintain a satisfactory level of training readiness in all areas supported by the system.

II. BACKGROUND/EVOLUTION OF RTSS

A. HISTORY

The Reserve Training Support System (RTSS) is essentially the same as the active duty system known as the Aviation Training Support System (ATSS). The ATSS concept was devised early in 1971 by the Ling Temco Vought (LTV) Aerospace Corporation to aid training coordination and scheduling at one of the Navy's A-7 aircraft Fleet Readiness Squadrons (FRS), Attack Squadron One Hundred Twenty-Two (VA-122) at the Naval Air Station at Lemoore, California. The initial version was tested at VA-122's Fleet Readiness Aviation Maintenance Personnel (FRAMP) Department. The primary job of a FRAMP Department at an FRS was to ensure that enlisted aviation maintenance personnel were provided to fleet operational squadrons adequately trained to perform their jobs. The original goal of the system was to provide a training support system oriented toward enlisted aircraft maintenance training, and was developed out of a need for relief in assigning courses and classes and tracking students. ATSS also alleviated the manual generation of reports, forms, and other paperwork associated with a major training syllabus. It was initially a small software package consisting of 20 programs designed to manage 101 data items concerning the receipt, transfer, and course/class assignment of enrolled and future FRAMP students.

The early success of ATSS led to adaptations by other communities. The system has had a number of titles associated with it since 1971: originally known as Computer-Managed Personnel (CMP) System, then Personnel Management System (PMS), and then Versatile Training System

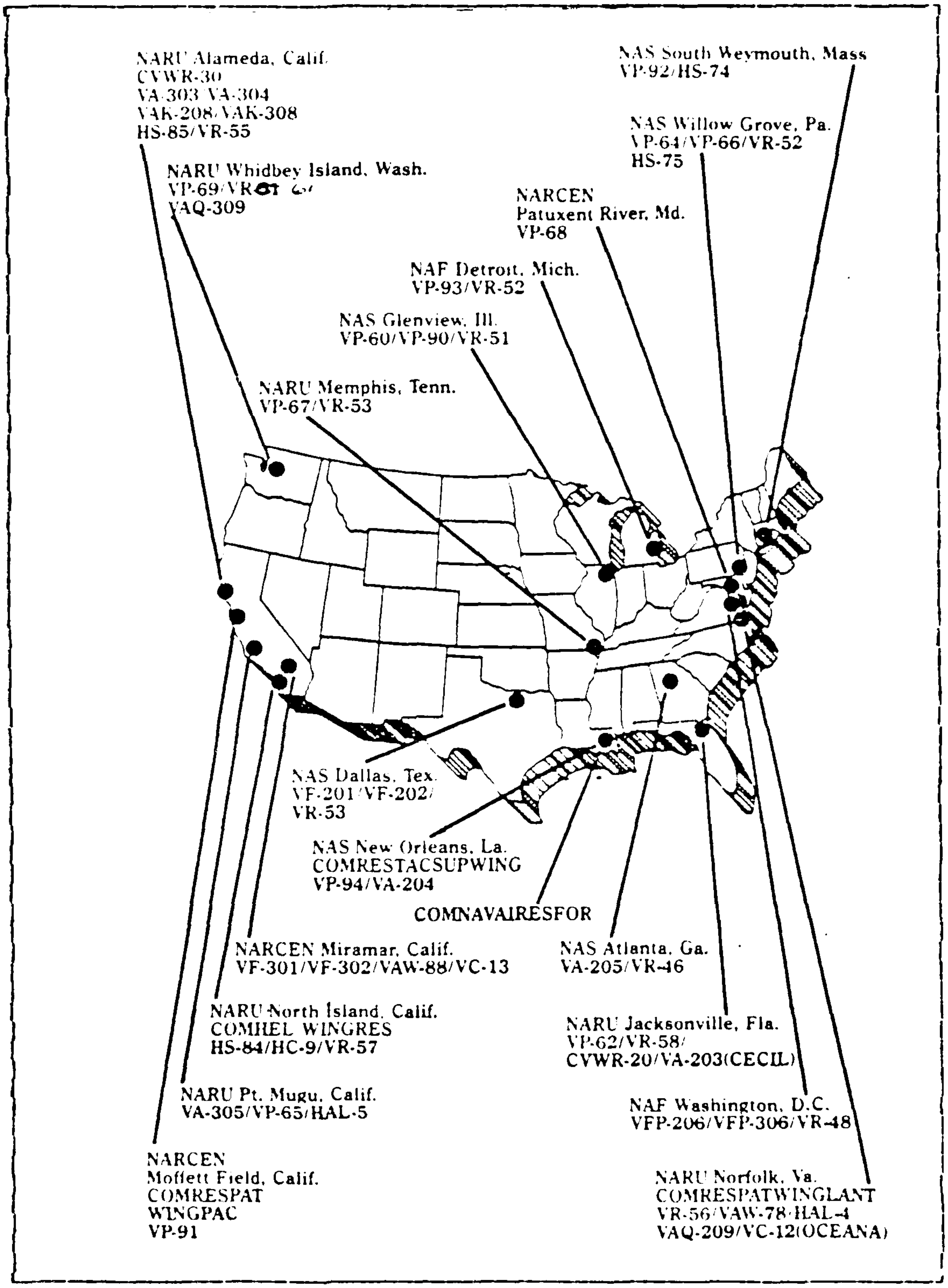


Figure 2.1 RTSS Geographical Distribution

quantifiable dollar costs and non-quantifiable readiness costs of maintaining dysfunctional or manual systems. Second, even when the software is received, it requires some massaging to fit the Reserves own unique needs. The Fiscal Year 1984 Reserve budget allowed for little else than maintenance of software; the Fiscal Year 1985 budget was cut in half, but will allow for some improvement.

A new contractor, Martin-Marietta Corporation, is in place as of the beginning of Fiscal Year 1985. Software development and hardware installation efforts for ATSS/RTSS were abridged appreciably during the contract negotiation and letting process from July 1983 - September 1984. Periodic continuance renewals of the previous contract with the Syscon Corporation served only to keep a maintenance man on at existing sites, but delayed needed consulting and development services.

Other difficulties include the delays that the Naval Weapons Center is experiencing in getting the new version of RSTS/E implemented; it seems that Martin-Marietta does not have the appropriate maintenance contract with Digital Equipment Corporation for operating system updates. Also, as a result of drawn-out contract negotiations, there are three month delays before software changes can be issued through the new contractor. Additionally, the ATSS software still cannot track personnel at changing sites.[3]

Emphasis for implementation of the RTSS program through 1984 by CNAVRES has been in procurement and installation of hardware and peripherals at the various sites. This emphasis is now shifting toward bringing software on-line and beginning user training. Implementation has been slow

---

[3]We found an ironic anecdote in our ATSS background research: according to the minutes of the September 84 ATSS Configuration Advisory Board Meeting, several users volunteered their services to NAVAIR as model managers in evaluating commercial off-the-shelf software. Somebody beat us to the punch, at least in spirit. See [Ref. 7: p. 2].

due partly to the newness of ADP to the Reserves, but more so to the annual funding constraints and contractor negotiation problems of ATSS. At this writing, a development hiatus of several months caused by a major budget slashing of contractor support has finally been cleared up; CNAVRES is slowly forging ahead.

Networking capabilities from field sites to New Orleans are still in the planning stages. While it was initially hoped that a DEC product known as DECnet would serve as an adequate off-the-shelf candidate for the interface software, development and implementation of the Defense Data Network(DDN) have precluded a stand-alone network. OPNAVINST 2070.4 of 7 March 1984 has mandated that ADP systems requiring long haul data communications include provisions for using the DDN as their primary data communications medium. Attempts by CNAVRES Code 46(ADP Plans) in mid-1984 to waiver RTSS from this Defense Communications Agency requirement were unsuccessful. Code 46 has since been investigating the necessary means by which to comply, including soliciting bids for protocol conversions for the RSTS/E, RTSS, and DDN interface.

In addition to the above software and hardware problems, we uncovered many deficiencies in computer support. Specifically, no <u>clearly</u> written users manual was or is now in existence for RTSS (Air). No funding for revising, updating, editing, or rewriting the old, existing manuals is available at this time. The instructions, manuals, and minutes of meetings that we read in our research stressed that RTSS(Air) is a training system, placed in the field for the benefit of the user; usage is limited to training oriented subjects, and any use which cannot be directly related to the training of aviation personnel cannot be supported on this system. The assertions were unclear given that ATSS is now under Navy ADP guidelines.

Other specific problems with the RTSS system(s), as seen by the users and the experts, will be addressed with the larger issues of AIS management in the next chapter.

III. AUTOMATED INFORMATION SYSTEMS

A. INTRODUCTION: THE VALUE OF INFORMATION

The Naval Reserve is the recognized backbone of a strong mobilized nation when all-out war occurs. It is now wrestling with the nontrivial questions of meeting distributed real-time personnel and mobilization information requests. In its information processing arena, however, it faces many of the endemic organizational problems of a still young computer age: lack of an implemented top down design strategy, and possession of antiquated hardware and software. The software for its RTSS(Air) project, derived from the initial Aviation Training Support System, was developed in an era when computers were used only for data collection, processing, and storage. Used only as a computing device, it had limited strategic management value; at that time there was no intent for a correlation between strategic objectives and computer system utilization. Its value as a decision making tool in helping to evaluate training and mobilization readiness was and is extremely limited.

Today information is a powerful tool that influences the health and well-being of all organizations, government or business. Members of a 1983 blue ribbon panel appointed to study the Navy's utilization of automatic data processing equipment unequivocally stated: "Virtually every action by a commander, manager, or administrator in the Navy, as in any large organization, involves the acquisition and understanding of information: information about the organization, about its status, about its resources, about its environment. His actions usually result in the creation and promulgation of policies and directives." [Ref. 8: p. 2]

Information is a key resource in both the development and execution of organizational strategies. In this context, it is essential to develop an information system based on the strategic objectives and direction of the organization [Ref. 9: p. 39]. While it is the task of the lower levels of the organization to collect data and organize it in a meaningful form, it is the role of the strategic planners to define the scope of the desired information.

The use of computerized information systems has greatly aided the collection, compilation , and presentation of data and information. The collection process is no longer a difficult or costly task. As a result, there is a proliferation of available data that must be converted into useful information and analyzed. Managers at different levels of an organization require different information. Managers must be able to extract information relevant to their realm of decision making. Information management must emerge as a critical discipline for an organization's strategic decision-making process.

Information management must be an aggressive program that presents the correct amount of substantive information. In dealing with the volume of information, more information will enhance decisions up to a certain point; beyond that, a law of diminishing returns sets in. To gather the correct amount of information, managers must understand how to effectively use current and predicted future technology.

For the remainder of the chapter we will address this technology issue and some specific problems, past and present, for the Navy and Naval Reserve, and present the latest Information System Strategic Plans for the Navy and the Reserves. We will present a mixture of design (redesign) strategies to cope with design problems, and recommend a state of the art problem solving approach that we found viable, functional, and relatively immediate.

B. PROBLEMS

1. Naval Reserve

The onslaught of new computer technology in the last 20 years remains clearly in the headlines. Unfortunately, the depth to which this advancing technology can permeate to the field level in agencies as large and complex as DOD or the Reserves is constrained by educational, fiscal and bureaucratic methodologies that stifle initiative and reward stagnation and the status quo. A January 1984 study, Analysis of Naval Reserve Force Information System Management Requirements, was both blunt and specific in its cited problem areas and recommendations. The list of problems could be a primer for a 'what not to do' Information System reference book.

First, they saw no single organization taking responsibility for the automated information systems in use by the Naval Reserve. Many of the systems are the result of "favors" or "hand-me-downs" from others, and in most respects not well-designed for Reserve use because they were never intended for this community. The lack of accountability resulting from AIS functional sponsorship of Reserve systems by other commands has been a major contributing factor to the current inadequate state of Reserve information systems. [Ref. 10: p. 2-3]

Second, the bureaucratic headache of Life Cycle Management resulting from the Brooks Act of 1965 and subsequent directives addresses only the issues of AIS acquisition and development plans. The need to view information as a resource in an overall Information Systems Management Plan is lacking.

Third, current Reserve AISs do not get the job done. They do not contain all the data needed for desired monitoring and control functions. They do not provide for easy information retrieval in desired formats. They currently contain duplications of data, with inconsistencies in definition processing and data entry which lead to confusion and/or inaccurate measures of effectiveness. The present systems were developed mostly during a period when available hardware and software were more expensive and had less capability. They were usually a response to a crisis management need and often developed with virtually no user interface. [Ref. 10: p. 3-1]

Fourth, staff proficiency in AIS planning and leadership was nonexistent. In the finding's words: "So if the distributed architecture...recommended...is to become a reality, it is through the 'hands on' involvement of line management on the staff and in field activities" [Ref. 10: p. 3-5]. Effective use of new hardware and software technologies guided by proper application of information resource management skills must be the approach for the leaders of the 80s and 90s.

Among several specific organization realignment suggestions, a fifth and major recommendation was for the development of a long range technical architecture for information systems [Ref. 10: p. 4-6]. It is to be based on a distributed model whose major components are:

a) Electronic workstations
b) Distributed computer system hosts
c) An advanced electronic data network

Perhaps the strongest and most important recommendation the study put forth was to adopt as a goal for long range planning purposes "the ability of all members of the Reserve claimancy to accomplish their assigned tasks with the assistance of automated information systems" [Ref. 10: p. 4-2].

To illustrate the concept of information as a resource in the Reserves, some examples of present day effectiveness are pertinent. In their preliminary draft of an Information Systems Plan for COMNAVRESFOR, the team noted deficiencies of various Automated Information Systems as they related to functional needs. Among them:

a) Fleet Programs

The total picture of I.S. use in Fleet Programs, but equally in Logistics/Shore Support Programs and Staff Programs, is one of solutions to information needs, and partial redundancies and dissatisfaction with the accuracy, timeliness, format, and ultimate usability of the information received. The dominant perception through the 31 and 32 divisions <Programs and Training Support> is expressed in the remark of one Program Manager that 'much of the time we seem to be working for the computer rather than the computer for us'. [**Ref. 10: Appendix 4-46**]

b) Logistics/Shore Support Programs

RTSS(TM), used for such purposes as obtaining billet listings to determine allowances and field inquiries about billets, as well as for addresses and address labels, is less than satisfactory as a provider of required data. Two program managers in fact try to minimize reliance on it due to unavailability of the sorts most useful in program management (e.g. alphabetical by type of unit), and the system's slowness. The hand sort which two program managers presently revert to takes approximately four hours. A program manager who does rely on RTSS(TM) counts on a one-day delay to receive requested output. Since that often is not soon enough for fielding senior's inquiries, one result is that great volumes of paper are kept from which to generate manual reports. Code 3124 relies for much of their data on a report completely outside the organization, a NAVSUP report which aggregates manning data on NAVSUP-supporting units, since it is more accurate and timely.

A further shortfall in RTSS(TM) at present is that some data fields are not or are infrequently updated (for example, education, address and telephone numbers). Fields that are never updated would better be removed from the system altogether, in some officers' estimation, in that wrong data often are more deleterious in their effects than no data at all. [**Ref. 10: Appendix 4-47,4-48**]

c) Staff Programs

Lack of accuracy and completeness are a continuing problem, with monthly reports frequently at variance with manually maintained figures by 20% or more....Short term emergency mobilization needs, such as Grenada and Lebanon, essentially must be managed with manual techniques due to lack of responsive terminal facilities and an accurate mobilization database. [Ref. 10: Appendix 4-49,4-50]

d) Flight Support

The RTSS(Air) software is behind the times in format and content - content too old for flight reports. The timeliness of these content data will be enhanced through modification to process.

Code 55 <Flight Support> has a need to transfer information from system to system in a rapid manner to facilitate management decisions. This need affects 55's managerial effectiveness.

An example of this sort of concern is the inability of 55 to gain access to requirements except NECs which helps but an individual's experience is the more important access so far as readiness is concerned. [Ref. 10: Appendix 4-52]

2. Navy

The Navy did not escape the scrutiny of close observation of its ADP sins. The blue ribbon panel funded by the National Research Council found that the entire Navy was operating with "computing equipment produced in the 1960's", and too little attention being paid to "policy development and strategic planning...<and> to the potential of management-level information systems" [Ref. 8: pp. 4,5]. The panel published a number of findings in a narrative formatted ten page chapter of the report. The following points are pertinent to the Reserves' implementation and expansion of a 14 year old borrowed system, and are quoted verbatim from the report:

a) Hardware

Many problems arise when obsolete equipment is allowed to remain in operation. First, the cost of operation and maintenance is much higher for an early-generation machine, compared to that for a current-generation system of equal capacity. Savings in energy, floor space, and maintenance can usually repay the purchase cost of a new central processor in less than two years. Reliability and repairability is far superior for current-generation equipment. Policies that encourage reusing old equipment do not properly recognize these potential savings.

b) Software

A more difficult concept is that software becomes obsolete just as surely as does equipment. Equipment replacement is usually viewed as a modernization project, necessary to improve reliability, maintainability, and operability. But modernizing equipment and failing to question whether software is also obsolete really addresses only a part of the obsolescence problem.

Early-generation software systems were built as outgrowths of manual record-keeping systems,...<and> were expected to produce reports, update sequential files...,and perhaps punch cards...The early computing equipment was not designed to permit remote access to files; and often even if it employed random-access storage devices, these were often not managed by a database software.

When early-generation software is rehosted on a modern computing system, important capabilities of the new equipment are left unexploited. Furthermore, the requirements for an applications system do not remain fixed; they change over time--either from external pressures to modify a function or incorporate a new one, or because a better way has been found to perform the function. When a system is modernized, users should seriously question whether the software systems are still relevant to their information needs, or whether new software should be designed to exploit the capabilities of the new equipment. Thus, by incorporating new software that permits on-line access to a current database instead of an older system of numerous batch-updated programs, the jobs of clerks and managers could be redesigned, perhaps greatly reducing errors and operational costs. The resulting software system may be more useful, more efficient in terms of labor and machine resources, and truly exploitive of the equipment on which it runs. [Ref. 8: pp. 10-11]

C. STRATEGIES

The Chief of Naval Operations has not turned a deaf ear to these findings. OP-01, as functional sponsor, has delegated overall management of the Manpower, Personnel, and Training Information Systems (MAPTIS) Program to OP-16.

Commander, Naval Reserve Force is one of the six echelon 2 commands encompassed in the MAPTIS functional community. The Reserves' involvement as an echelon 2 command in this regard is limited to those AISs specified in Memorandums of Agreement with OP-01; RTSS is one of those systems. [Ref. 4: p. iii]

The Program Objectives Memorandum for 1987 (POM 87) MAPTIS Program Strategic Plan is an extensive work that says all the right things for correcting the wrongs of AIS problems. It lays out ambitious and challenging guidelines by which it intends to progress for the Fiscal Years 87-91. The document is most encouraging in tone with regard to AIS planning:

> The functional community's dependence on the use of automated information support has reached a point where it has become critical to MPT mission accomplishment [Ref. 4: pp. iii].
>
> Paramount to insuring sound information system planning is a direct relationship to front-end mission planning [Ref. 4: pp. iv].
>
> The rapid pace of change in ADP technology must be planned for if we are to avoid obsolescence [Ref. 4: pp. 2-34].

The plan, which provides information support goals and objectives to the MPT commands, is the vehicle for communication and concurrence between MPT functional managers and MAPTIS information support managers. It has four major goals defined and established:

a) Achieve an Effective and Efficient Operational Posture

b) Improve the Management of Resources

c) Integrate Modern Information Technology into the MAPTIS Program

d) Improve the Quality of MPT Data and Information

An adjunct guideline to this document cites the DOD thrust to improve productivity and mission performance in meeting the above goals through the application of end user computing technology, i.e., microcomputers. This Department of the Navy (DON) document encourages MAPTIS program managers to "incorporate end user computing technologies, as much as possible, into the MAPTIS Program as a cost-effective method of distributing information processing support to functional users." [Ref. 11: p.3]

Getting back to the POM 87 plan, the Reserves are specifically singled out as one of the major challenges in meeting the DCNO's objective of improving fleet readiness by continuing to effectively and efficiently fulfill manpower requirements. Along with a trained active force and increased emphasis on contractors and civilian employees, the plan cites the intent to expand the Naval Reserve over the next five years so that it will be able to take over missions from the active force. It stated unequivocally:

> Automated information support is essential if Navy MPT managers are to succeed in these efforts. The ability to rapidly assess all elements of the total force is crucial to meeting the peacetime, mobilization, and wartime requirements levied on the Navy's MPT managers. This assessment depends on quality information, i.e., information that is accurate, timely, complete, and understandable. Meeting wartime information requirements involves planning, developing, and maintaining procedures and associated data processing programs and equipment that will support (1) the maintenance of the Navy's active duty, civilian, and Reserve forces, (2) the mobilization of the inactive members of the Navy, and (3) the smooth transition from peacetime to wartime management of the total force. [Ref. 4: pp. 1-1,1-2]

One of the strategic goals of the plan is for a central pay and personnel system called the Source Data System (SDS). With a pilot site in place and successful at a personnel activity at Lakehurst, New Jersey, future software enhancements are to extend the variety of functions to mobilization, order delivery and processing, and Reserve personnel accounting and drill reporting. Plans call for COMNAVRESFOR to develop a distributed personnel, pay, and training system by the second quarter of Fiscal Year 1986. Integration testing of initial Reserve support applications will begin in FY 86. Software used in this consolidation of Active and Reserve personnel data base systems is to be tested in FY 86. In January 1987, SDS and COMNAVRESFOR are to update the implementation strategy and plans for exportation to field sites. [Ref. 4: pp. 2-3 to 2-7]

Nonetheless, RTSS is not going to disappear soon. The Five Year Defense Plan indicates a requirement for funding through 1991. A MAPTIS budget increase of $924,000 is being requested for POM 87, all of it earmarked for the three RTSS systems. Contractor support, software conversion, and DEC/VAX-730 hardware are requested to support the systems and "migrate programs to a more current technology" [Ref. 4: p. 3-60,3-61]. Code 461 in New Orleans indicated that much of the software conversion was expected to be done in-house once the new equipment arrived.

One area missing from this long-range plan involves training; specifically, training for the Officer corps. In the past computer training has been aimed at the enlisted person who will be operating the computers. In fact, basic computing skills are now being taught at enlisted 'A' school. This is not sufficient to achieve proper utilization of information systems. It is the officer corps who should know how to convert the vast amount of data into usable information. They must receive a larger portion of information systems training in future years in order to make proper use of current technology and effectively manage the Naval Reserve in the rest of the 1980's and the 1990's.

D. CURRENT TECHNOLOGY

It is no wonder that the Naval Reserve has been slow to technological change. Only since the latter years of the enroll; the information is entered directly into the Activities table without considering the student data entity.

2. Data Duplication

The problem of data duplication is illustrated in Figure 4.3 . The table shown contains information about a reservist's Social Security Number and his Reserve Unit (UIC, Long Name, Address, City, State, Zip). It is immediately obvious that this design will waste a large amount of storage. The same information is stored about the Reserve Unit for every member of that unit. One copy of each reserve unit's address is sufficient. Data duplication occurs because information about two data entities (Reservist and Reserve Unit) is stored in the same table.

Reserve Information

SSN	RUIC	ADDRESS	CITY	ST	ZIP
111111111	29604	208 J ST	TULSA	OK	34567
222222222	29604	208 J ST	TULSA	OK	34567
333333333	10864	1 STONE RD	MIAMI	FL	29456
444444444	29604	208 J ST	TULSA	OK	34567

Figure 4.3 Data Duplication (uncorrected)

Eliminating data duplication is a relatively easy process. The method is very similar to dealing with modification anomalies. Two tables are created so that each data entity has one to store its information in. Figure 4.4

The deletion anomaly problem can be easily explained using Figure 4.2 as well. If the student with SID 100 drops out of school and that record containing his activity information is deleted, we lose two important facts. First, that student 100 was participating in skiing. This was the intent of the deletion and is an acceptable loss. We also lose the fact that to participate in skiing it costs $200. This an unacceptable loss resulting from poor database design.

Student

SID	Activity
100	Skiing
150	Swimming
175	Squash
200	Swimming

Activity

Activity	Fee
Skiing	200
Swimming	50
Squash	50

[Ref. 16: p. 287]

Figure 4.2 Modification Anomalies (corrected)

The design to eliminate the modification anomalies in Figure 4.1 is shown in Figure 4.2 . Notice that we have provided a means for storing information about students and activities independently by creating two separate tables. If student 100 is deleted, we still know that skiing costs $200. If we want to know how much student 200 has paid for his extra-curricular activities, two look-ups are required. First in the Student table and then in the Activities table. Additionally, if volleyball (fee = $45) is added to the activities list, we don't have to wait for a student to one data entity without first adding a fact about another. A deletion anomaly occurs when facts about one entity are lost when a fact about a different entity is deleted. These two anomalies are illustrated in Figure 4.1 and Figure 4.2

Student Activity Record

SID	ACTIVITY	FEE
100	Skiing	200
150	Swimming	50
175	Squash	50
200	Swimming	50

[Ref. 16: p. 287]

Figure 4.1 Modification Anomalies(uncorrected)

We have chosen a very common example to explain how these modification anomalies occur and how they are eliminated. In Figure 4.2 there is a table to collect information on students(by Student Identification Number - SID), the extra-curricular activities activities, and the fee for that activity. The insertion anomaly problem occurs if a new activity, such as volleyball (costing $45), is added to the list of extra-curricular activities. We want to add this to the database. However, this is not possible until the first student enrolls in volleyball and there is a valid SID to enter along with the activity and fee. This is unacceptable. The only way to add a fact about one data entity (an activity) is to first add a fact about another (a student). [Ref. 16: p. 287]

B. NORMALIZED FORMS

1. Modification Anomalies

The design of the records for the Naval Air Reserve prototype involved the transformation of approximately 600 data fields of the old sequential file system into a relational database. One important aspect of this design process is the elimination of modification anomalies and data inconsistencies. Modification anomalies occur when changing or modifying data yields unexpected results. Additionally, most incidences of data duplication are eliminated. Data normalization is the method of accomplishing this. Normal forms are the means by which data normalization is done.

To adequately explain how we arrived at the prototype design in Appendix A, we must first present the normal forms in database theory. We used five principal normal forms [Ref. 15: p. 120]. There are numerous articles on normalization which expand on the five normal forms. These include domain key/normal form (Fagin) and Boyce-Codd normal form [Ref. 16: p. 288]. To explain the normalized forms, we found that normal forms one through five as presented by Kent are the most logical and understandable.

Normalized forms provide guidelines to the database designer that eliminate errors and data inconsistencies in the final design. These guidelines are in the form of progressive design steps that methodically transform a collection of data fields into a database free of modification anomalies and data inconsistencies. Normalization is a stepwise refinement of the tables of a database.

The culprits in database design are modification anomalies and data inconsistencies. There are two major types of modification anomalies--insertion and deletion. Insertion anomalies occur when we cannot insert a fact about

IV. THE PROTOTYPE

A. INTRODUCTION

The purpose of this chapter is to present the Naval Air Reserve Prototype, the assumptions made, the issues dealt with, and the prototype design. There are numerous appendices associated with this chapter. Each appendix is a portion of the database design. This chapter discusses how we designed the database with the actual work appearing in the respective appendices.

A majority of this chapter will discuss how the ORACLE database management system was used to implement the many attributes of an effective automated information system discussed in chapter three. It is important to note that this prototype is only an example of how the Naval Air Reserve can use current technology to implement its many disjoint systems 'under one roof'. We will attempt to show that the needs of the Reserve can be best suited with a fourth generation, relational database system irrespective of the specific commercial product chosen. The importance of this prototype is in the structure of the data itself. This will not vary greatly between products. Throughout this chapter, we will present the shortfalls of ORACLE to facilitate a more informed decision for the actual system. Prior to presenting ORACLE, however, a discussion of database design and the use of normalized forms is necessary.

The computer industry has not developed one set of terms to define its vocabulary, so we will attempt to use the same term in referring to a concept or thing.

consistent and timely. The prototype of RTSS we have developed can serve as a working model for the Naval Reserve to develop an independent query and report oriented management information system using off-the-shelf current-generation technology.

- Develop a prototype with only basic functions: input,inquire,report,format.
- Let managers and analysts play with it to determine user needs.
- Add those functions common to many users and put in operation.

The point is that it is a selected database; users will want different functions or will quickly become bored with it. If it can be kept relatively simple, it can be changed or replaced with small investment. [Ref. 14: p. 9]

The Commerce Department's situation is certainly analogous to the state of affairs that the Reserves finds itself with RTSS and, indeed, virtually all of its Automated Information Systems. The history of mistrust of current AISs by those most in need of them is the most immediate indication that the users, not just the designers, need to be personally and dynamically involved in the process to make the system work. As to the use of microcomputers, the Navy's contract with Zenith Data Systems for the Z-100 and Z-150 (IBM-PC compatible) microcomputers could be an excellent vehicle for integrating modern information technology into the Reserves AIS strategy. It would also fit in with the CP-16 MAPTIS Program Strategy as cited above. In fact, the relational database product we use in our prototype is advertised to be completely downloadable to an IBM-PC/XT, which is essentially a Z-150 with a 10 megabyte hard disk.

On the other hand, it might be necessary to continue expanding a prototype RTSS system to the point where it is no longer simple, or perhaps there might be a desire to segment the database into an ad hoc part and a static part. In fact, we use more than 80 data elements in our prototype, and discuss more than the basic functions mentioned.

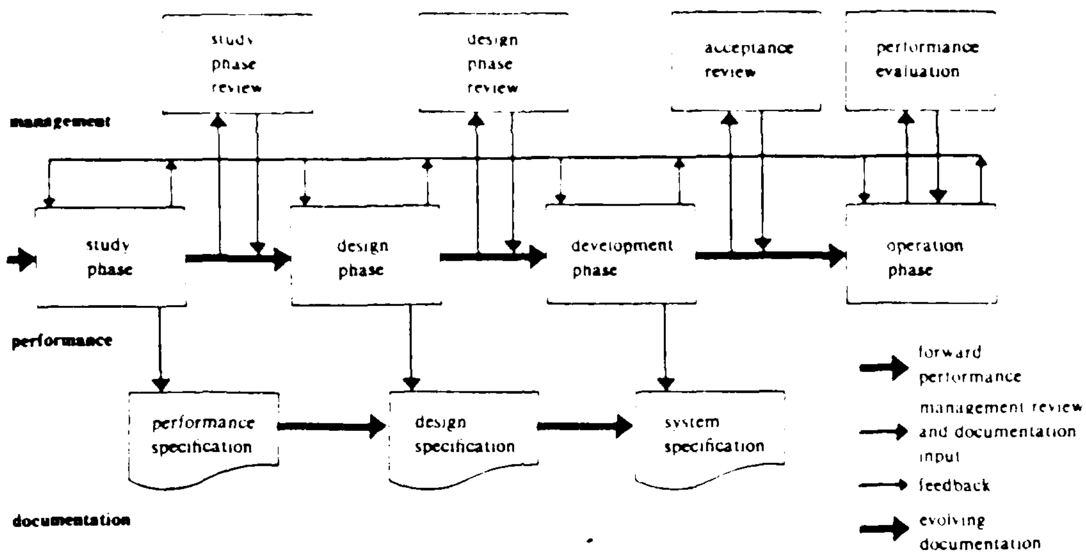
Our point is that the fourth generation software of today, like that which we used in our prototype, is a <u>powerful</u> tool that has the capabilities to be molded into user oriented results. Available information is more

Experimenting with fourth generation languages as a productivity tool is now going on under the MAPTIS strategy at NMPC-4 [Ref. 4: p.1-16], and the concept of prototyping is receiving more attention in the Federal environment. Consider a December 1984 article by the Deputy Commissioner of the Financial Management Service, U.S. Treasury Dept, Mr. Marcus W. Page. In it he commented on the Dept of Commerce's approach to bringing comparable resource data together from several bureau-level organizations. Initially their focus was on rebuilding the various different systems to produce comparable data. They moved from there towards one common system in each function being shared in the department. Neither approach solved their data incomparability problem. They then transitioned to what they termed a data "bridge", in which a selected number of data elements from several systems were combined into a common system. Developed by a contractor in a relatively short period of time, it had a relatively low cost compared to a major rebuilding effort. While not without its problems, the system at Commerce has been considered a success. The key to its success was in its simple consolidated data. For an agency building a selective data base, Mr. Page offered some sound advice:

> The real value of the selected data file method that Commerce used is that it fills a gap relatively quickly and provides a common body of knowledge that managers and analysts can manipulate by downloading sections of the file to microcomputers using <spreadsheet software> and report generators.
>
> Virtually every new agency organized out of predecessor agencies has gone through the same process of trying to bring together a consistent data file of selected information from the various bits and pieces. The difference today is that the tools are getting better to support data base creation.
>
> Perhaps the most reasonable and certainly most economical way for an agency to approach this situation would be the following four steps.
>
> - Define specifically a limited number of data elements using...approximately 80 as a start.

One relatively new method of developing requirements analysis is by prototyping the system using fourth generation user friendly query languages. Prototyping, as in aircraft development, has come to mean an example which is never actually used in service, only as an experimental model to display or sell operating characteristics. A software prototype, however, is a pilot working model of the real system which does not have to be discarded, but can be built on or changed at will. In contrast to the traditional software development approach which goes through lengthy stages of requirements analysis, design, development and implementation before a product is seen, the output of the prototyping process is generally a mini-version of a desired end product built with the close involvement of the users. On a limited scale, the user can see what the system can do in roughly an order of time less than the traditional process. This builder/user team has little difficulty understanding their own output; they can simply modify those items they are not happy with. When the team is satisfied with the performance of the prototype, the development process continues and expands, but the system also can go into use. As familiarity with the system grows, the users themselves become the systems developers.

The Naval Reserve can benefit from prototyping in several ways. The specifications that result are more complete because the Naval Reserve can better evaluate the system and tailor it to their needs. By interfacing with several levels of Naval Reserve management during the prototype design, the developer can deliver a result that is more responsive to the whole organization. The strategic planners can easily obtain the summary information they need while the operational and supervisory personnel will experience more efficient data input, maintenance, and output operations.



[Ref. 12: p. 26]

Figure 3.1 Traditional Development Process

analysis era and fell prey to these two problems (poor documentation and not meeting users' needs).

Requirements analysis can be accomplished in numerous ways. The traditional approach has been to identify the functions to be accomplished, decompose these functions into elementary units and provide a written document to the user explaining what the system will accomplish. We found no real requirements analysis documentation in our look at ATSS/RTSS, and considering when the development occurred, none is expected to exist.

success. The time allocated should be a function only of what a reasonable estimate of the development time is and not linked to any internal or external politics of the organization. Likewise, the money allocated should be sufficient so that the project team isn't handicapped. This is the ideal situation and is not likely to occur with great frequency.

A situation which is more likely to occur is where the development team finds that most variables and resources have been dictated by the organization. These decisions to commit resources are usually driven by factors not relating to the software project. This usually involves the organization's management mandating the project completion date or a specific strategy that leaves but a few options open to the developers. The development team must strive to produce the same result despite factors that threaten to detract from the overall quality.

Regardless of the environment in which the software development takes place, it is the analysis and definition of the system's requirements that can create the fundamental difficulty for the software development team. Requirements analysis is the cornerstone from which software will be built. It is the input to the design team's effort which is then delivered to the programming team for coding (see Figure 3.1). The effects of a poor or non-existent requirements analysis are far-reaching and devastating.

The requirements analysis methodology evolved in the early 1970's and was a substantial improvement in the previously unstructured development process. Software products from that era were typically without documentation and very difficult to maintain. In many cases, the product did not satisfy the needs of the user because care was not taken to ensure that user and developer were in agreement. The software for RTSS was developed prior to the requirements

To extract the optimal performance from a state of the art computer, new software must be used. The 1960's software would still be inefficient on the new computer and most of the new hardware benefits would be lost. This was most recently illustrated by the failure of NAVAIR to effectively interface the new VAX computer with the old RSTS/E operating system for ATSS. The same argument applies for current technology software installed on the old computers. Many of the benefits of the new software would be lost. There must be parallelism in the RTSS development plan. RTSS development must not follow the path of ATSS if the VAX upgrade is implemented in 1987.

E. SYSTEMS DEVELOPMENT

Updating or rejuvenating an obsolete computer system, whether large or small, is a major undertaking that must be managed by personnel trained in the systems analysis disciplines. With the advent of the latest generation of computer hardware that offers greater capacity and speed at a cheaper price, the selection of hardware is less critical to the overall success of the system development process. Hardware is still important. The size and complexity of the operations being automated must be correctly gauged to ensure that appropriate hardware is acquired. But since there is usually less risk in selecting the correct hardware components, the deterministic aspect of the system's development will be its software.

The software development personnel, their methodology, and the technology they use are critical elements in the eventual success or failure of the system. The ideal situation that the development team would like to encounter is where the time and money allocated for development is sufficient and that management is committed to the project's the policies and procedures of the organization and there is considerable risk involved with this change. Second, the large initial cash outlays involved with developing and implementing a new computer system can cause managers to side-step the issue. Finally, there is a good deal of risk involved with using a new technology that has not stood the test of time. The traditional approach to maintaining computing systems has been to minimize cost; spending as little as possible maintaining existing systems leaves many resources available for other purposes [Ref. 13: p. 3.]. The current systems will be replaced only after they have become unmaintainable and too costly.

Correctly applied, current technology allows Naval commanders to approach their designated missions with information that:

a) Has been collected and recorded simply
b) Has improved accuracy
c) Has been speedily reported, collected and distributed
d) Leads to summaries that are timely and to the point
e) Has enabled both manpower commitments and costs to be reduced [Ref. 8: p. 3]

Rapid progress in information systems technology has provided the potential for major advances in their effectiveness. But these advances are neither automatic nor easily attained. A key ingredient for the success of current information systems is a comprehensive plan. This brief analysis merely frames the problems that face the Naval Reserve and RTSS. The problems we have addressed have evolved because of a lack of strategic planning that includes the use of new technology in planning for computer systems.

The new technology that is chosen by the Naval Reserve must have hardware and software compatibility. There must be parallel progress in both the hardware and software areas.

Carter administration has the strategic value of a modern Reserve force been translated into meaningful funding. The Reserves have traditionally piggy-backed the active Navy's procedures and systems. Computing policies and equipments have followed suit, including the tendency to be several generations behind.

Our research found an eager but skeletal ADP staff at Code 46 in New Orleans. Recognizing their need for assistance in this volatile field, they immediately acted upon the NAVRES ADP Study recommendation to implement an Information Systems Support Unit (ISSU) composed of top-rated SELRES ADP professionals, only to have the program recently cancelled in the political and fiscal fight of mobilization vs. non-mobilization billets. The staff made a concerted effort in late 1984 to canvass the SELRES field for a selective group of qualified and experienced professionals that would meet regularly in support of CNAVRES Information System plans and goals. No sooner did they have the unit selected (September) when authorization was lifted at budget control levels (November) on the grounds that the unit would have been a non-mobilization unit.

Whatever the reason, this unit must be established. To move towards implementation of an AIS without the guidance of these proven ADP professionals would be a grievous mistake. The ISSU must provide the plan for managing the Reserve's resources through automation. The <u>only</u> way to adequately prepare for mobilization in the <u>future</u> is to manage <u>today's</u> resources with an automated information system.

There are some natural inclinations in management's decision whether or not to upgrade technologies. When a computer system has been in operation for a long period, management is reluctant to move to a new technology for several reasons. First, the system has become imbedded in shows the two tables that are necessary to eliminate data duplication. This design has the RUIC field in both tables to act as a cross-reference. Eliminating all but one copy of the Reserve Unit's address significantly improves the performance of the database.

Reservist Information

SSN	RUIC
111111111	29604
222222222	29604
333333333	10864
444444444	29604

Reserve Unit Information

RUIC	Address	City	ST	ZIP
29603	208 J ST	TULSA	OK	34567
10864	1 STONE RD	MIAMI	FL	29456

Figure 4.4 I ta Duplication (corrected)

3. First Normal Form

We have identified the major inherent problems in database design. There are others that occur infrequently and will be discussed as they pertain to the normal forms. Each of the normal forms will now be explained and an example from our prototype design will be used to illustrate these concepts, where possible.

First normal form is the easiest to satisfy because it is the broadest. A relation is in first normal form if all occurrences of that record type contain the same number of data fields and none of these fields contain the same type of information. These are called repeating groups. The data elements dictionary for RTSS had numerous repeating groups. This is not acceptable in relational theory. For example, the data field Secondary Navy Enlisted Classification (SNEC) was allowed to repeat in one record for up to four occurrences. We eliminated this first normal form violation by creating the table shown in Figure 4.5 Note that the number of data fields is only two and that all records are the same length. Repeating groups are eliminated by making a separate record for each SNEC. The procedure shown in Figure 4.5 was used in all cases where we found first normal form violations in the RTSS data elements dictionary.

4. Second and Third Normal Form

Second and third normal forms are usually discussed together. These deal with the relationship between the key and nonkey data fields. A key is one or more data fields that uniquely identifies a record. A nonkey data field "must provide a fact about the key, the whole key, and nothing but the key" [Ref. 15: p. 120].

Second normal form is violated when a nonkey data field is a fact about only part of a composite key. Figure 4.6 shows a table that keeps track of the reserve unit and reserve billet sequence code (RBSC) relationship. The address of the Reserve Unit is also stored. Data duplication is an obvious problem with this design. Second (and third) normal form eliminates this problem. In this particular case, the reserve unit's address is a fact about only Reserve Unit and not about RBSC. Figure 4.7 shows the

SNEC Information*

SSN	SNEC	SNEC	SNEC	SNEC
111111111	0621	2813		
222222222	2096			
333333333	1104	4211	2107	

* this design violates first normal form

SNEC Information#

SSN	SNEC
111111111	0621
111111111	2813
222222222	2096
333333333	1104
333333333	4211
333333333	2107

this design is in first normal form

Figure 4.5 **First Normal Form**

design in correct second normal form. All of the nonkey data fields now describe the whole key and not just a part of it.

Third normal form is violated when a nonkey data field is a fact about another nonkey data field. This is very similar to second normal form and is resolved in the same manner as shown in Figure 4.7 When a relation is in third normal form, every data field is either part of the key or provides a single valued fact about exactly the whole key [Ref. 15: p. 121].

Reserve Unit-RBSC

RUIC	RBSC	ADDRESS	CITY	ST	ZIP
23456	12345	206 J ST	TULSA	OK	45678
23456	12350	206 J ST	TULSA	OK	45678

Figure 4.6 Second Normal Form (uncorrected)

RUIC	RBSC
23456	12345
23456	12350

RUIC	ADDRESS	CITY	ST	ZIP
23456	206 J ST	TULSA	OK	45678

Figure 4.7 Second Normal Form (corrected)

5. Fourth and Fifth Normal Forms

Fourth and fifth normal forms have evolved to solve those problems involving multivalued facts. Multivalued facts correspond to many-to-many and many-to-one relationships. An example of these two relationships is shown in

Figure 4.8 . Part A shows that a student may be enrolled in more than one class and each class may have more than one student. This a many-to-many relationship. Part B shows that a father may have more than one child but that a child may only have one father(natural). This is a many-to-one relationship.

There were not many incidences of multivalued facts in the Naval Air Reserve prototype design so we will not continue with a discussion of fourth and fifth normal forms. In designing our prototype, we accounted for the few multi-valued facts in the proper manner.

6. Tradeoffs

The result of a fully normalized design is a database free of modification anomalies and data inconsistencies. One major advantage of full normalization is minimizing maintenance problems involved with a continuously updated database. Fewer, simpler rules exist for personnel who update information in the database when design-related errors have been eliminated. The computer operators can concentrate their efforts on entering correct data into the database.

There is a tradeoff involved with this 'superior' design. A performance penalty may be paid for a fully normalized design. Retrieval can be expensive, since data which may have been retrieveable from one record in an unnormalized design may have to be retrieved from several records in a normalized form. The advantages outweigh this penalty because most records are smaller and can be accessed individually without concern for the physical structure of the database. [Ref. 15: p. 120]

The proper balance between a fully normalized effective design, and a less normalized, but more efficient design, is difficult to achieve. The correct balance for

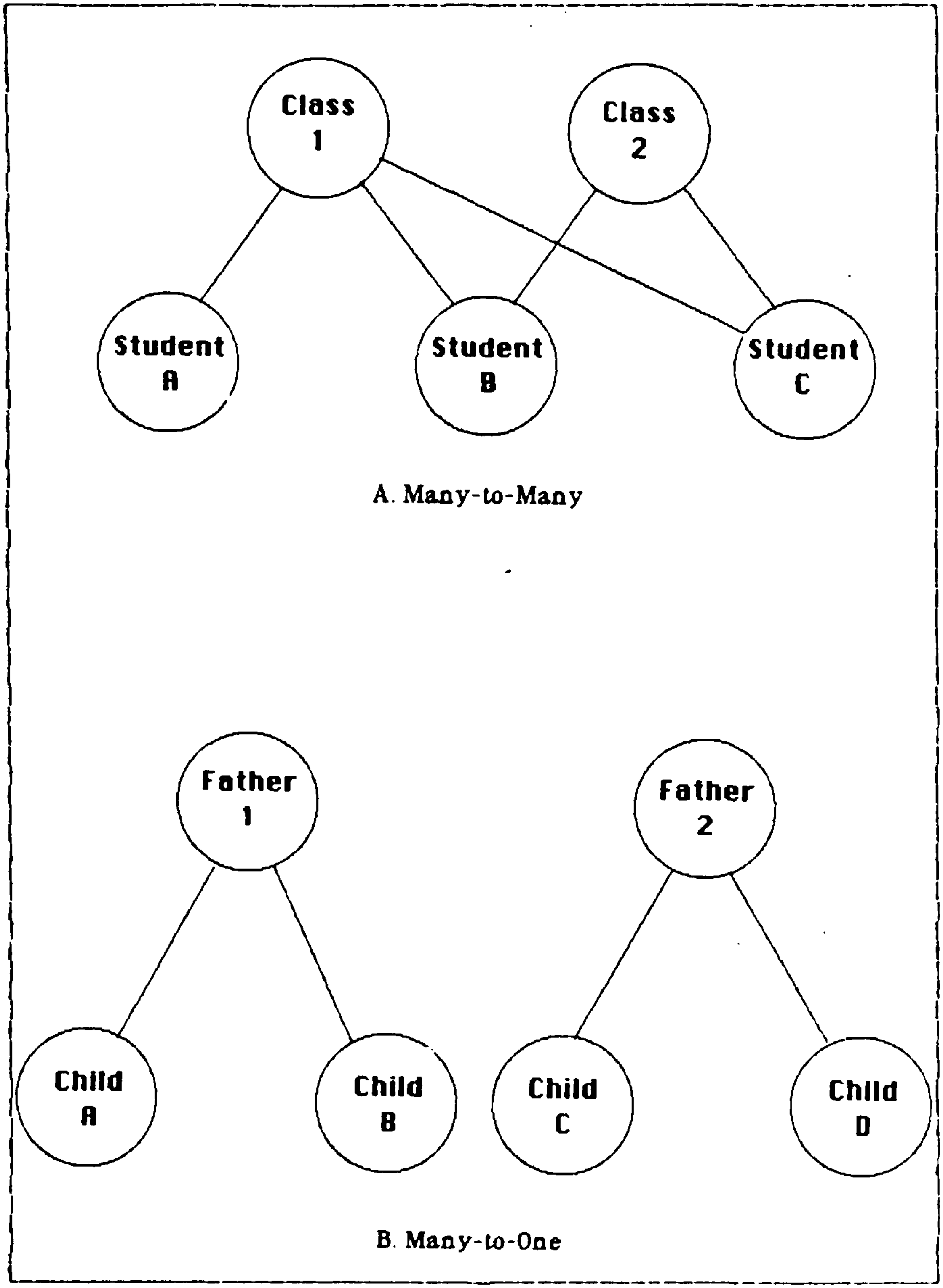


Figure 4.8 Multivalued Relationships

one organization may be different for another. This is a function of the way the organization uses the database. If retrievals are predominant, a design that contains some well-placed data duplication would be a more appropriate database design. In the case of the Naval Air Reserve prototype, a small amount of data duplication has been allowed. This will improve efficiency of the design without permitting any modification anomalies. A later section on improving efficiency using ORACLE features will explain how greater efficiency was achieved.

Appendix A, the data elements dictionary, shows the physical structure of our design. This design contains all of the data entry minimum requirements discussed in COMNAVAIRESFORINST 1510.1 (of 29 August 1984)[4] with the following exceptions:

a) BSN/CURRENT
b) BSN/DATE ASSIGNED
c) BSN/PREVIOUS
d) BSN/DISTRIBUTED
e) BSN/TRAINED
f) NEC/DISTRIBUTED
g) NEC/TRAINED

This design is the first iteration of the prototype design and will be tested by the end users. Recommendations for changes will be incorporated into later versions of the prototype.

Throughout the rest of this chapter, we will use examples from the prototype to illustrate a concept. You should refer to Appendix A frequently so that the figures are meaningful.

---

[4]COMNAVAIRESFORINST 1510.1 provides policy and guidance for the utilization of the Reserve Training Support System for Air by COMNAVAIRESFOR commands.

C. ORACLE DATABASE MANAGEMENT SYSTEM

1. The System

Now that the basics of database design theory have been explained, we can present the specifics of the Naval Air Reserve prototype. The first step in presenting the prototype is to introduce the database management system that we chose to implement the design on. Once you are familiar with the features of ORACLE, we will discuss how they were used to incorporate the important attributes of an effective management information system into our design.

The ORACLE database management system is marketed by Relational Software, Incorporated,[5] of Menlo Park, California. The first copy of ORACLE was installed in June 1979. Relational Software has implemented ORACLE on Digital Equipment Corporation's (DEC) PDP 11/23 through 11/70 series processors using RSX-11M, IAS, and UNIX operating systems and the VAX 11/780 under VMS and UNIX. Starting in 1981, they also began offering versions that ran on International Business Machines(IBM) Corporation's 360, 370, 43xx, and 303x series processors under VM/CMS and MVS operating systems. The company recently announced that all of ORACLE, not just a subset, will be available to run on the IBM PC XT and PC AT.

2. System Features

ORACLE provides a wide range of features which are listed below:

a) Fully integrated data elements dictionary.

b) Provisions for use in both distributed and local/ central environments.

---

[5]Relational Software, Inc. has since changed its name to ORACLE, Inc.

c) Provides full support for IBM's relational language SQL(also called SEQUEL 2, for more details see subsection 4).

d) Fully reentrant code.

e) Multitask processing.

f) Interactive applications generator.

g) Full function report writing facility that includes both text processing and procedural language options.

h) Operates in batch and on-line(interactive) environments.

i) A host language interface and precompiler to allow COBOL, FORTRAN, and C applications programs to interface with SQL and the database.

j) Data communications support for all DEC equipment mentioned earlier and for IBM 360 and 370 systems.

k) Security and privacy mechanisms operating at the database, table, record, field, field value, and key levels.

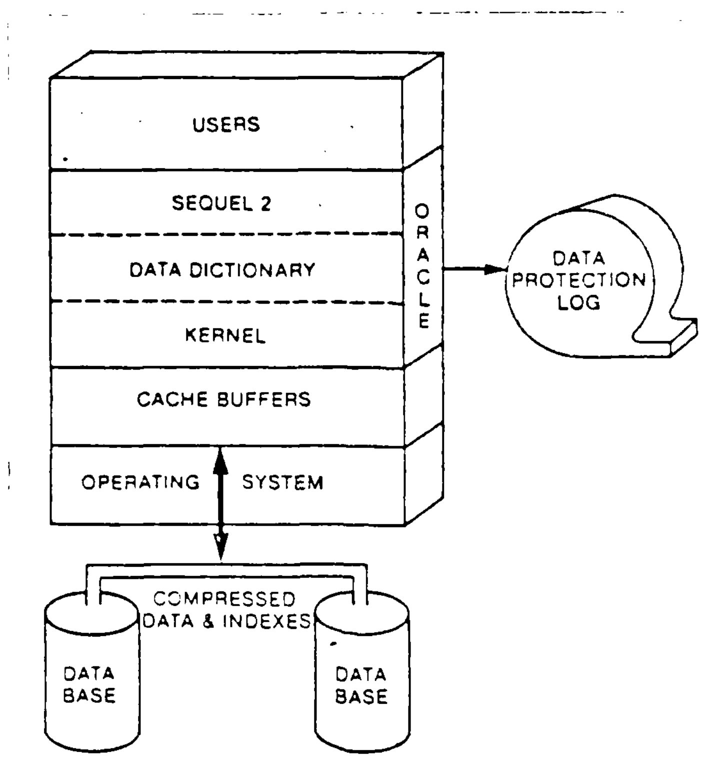
l) Restart/recovery options which are automatically triggered when failure is sensed.

Many of these features were used in our application and will be described in more detail in the appropriate sections later in this chapter.

3. System Components

ORACLE can be divided into three major components: the SQL data language, a data elements dictionary, and a kernel(see Figure 4.9). SQL is an English-like, high level language that will be discussed in detail in subsection 4.

The fully integrated data elements dictionary is a very powerful tool for us as database designers and for the Naval Air Reserve as the eventual end-users. The dictionary contains table, row, and column definitions, views of tables



[Ref. 17: p.112]

Figure 4.9 ORACLE Component Block Diagram

and data fields, and information about users and their data access privileges [Ref. 17: p. 111].

The table, row, and column definitions are assigned dynamically when they are created. Any modifications to the database are automatically entered without the need for the designer to reorganize the database or recompile the existing programs [Ref. 17: p. 112]. The dictionary also keeps track of the views of all tables, who created the view, and who has access to it. A view is a restriction placed on a table so that only selected columns and rows are visible to selected individuals. In other words, the database designer can specify what tables a user can see, and within those tables, what columns and rows that user can see. This feature allows information to be hidden from those individuals who should not have access to it. All security and control information on data and users are kept in the dictionary.

The third component of ORACLE is the kernel. The kernel allocates resources for ORACLE in the same manner that the operating system allocates resources for the computer system. The following paragraph, [**Ref. 17: p. 113**], best describes the kernel:

> The kernel automatically reallocates and reuses space from a central storage pool to optimize the physical location of related data items within the database. In addition, the kernel dynamically manages all ORACLE buffers, using an LRU (Last Record Update) caching algorithm to maximize reuse of core-resident information The kernel also enables ORACLE to support multiple concurrent batch and on-line terminal updates and queries to the database. It can overlap I/O operations up to the limit of the hardware configurations.

4. SQL

SQL is an English-like language that supports five functions: query, data definition, data manipulation, data control, and host language coupling. The English-like characteristics of SQL are ideal for non-procedural queries of the database. The user describes what he/she wants using English terms, such as shown in Figure 4.10, and ORACLE formulates a procedure to find the desired data. [**Ref. 18: p. 562**]

The query function relies on an operation called mapping for its success. Using Figure 4.10, the concept of mapping is that a known quantity (RUIC = 29604) is used to find a desired quantity (SSN and Last Name) from a relation

```
SELECT SSN, NAMELAST
FROM NAME
WHERE RUIC = 29604;
```

Figure 4.10 Non-Procedural Language

or table (NAME). The three basic terms used for queries are SELECT, FROM, and WHERE. SELECT is used to declare the desired information, FROM gives the relation(s) to be searched, and WHERE provides the known fact that must be satisfied by the desired data. Figure 4.11 provides some basic examples of the query facility. For a more detailed discussion of SQL, consult references 17 and 18.

The data definition language (DDL) is used to define tables, rows, columns, views, indices, and clusters (indices and clusters are discussed later in the section on improving system performance). The data definition language facility is primarily a tool for the database designer. However, once the database is installed, the DDL can be used by the user to adapt the system to meet the needs of the dynamic environment it is supporting. This facility describes the data structure that will be used by the other features of ORACLE. A data structure must first be defined with DDL before it can be accessed by the other SQL facilities. Figure 4.12 illustrates now two of the most common terms are used. The two examples are self-explanatory since they are English-like. The number in parentheses is the maximum number of spaces that a particular data field can occupy. NOT NULL is the SQL method for making a field mandatory.

The data manipulation language (DML) allows users to change the value of data stored in the database. There are three operations that can be performed: insertion, deletion,

1. Nested Mapping--Find the SSN and Name of all Reservists in the Houston area

```
SELECT SSN, NAMELAST
FROM NAME
WHERE RUIC IN
      SELECT RUIC
      FROM RESERVUNIT
      WHERE CITY = 'HOUSTON';
```

2. Count--How many different PNEC's are there at RUIC 29604?

```
SELECT COUNT (UNIQUE PNEC)
FROM NAME
WHERE RUIC = 29604;
```

3. Join--a join operation must occur when a query returns results from more than one table. The join operation is implicit.
   What are the SSN's and RUIC's of all reservist's having NEC 0612? (NEC's are stored in two tables-NAME contains the PNEC's and NOBC_SNEC contains the SNEC's)

```
SELECT NAME.SSN, NAME.RUIC
FROM NAME, NOBC_SNEC
WHERE NAME.PNEC = '0612'
OR NOBC_SNEC.SNEC = '0612';
```

Figure 4.11 ORACLE Query Function

```
CREATE TABLE NAME
(SSN NUMBER(9) NOT NULL,
 NAMELAST CHAR(20) NOT NULL,
 NAMEF CHAR(14));
```

Figure 4.12 Data Definition Language

and update. Figure 4.13 illustrates how these are accomplished.

```
1. Insertion

        INSERT INTO SPECIALIST
               SELECT NAME.SSN, NAME.RUIC
               FROM NAME, NOBC_SNEC
               WHERE NAME.PNEC = 0612
               OR NOBC_SNEC.SNEC = 0612;

2.  Deletion

        DELETE MOB_BILLTS
        WHERE AUIC = 32314;

3.  Update

        UPDATE NAME
        SET PAYGRADE = '05', DOR = 850115
        WHERE SSN = 123456789;
```

Figure 4.13 Data Manipulation Language

An insertion is used to to insert new data into a table or to extract data from one table and place it in another. Example 1 in Figure 4.13 illustrates the latter. In this case, the SSN and Reserve UIC for any Reservists who have a NEC of 0612 are extracted from their current table and placed in the SPECIALIST table. The data extracted from the original table now exists in two tables--the original table it was extracted from and the new SPECIALIST table. Notice that this example is a query operation linked with an insertion. The insertion provides a location for the results of the query to be deposited.

A deletion is used to delete a row or rows from a table. Example 2 of Figure 4.13 shows the deletion of all rows from MOB_BILLTS where the Active UIC is 32314 (assume this active unit was decommissioned).

Indexing can also be used to ensure that a column within a table has a unique value. In other words, a data element may only have a specific value assigned to it once in the table. The primary use of this feature in our prototype is with the data element 'SSN'. We created a unique index so that a person's SSN will only be listed once. This eliminates the possibility of a member's record being entered more than once. If this were to occur, the data in each record would be suspect. Figure 4.18 is an example of how we created a 'UNIQUE' index for SSN on the NAME table. Appendix D contains the list of indices used in the prototype.

```
CREATE UNIQUE INDEX SSN
NAME(SSN);
```

Figure 4.18 **Indexing**

Creating an in index improves performance but also increases the amount of data stored in primary memory. Additionally, if there are excessive indexes, each time a record is added to a table that is indexed, ORACLE updates each. The more indexes there are, the more work ORACLE has to do to keep them current. This will slow down the entire system. [Ref. 19: p. 201]

3. Clustering

Clustering of data together in the same physical space is a way of guaranteeing that two tables that are often joined together are also stored together. Clustering is totally transparent to all queries against the data.

b) Consistency checks--used to ensure that values in certain data fields are valid in relation to entries in other data fields. This is also done within one record or between records.

F. IMPROVING SYSTEM PERFORMANCE

A relational database system can be inefficient when data from very large tables or from several tables is retrieved. If there is no plan for storing related data, each search through the database must look at every record. In designing our prototype, we developed a plan that limits the number of records that are viewed before the correct one is found. This is done in ORACLE with indexing and clustering.

1. Indexing

Indexing is used to quickly locate the pages[9] of the database that contain specific records of a table. The ORACLE index is the same as the index in the back of a text-book. It provides a means of finding subjects without looking page by page through the book. The index in a book is organized in alphabetical order and allows you to find all incidences of the desired subject.

The indexing system of ORACLE uses this same principle. It uses its indexes to locate pages on disk that contain the desired records. This is an excellent way to shorten the time it takes to execute a query. The index eliminates the need to search the entire database to locate the desired record; only the page where the record resides must be searched.

---

[9]Database pages are physical areas on disk storage devices that hold the data in the database like paper pages hold the information in a book. [Ref. 19: p. 197]

ORACLE does not have the ability to verify that data contains a prescribed pattern. For example, RBSC has a pattern of four digits and then a letter. We can not check that this pattern is adhered to at input time. Consequently, RBSC is specified as character, with a maximum of five spaces in the data elements dictionary.

3. Reasonableness Checks

The final check is a reasonableness check. After the transaction is confirmed as valid and the format is correct, we must check to see if the value of the input data falls within reasonableness constraints. For example, for a two-day Weekend Training(WET), we do not want the number of drills entered to be greater than four. Oracle supports the following reasonableness checks:

1. Field Constraints--Each of these three are accomplished in ORACLE with the Interactive Applications Generator mentioned earlier.
   a) Range--checks the upper and lower limits of the data entry to ensure they are with in acceptable bounds.
   b) Completeness--used to ensure that mandatory fields have been assigned values and that fields with a set number of characters have been completely filled in.
   c) Code--used to verify that the contents of a data field contain a code that is authorized in a list of current codes.
2. Interrecord/Intrarecord--Each of these are also accomplished with the Interactive Applications Generator.
   a) Completeness checks--used to ensure that certain fields are filled in as a result of entries into other data fields. This occurs within a record and between records.

2. Format Checks

Format checks ensure that the data to be entered conforms to certain pre-defined criteria describing the physical structure of each data item. ORACLE accomplishes this through its active Data Elements Dictionary.

The data elements dictionary for our prototype is contained at Appendix A. It lists the the data field name, its type (number, character, or date), its length (this is the maximum length allowed), and whether the field is mandatory. Null means the field can be blank; Not Null means it must be filled. A field may be of type 'character' even if it contains only numerics. The choice is based upon whether arithmetic will be performed on the data field. Arithmetic operations are not permitted on character type fields. [Ref. 21: p. 8]

The data elements dictionary will perform the following format checks on all data input transactions automatically:

a) Length checks--ensure data does not exceed the maximum prescribed number of characters.

b) Character-type checks--ensure data is of the prescribed type.

In addition to the data elements dictionary enforcing prescribed format, our prototype has a series of input programs that contain additional format checks. These programs, created with the Interactive Applications Generator, allows the input operator to execute a "screen" that aids in the input process. Appendices B through F contain a picture of the screens and the interactive programs that create them. These input screens enforce the criteria that there must a fixed number of characters.[8]

---

[8]For example, RUIC will always contain five numbers, SSN will always contain nine numbers, and STATE will always contain two characters.

feature called AUDIT TRAIL that is supposed to record certain events. The designer was to specify what events, such as unauthorized access attempts, would trigger the audit trail. The audit trail would then take a picture of what happened and who attempted it.

After a futile attempt to implement this feature, we contacted ORACLE, INC. and were told that the feature did not work and was no longer supported. This is a weakness of some significance, but not fatal. This shortcoming was partially offset by procedures designed to protect the database.

E. DATA INTEGRITY

Data integrity involves the correctness of data. The purpose of enforcing data integrity rules is to ensure that correct data is entered into the database and that it remains correct while there. The techniques to enforce data integrity are transaction validation, format checks, and reasonableness checks. [Ref. 20: p. 218]. We will discuss what each is and how they were implemented in our prototype using ORACLE.

1. Transaction Validation

A transaction validation is concerned with ensuring that an input or modification is authorized. This was discussed in great detail in the Data Security section. ORACLE enforces this with the privilege facility. Only transactions authorized by the owner of the table are allowed (refer to Figure 4.16). Proper security is the first step toward sound data integrity.

```
CREATE VIEW OPS AS
SELECT NAME.SSN,NAMELAST,NAMEF,RANK_OR_RATE,DOR,
       CLRACCESS,ADDRESS,CITY,STATE,ZIP,HOMEPHON
FROM   NAME, RUIC_BILLETS
WHERE RUIC-BILLETS.DEPT='OPS';
```

Figure 4.17 Alternative View of Data

listed in the two SELECT lines are gathered for the individuals whose SSN's were collected from RUIC_BILLETS. This information is the subset called the OPS view. The view does not physically contain any data, so creating a view does not mean storing the same data twice. This would be very inefficient and present some difficult data maintenance problems.[7]

The procedure in Figure 4.17 should be repeated for each department. The view is created with the department head as the owner and controller of all privileges. The department head can look at the entire NAME table (by using the SELECT option shown in Figure 4.16) but can only manipulate the data in the OPS view of NAME. This arrangement allows the command to maintain control of the entire database and gives the department head freedom to manipulate his subset of the database.

Now that we have described how this prototype protects the data from unauthorized access on several levels, we must briefly discuss how an unauthorized access attempt can be detected. The latest version of ORACLE has a

[7]Storing the data twice about the same person would mean that if the data was changed, it would have to be updated in two places. The overhead required to do this would render views useless. This does not occur.

GRANTOR	GRANTEE	CREATOR	TNAME	TIMESTAMP	A	D	I	S	U
1. UPPER LEVEL									
WINSLOW	WINSLOW	WINSLOW	NAME	29-NOV-84	G	G	G	G	G
WINSLOW	WINSLOW	WINSLOW	LANGUAGE	04-DEC-84	G	G	G	G	G
WINSLOW	WINSLOW	WINSLOW	EDUCATION	05-DEC-84	G	G	G	G	G
WINSLOW	WINSLOW	WINSLOW	RESERVUNIT	20-DEC-84	G	G	G	G	G
WINSLOW	WINSLOW	WINSLOW	MOB_BILLTS	20-DEC-84	G	G	G	G	G
WINSLOW	WINSLOW	WINSLOW	CLEARANCE	20-DEC-84	G	G	G	G	G
WINSLOW	WINSLOW	WINSLOW	NOBC_SNEC	16-JAN-85	G	G	G	G	G
WINSLOW	WINSLOW	WINSLOW	RUIC_BILLET	16-JAN-85	G	G	G	G	G
WINSLOW	WINSLOW	WINSLOW	MOBILIZE	16-JAN-85	G	G	G	G	G
WINSLOW	WINSLOW	WINSLOW	DRILLS	16-JAN-85	G	G	G	G	G
WINSLOW	WINSLOW	WINSLOW	ACDUTRA	28-JAN-85	G	G	G	G	G
2. MIDDLE LEVEL									
WINSLOW	SEEGER	WINSLOW	NAME	30-JAN-85	Y		G	G	G
WINSLOW	SEEGER	WINSLOW	RUIC_BILLET	04-FEB-85	Y		Y	Y	Y
WINSLOW	SEEGER	WINSLOW	EDUCATION	05-DEC-84	Y		G	G	G
WINSLOW	SEEGER	WINSLOW	LANGUAGE	05-DEC-84	Y		G	G	G
WINSLOW	SEEGER	WINSLOW	DRILLS	16-JAN-85	Y		G	G	G
WINSLOW	SEEGER	WINSLOW	CLEARANCE	20-DEC-84	Y		Y	Y	Y
WINSLOW	SEEGER	WINSLOW	NOBC_SNEC	16-JAN-85	Y		G	G	G
WINSLOW	SEEGER	WINSLOW	ACDUTRA	28-JAN-85	Y		G	G	G
WINSLOW	SEEGER	WINSLOW	RESERVUNIT	20-DEC-84	Y		Y	Y	Y
WINSLOW	SEEGER	WINSLOW	MOB_BILLTS	20-DEC-84	Y		Y	Y	Y
3. LOWER LEVEL									
SEEGER	KURTH	WINSLOW	DRILLS	16-JAN-85			Y	Y	Y
SEEGER	KURTH	WINSLOW	ACDUTRA	28-JAN-85			Y	Y	Y
SEEGER	KURTH	WINSLOW	NAME	30-JAN-85			Y	Y	Y
SEEGER	KURTH	WINSLOW	EDUCATION	05-DEC-84			Y	Y	Y
SEEGER	KURTH	WINSLOW	LANGUAGE	05-DEC-84			Y	Y	Y
SEEGER	KURTH	WINSLOW	NOBC_SNEC	16-JAN-85			Y	Y	Y

KEY:

A - Alter (the current record structure)
D - Delete (a current record)
I - Insert (a new record)
S - Select (a current record)
U - Update (current data)
G - Grant (has the privilege and can grant to others)
Y - Yes (has the privilege)

Figure 4.16 Three-level Privilege Structure privileges on all tables and the authority to grant those privileges to others.

The middle level shows the set-up for middle level managers, such as a department head. Note that Seeger (acting as Operations department head) has less privileges than the upper level. Additionally, he has only limited authority to grant privileges to others.

The lower level of privilege options is for those individuals who perform record input and update. These individuals have update and insert must receive their privileges from an authorized user in the upper two tiers of the privilege scheme. This ensures privileges are controlled at a high level and that managers know who have access to the data in the database.

Note that only the upper level has the delete privilege. This privilege is reserved for this level because we are not allowing any deletions in our database prototype. If a record is to be deleted, the input operator marks the record for deletion and the person holding DBA privileges will periodically move the deleted records to a history file.

The security mechanism for records, data fields, and data field values is the view. A view is a logical partitioning of data so that the user only sees a subset of the table (or tables). A view is like a window in a building. It allows people to look at only a certain portion of what's inside. When a view is created it allows specified people to see portions of the database.

Figure 4.17 shows how a view was created for the prototype. This view allows the Operations department head to see only the Operations department subset of the table "NAME." This view requires gathering information from two tables. First, all SSN's from the RUIC_BILLETS table with a Department of 'OPS' are chosen. Then, all the data fields that logically restrict the data that a user can see. We will discuss each of these methods in the following paragraphs, using examples from the prototype.

The use of privileges to control access was mentioned briefly in an earlier section. To review, the following privileges can be granted (or revoked) by the owner of a table:

a) SELECT
b) INSERT
c) UPDATE
d) DELETE
e) ALTER
f) INDEX
g) CLUSTER

When a user attempts to manipulate the data in a table, ORACLE will first verify that the operation has been authorized by the owner of the table. If that particular privilege has not been granted, ORACLE will not allow the operation and generate an error message saying that the table does not exist. In other words, if an unauthorized operation is attempted by a user, ORACLE will mask the table as if it does not exist. The table is blocked by the system from unauthorized access.

It is the responsibility of the owner of the table to carefully grant privileges to selected users. Each users' needs must be analyzed on a case basis. Each privilege granted must be justified before it is granted. Granting of privileges must be closely tied to job requirements. Figure 4.16 shows three levels of privilege options.

The upper level is for the database administrator. In the prototype, Winslow serves as the DBA[6] and has all

---

[6] In actuality, the DBA has more privileges than those shown. This prototype was designed on ORACLE installed on the Computer Science VAX 11/780 and a computer science staff member retained full DBA privileges.

Winslow now has access to all application programs and facilities that are available to the user under the VMS operating system. He chose ORACLE in our example and typed "ORACLE" next to the "$" prompt. The "$" appears as a prompt at all times when you are in the VMS environment. Winslow then types "UFI" (for User Friendly Interface) to enter that particular facility. UFI is the primary means that a database designer or user talks to ORACLE using the Data Definition, Data Manipulation, and Data Control languages.

ORACLE requires the user to log on again so that it can check its own list of access privileges for proper authorization. This is a two-tiered process. First, the username/password combination is checked for access to the database itself. An authorization failure will result in the user being asked to log on again (in case the user typed the information incorrectly). ORACLE will tolerate two such failures and will exit to VMS if a third occurs. If successful, the message shown in Figure 4.15 will appear, followed by the "UFI>" prompt. We show how to enter the UFI facility, but the log on procedure for all of the ORACLE facilities is the same.

4. Data Access

The second tier of user authorization cross-references the username against a table that records who owns what tables and who has been granted privileges on those tables. This is an extremely powerful feature of ORACLE that provides a high level of security on the data resident in the database.

This security mechanism operates at the table, record, data field, and data field value level. It works using two different methods; through the control of privileges granted on a table and through the creation of views on' to the VMS operating system and ORACLE database management is shown in Figure 4.15 . The user must type in his name and then his password next to the appropriate prompts. This action places the user in the operating system environment. In this case, Winslow has successfully logged on to the VAX 11/780 computer and the VMS operating system. Note that a string of X's appear where the password is typed. This is an additional security feature that hides your password from anyone looking at your screen. In most cases, the password would be invisible, but we show 'xxxxxx' to indicate a masked password.

```
   You are connected to VAX/VMS 11/780.

ENTER USERNAME: WINSLOW
ENTER PASSWORD: xxxxxx
  Welcome to the C.S. Department's VMS 3.7 system.
  Current Software Versions: VMS 3.7        FORTRAN 3.5
                             PASCAL 2.5     ORACLE 4.0
$ ORACLE
$ UFI
  ORACLE Utilities, Copyright(c) 1979,1980,1981,1982,RSI
  UFI version 3.5 - on Tue Jan 29 18:46:24 1985
   Connecting to: ORACLE V4.0.12 - Production
ENTER USERNAME: WINSLOW
ENTER PASSWORD: xxxxxx
UFI>exit
     Logged off from ORACLE.
```

Figure 4.15 User Authorization (Log on) Procedures protected from unauthorized users and the data must be logically partitioned so that only those users who have access to the database and the need to know are authorized to see specific data.

2. Navy ADP Security Program

OPNAVINST 5239.1A (DON ADP Security Manual) requires a combination of hardware, software, and administrative procedures to protect data from unauthorized disclosure, modification, or destruction. Adequate protection is also required when the system is distributed or allows multiple users. RTSS is such a system and ORACLE accommodates this requirement with a lock out feature that will not allow a user to access a record that is being used by another user. This eliminates the problem of two users changing the same data and only the last of two being recorded. With lock out, each user sees the current data in turn and knows what he is changing.

This instruction also calls for an audit trail for cases where the data is of a sensitive nature. As will be discussed later, the ORACLE audit trail feature is not supported. With this exception, ORACLE can comply with these DON guidelines for ADP security.

3. Database Access

The first barrier for stopping unauthorized access is the computer center door. Physically limiting the people who enter the terminal room can help stop some unauthorized accesses. However, if the size of the organization or the number of users is large, this can be a time-consuming, expensive, and ineffective method of enforcing access limits.

The most common method of access authorization involves using a password scheme. The process of 'logging

```
GRANT SELECT, INSERT, UPDATE
ON NAME
TO ADAMS
WITH GRANT OPTION;
```

Figure 4.14 Data Control Language

The final function of SQL is the host language interface. The host language interface allows SQL statements to be included in FORTRAN, COBOL, and C programs. A precompiler (part of the ORACLE package) intercepts these statements and converts them into valid statements of the host language. The host language interface and precompiler may be used with a main menu to make entry into ORACLE's facilities easier and more user-friendly.

D. DATA SECURITY

1. Unauthorized Access

The Naval Air Reserve prototype must have the appropriate security measures built in to minimize the risk of unauthorized users viewing, modifying, inserting, or deleting data. There must be security provided to protect against unauthorized access from both local and remote sites. The remote site considerations involve securing communication links and transmission lines in a manner appropriate for the information that can be accessed. We will not address this aspect of security as it is beyond the scope of this thesis. Additionally, we will not address the area of physical security of the computer installation.

Protecting the database and the data are two distinct issues we will deal with. The database must be

An update is used to change the value of one or more data fields in either single or multiple rows. Example 3 of Figure 4.13 shows how to record the promotion of the member with SSN 123456789 to Commander (paygrade O5) and his Date of Rank to 15 January 1985.

The fourth function of SQL is data control. The data control language(DCL) enables the user to specify who will have access to his/her data and to enforce data integrity constraints. The phrase 'their data' implies ownership by the user. In fact, this is the case. The person who creates a table, 'owns' it and has the authority (and responsibility) to designate who else can interface with that table and its stored data. The following privileges, [Ref. 19: p. 182], can be granted by the owner of a table:

a) SELECT--data in your table.
b) INSERT--rows in your table.
c) UPDATE--columns in your table.
d) DELETE--rows from your table.
e) ALTER--the number of columns in your table or the characteristics of a column.
f) INDEX--create an index on a column of your table.
g) CLUSTER--your table with other tables.

The owner of the table has the authority to grant or revoke any or all of these privileges. The owner can also grant privileges to all users by using the term PUBLIC. Additionally, if the owner specifies GRANT OPTION when he grants privileges, the recipient of the privilege can grant that privilege to others. A example of data control language is shown in Figure 4.14 .

The use of the data control facility to enforce data integrity is essential to the development of a database maintenance policy. This is discussed in detail in the section on data integrity.

This is a way of shortening the time required to join the data of two tables when responding to a user-generated query. [Ref. 22: p. 25]

Figure 4.19 shows how we created a cluster to store the two tables together where 'RUIC' for each is the same. The cluster is created first, then each table is added to it. Notice that it is a three step process.

```
        1.  CREATE CLUSTER RUIC_AUIC
            (RUIC NUMBER);

    Cluster Created.*

        2.  ALTER CLUSTER RUIC_AUIC
            ADD   TABLE RESERVUNIT
            WHERE RESERVUNIT.RUIC = RUIC_AUIC.RUIC;

    Cluster Altered.*

        3.  ALTER CLUSTER RUIC_AUIC
            ADD   TABLE RUIC_BILLETS
            WHERE RUIC_BILLETS.RUIC = RUIC_AUIC.RUIC;

    Cluster Altered.*

* indicates response from ORACLE
```

Figure 4.19 Clustering

All records from these two tables where the 'RUIC' is the same will be stored together so that retrieval will be faster. These two tables were clustered because the queries that will be routinely generated will involve data about the RUIC that is contained in both tables. A cluster is more efficient than an index in this case because only one search must be accomplished to find the data in two tables. An index would only locate the page that the records are on in two different tables and then each would have to be searched.

G. RECOVERY

There is a degree of risk involved with storing information about an organization in a computer. Computers stop unexpectedly, disk heads crash, operators drop disks, programs have bugs, people input incorrect data, and procedures are ignored [Ref. 16: p. 415]. These occurrences (called media failures) can not be allowed to stop or slow down the organization. The data must be reconstructed to its original state. The most common way to ensure that the database can be reconstructed when some catastrophic event occurs is a recovery scheme.

1. After Image Journal

The scheme ORACLE uses is roll forward recovery. ORACLE uses an AFTER IMAGE JOURNAL to implement this roll forward scheme. Recovery is made possible by keeping a separate parallel journal of each transaction written to the database. A physical record of all changes to the database is produced and this can be used to reconstruct from the point where the loss occurs. After Image Journaling is completely transparent to the user. [Ref. 22: p. 26]

The procedure is as follows:

1. ORACLE records each event that modifies the database and places it into a file.
2. When the journal grows to a certain size or after a specified period of time, the files are copied from disk to tape.
3. Each file is given a sequential number to establish the correct order of each occurrence.
4. When a media failure occurs, the last copy of the database known to be good is retrieved. The first After Image Journal that was created after this good copy of the database is then applied. All After

Image Journals that were created after this first one are then applied in order until the database is completely reconstructed.

An effective recovery scheme using proven software and rehearsed procedures is essential to the operation of an organization that maintains its files on a computer system. The roll forward recovery technique that is implemented in ORACLE is an effective method. The After Image Journals that are created provide a fairly low-risk means of ensuring that database integrity can be maintained, even in the event of media failure. The key to effective recovery is in the procedures established by the computer center management. These procedures must be practiced prior to a media failure so that when one does occur, its impact is minimal.

H. REPORT WRITING

A modern database management system must have the facilities to process reports for the organization. It must be able to deal with straight text and a combination of text and database query results. This feature can relieve an organization of much of its old, outdated programs that extract data from many sources and presents them in report format. Because the data is consolidated in one database, these reports are usually easier to write and, more importantly, easier to maintain.

ORACLE has a facility for text formatting and report writing that is capable of database retrieval. There is an interactive feature that allows inputting a value for a variable before the data is retrieved and formatted. This is particularly useful when a report is being prepared for a Reserve Unit and the computer operator supplies the RUIC before the data is retrieved. It can also be used for a document for an individual reservist by supplying their SSN.

We attempted to develop several reports that were representative of those in existence for the Reserves. In general, we found the report writer cumbersome and difficult to work with. For a series of reports to be produced, the computer operator must issue several commands or write a program in either COBOL, FORTRAN , or C. Use of these languages is what we are trying to eliminate; this is a negative factor in many of the fourth generation languages today.

In each report we attempted, we discovered an apparent "bug" in the ORACLE software. This problem has been encountered by all those we know who have tried the report writing facility. This is not to say the data can not be accessed, for the query facility in the User Friendly Interface can accommodate short, simple reports. But the problem is a major obstacle in ORACLE's consideration as an integrated database management system for the Naval Reserve.

I. ORACLE'S GRADE

ORACLE is a relational database management system that provides a fairly easy means of developing a database prototype. Its active data elements dictionary is an excellent feature that aided us as designers and gave us a large degree of flexibility when changes were required.

ORACLE has the necessary security, data integrity, and performance enhancement features to satisfy this application for the Naval Reserve. In general, the features listed on pages 59 and 60 make this system a desireable product. However, the problems noted with the report writer feature are critical in the choice of a software system. If the support for this feature is improved and the problems are corrected, then this product would be a good choice for the Naval Reserve. A caution--this thesis did not set out to evaluate the software market for database systems. In critiquing ORACLE, we are not comparing it to other systems for performance or price. That analysis must be completed before an informed selection can be made.

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

After eight years of planning and implementation efforts, RTSS is not getting the job done. The outlook for a new contractor to effect a major breakthrough on a system that uses old hardware and ancient (by today's standards) software is grim. The Navy tried to upgrade just the training software and failed. Substantial progress towards the goal of an integrated training and mobilization database for the Reserves cannot occur using the present technology.

At CNAVRES, manual intervention is still a must in order to obtain training and mobilization data. RTSS (Air) remains linked to a fourteen year old ATSS system whose software still cannot transfer records from one site to another. Software updates are late, insufficient, or non-existent. The functional sponsorship for the entire system lies elsewhere. RTSS (TM) is dependent on an IMAPMIS AIS that is in need of extensive revision, again by a functional sponsorship outside the Reserves.[10] The RTSS systems are redundant and poorly designed. Progress since inception of the systems has been glacial.

The skeletal staff at Code 46 is determined but helpless. Overwhelmed with milestone management, they lack the time and budgetary control for effective AIS strategic planning. They recognize the need for redesign of the software away from the current system and applications software, but have not yet been successful in converting those needs into

---

[10]Code 45 (Mobilization/Readiness) was extremely frustrated with the state of ADP support for his needs. His RTSS/IMAPMIS experience has served to make him adamant that any future RTSS must not be associated with or dependent on any other system. [Ref. 23]

POM items. Money for ADP has been cut back in past years. Virtually all ADP funds are being funnelled into maintaining the current inept system. Innovation has no priority on the mobilization battlefield.

Code 461 is presently trying to find the money to convert the file structure of RTSS from the RSTS/E operating system to the VMS operating system. This in-house, i.e. non-NAVAIR, effort will take considerable contractor support. A PDP 11/45 stands ready at New Orleans for research and support in the conversion efforts.

The DDN problem must also be resolved. Conversion of the current system to allow terminal to host processing on the DDN is still an issue. Estimates for initial protocol conversion for the RSTS/E operating system by various contractors have ranged from $150,000 to $300,000, with approximately $50,000 required annually for maintenance and support. Yet wide area network interface products already exist for VAX(VMS) machines, and the progress in local area networks for microcomputers such as Z-150s yields a new product monthly. The major studies cited previously in this document have looked at both Navy and Reserve AIS problems and determined that cumbersome redesigning efforts on obsolete systems are a waste of valuable resources. Off-the-shelf technology is available <u>today</u>.

All of these issues and questions can best be resolved by those trained and experienced in dealing with them. The Information Age is a reality whose workings are fast becoming the modus operandi for the Navy's business. The fields of networking, office automation, database management, and requirements analysis are full of pitfalls in which to lose valuable time and money. More and more dollars in the ADP world are going towards the labor costs of software maintenance, and less dollars for hardware that is becoming cheaper and more effective. Outside contractor sourcing for software services has become a way of life for the Navy. Yet, for the cost of some additional ACDUTRA drills, the Reserves have many of those resources readily available. There are many individuals who are not only skilled in the intricacies of AIS management, but well versed in the unique needs of the Reserves. The Information Systems Support Unit must be implemented if the Reserves are to efficiently and effectively plan for the automation of training and mobilization.

Software prototyping using off-the-shelf fourth generation query and database technology is fast becoming an acceptable alternative to the lengthy and complex methodology traditionally used. Certainly, no system can hope to be free from potential error. Management must still control, coordinate, and plan. Users must work closely with developers until they're skilled enough to go it alone. The primary advantage of prototyping is the ability to get results quickly, and then to continually change as problems arise or, more likely, needs change. This cycle of feedback and change is not available using the software and requirements analysis techniques of the past.

B. RECOMMENDATIONS

Several alternatives face the Reserves as they attempt to fix some rather extensive AIS problems:

1. The Reserves could simply continue on the present course of letting the Active Navy design and implement a follow-on system, as may be the case with the present Source Data System plans and efforts. The Reserves will have to live with whatever that system becomes, and it is only now in the planning stages for including Reserve requirements. However, there is a notable history of the Reserves taking a back

seat when it comes to sharing common resources; the hand-me-down nature of support received from ATSS facilities by sharing PTSS sites is painfully applicable. The delay before SDS implementation also means another five years (approximately) of chaos in handling training and mobilization data.

2. A major effort could be made to redesign the current applications software and file structure to run on the more efficient and DDN compatible VAX(VMS) hardware. In terms of dollars and results, this would probably be the cheapest way to 'fix' the present disarray of applications. It would leave just that: a 'fixed' antiquated system of applications with a database system that is really a file management system and has little or no room for expansion or major changes.

3. Re-designing with both new hardware and software is a third option which is not overwhelming when management acknowledges that they still must do things manually. This option brings to the Reserves the functional sponsorship of a system of their own. Design from the beginning can best serve their own unique needs, both present and future. Field planning teams, under the guidance of an ISSU could gather and forward a set of initial requirements for a prototype. The basic requirements of DDN, VAX(VMS), and (large) microcomputer compatibility can be met by several products on the market today. A benchmark or standard could be developed by the ISSU for a competitive runoff of the eligible fourth generation languages. The ISSU could then pick a candidate to buy for their prototyping tool, and the Reserves would be on their way to the 1980's.

We are naturally biased in the selection of the third alternative as our own recommendation. Our efforts at prototyping are only the first iteration and much remains to be done. We were disappointed with the lack of support we found in the report writing facility of the ORACLE software package, but were impressed with the ease of prototyping using a relational software package. The next step in this effort should be to develop report writing once the bugs are fixed. Additionally, experimentation with a medium-sized database to evaluate the efficiency of the system is necessary.

The ISSU and prototyping will go a long way towards bringing the Reserve manager of today some meaningful AIS leadership and technology. The recent upgrading of OP-945 (Information Systems Division) to a flag billet, and the use of new software productivity tools at NMPC-4 are indications that the Navy means business in its 'Battle of the Byte'. We strongly urge the Reserves to take note, take heart, and move toward a sounder and stronger AIS future.

APPENDIX A

DATA ELEMENTS DICTIONARY

TNAME	CNAME	COLTYP	WIDTH	Comments
NAME	SSN	NUMBER	9	NOT NULL
	NAMELAST	CHAR	20	NOT NULL
	NAMEF	CHAR	20	NOT NULL
	NAMEMI	CHAR	3	NOT NULL
	RANK_OR_RATE	CHAR	5	NOT NULL
	CLRACCESS	CHAR	6	
	PEBD	NUMBER	6	NOT NULL
	DOB	NUMBER	6	
	SEX	CHAR	1	
	ETHNIC	CHAR	1	
	RACE	CHAR	1	
	DEPEN	CHAR	2	
	STREET	CHAR	30	NOT NULL
	CITY	CHAR	18	NOT NULL
	STATE	CHAR	2	NOT NULL
	ZIP	NUMBER	9	NOT NULL
	HOMEPHON	NUMBER	10	NOT NULL
	RUIC	NUMBER	5	
	TRUIC	NUMBER	5	
	RCDSFLAG	CHAR	1	
	PRECD	NUMBER	8	
	PAYGRADE	CHAR	1	
	EOS	NUMBER	6	
	MOD	NUMBER	1	
	EXPAA	NUMBER	6	
	DOR	NUMBER	6	
	BUSPHONE	CHAR	15	
	COCI	CHAR	9	
	PNOBC_OR_PNEC	CHAR	4	
	SUBCD	CHAR	5	
	LINEAL	NUMBER	8	

TNAME	CNAME	COLTYP	WIDTH	Comments
RUIC_BILLETS	RUIC	NUMBER	5	NOT NULL
	RBSC	CHAR	5	NOT NULL
	O_E_INDEX	CHAR	1	NOT NULL
	ABSC	CHAR	5	NOT NULL
	NOBC_PNEC	CHAR	4	
	ARATE	CHAR	5	
	IRATE	CHAR	5	
	SSN	NUMBER	9	
	DESCRIPTION	CHAR	30	
	DEPT	CHAR	3	NOT NULL

TNAME	CNAME	COLTYP	WIDTH	Comments
MOBILIZE	AUIC	NUMBER	5	NOT NULL
	MOBTIME	NUMBER	4	
	MOBDATE	NUMBER	6	
	MOBEVCODE	NUMBER	2	
	RDD	NUMBER	6	
	MAJCLMT	CHAR	2	
	MCBSPNR	CHAR	2	
	ALLOFF	NUMBER	3	
	ALLENL	NUMBER	3	
	ADRLOFF	NUMBER	1	
	ADRLENL	NUMBER	1	
	ADMSTROF	NUMBER	3	
	ADMSTREN	NUMBER	3	

TNAME	CNAME	COLTYP	WIDTH	Comments
DRILLS	SSN	NUMBER	9	NOT NULL
	DRILLTYPE	CHAR	1	NOT NULL
	NUMDRILLS	NUMBER	2	NOT NULL
	DRILLDATE	NUMBER	6	

TNAME	CNAME	COLTYP	WIDTH	Comments
ACDUTRA	SSN	NUMBER	9	NOT NULL
	ACDUDATE	NUMBER	6	NOT NULL
	ACDUTYPE	CHAR	1	NOT NULL
	ACDUDAYS	NUMBER	2	
	AUIC	NUMBER	5	
	ACDUPAY	CHAR	1	

TNAME	CNAME	COLTYP	WIDTH	Comments
NOBC_SNEC	SSN	NUMBER	9	
	NOBC_OR_SNEC	CHAR	4	

TNAME	CNAME	COLTYP	WIDTH	Comments
LANGUAGE	SSN	NUMBER	9	NOT NULL
	LANG	CHAR	2	NOT NULL
	LANGCOMP	NUMBER	1	
	LANGREAD	NUMBER	1	
	LANGSPK	NUMBER	1	
	LANGWRT	NUMBER	1	
	LANGEXP	CHAR	1	

NAME	CNAME	COLTYP	WIDTH	Comments
DUCATION	SSN	NUMBER	9	NOT NULL
	EDUCYEAR	NUMBER	2	
	EDUCLEVEL	CHAR	1	
	EDUCMAJOR	NUMBER	2	

NAME	CNAME	COLTYP	WIDTH	Comments
ESERVUNIT	RUIC	NUMBER	5	NOT NULL
	AUIC	NUMBER	5	NOT NULL
	ADDRESS	CHAR	30	NOT NULL
	CITY	CHAR	18	NOT NULL
	STATE	CHAR	2	NOT NULL
	ZIP	NUMBER	5	NOT NULL
	COMMPHONE	NUMBER	10	
	AUTOVPHONE	NUMBER	7	
	RESCENUIC	CHAR	5	
	RUICLNAME	CHAR	25	
	AUICLNAME	CHAR	25	
	RESCENLNAME	CHAR	25	
	APC	CHAR	7	

NAME	CNAME	COLTYP	WIDTH	Comments
OB_BILLTS	AUIC	NUMBER	5	NOT NULL
	ABSC	CHAR	5	NOT NULL
	DSCRPTION	CHAR	30	
	DESIG_RATE	CHAR	5	
	PAYGRADE	CHAR	2	

NAME	CNAME	COLTYP	WIDTH	Comments
LEARANCE	SSN	NUMBER	9	NOT NULL
	CLRACCESS	CHAR	6	NOT NULL
	SECAGENCY	CHAR	1	NOT NULL
	SECTYPE	CHAR	1	NOT NULL
	SECDATE	NUMBER	6	NOT NULL

APPENDIX B

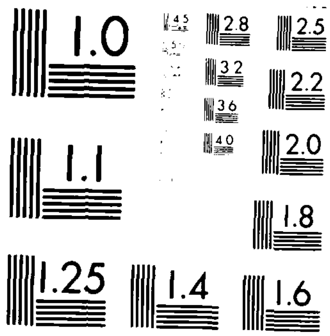
INTERACTIVE APPLICATIONS GENERATOR OUTPUT

```
- - - - - RESERVE UNIT INFORMATION - - - - - -

SERVE UNIT LONG NAME                          RESERVE UIC
XXXXXXXXXXXXXXXXXXXXXXXX                      XXXXX

REET                                          COMMERCIAL PHONE NO.
XXXXXXXXXXXXXXXXXXXXXXXXX                     XXXXXXXXXX

TY                         ST  ZIP            AUTOVON PHONE NO.
XXXXXXXXXXXXX              XX  XXXXX          XXXXXXX

                     RESERVE CENTER UIC
                           XXXXX

TIVE UNIT LONG NAME                          ACTIVE UIC
XXXXXXXXXXXXXXXXXXXXXX                       XXXXX


AE MODE: REPLACE   PAGE 1                              COUNT:  ?
```

AD-A156 827 A DESIGN METHODOLOGY AND PROTOTYPE FOR THE RESERVE TRAINING SUPPORT SYSTEM(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA M J WINSLOW ET AL. MAR 85 2/2

UNCLASSIFIED F/G 5/9 NL

END
FILMED
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

- - - - - - ACDUTRA INFORMATION - - - - - -

Soc. Sec. No.	Type	Days	Date	Pay	AUIC
xxxxxxxxx	x	xx	xxxxxx	x	xxxxx
xxxxxxxxx	x	xx	xxxxxx	x	xxxxx
xxxxxxxxx	x	xx	xxxxxx	x	xxxxx
xxxxxxxxx	x	xx	xxxxxx	x	xxxxx
xxxxxxxxx	x	xx	xxxxxx	x	xxxxx

Char Mode: Replace   Page 1   Count: 0

- - - - - - - RESERVE UNIT BILLET STRUCTURE - - - - - - -

Reserve UIC	RBSC	Off/Enl	Description
xxxxx	xxxxx	x	xxxxxxxxxxxxxxxxxxxxxx

ABSC
11111

Soc. Sec. No.	NOBC/PNEC
111111111	1111

Intended Rank/Rate	Actual Rank/Rate
xxxxx	xxxxx

CHAR MODE: REPLACE   PAGE 1   COUNT: 1

- - - - - - - - PERSONAL INFORMATION - - - - - - - -

SOC. SEC. NO. xxxxxxxxx

First Name xxxxxxxxxx | MI x | Last Name xxxxxxxxxxxxxx | Rank/Rate xxxxxx | Status x

Street xxxxxxxxxxxxxxxxxxxxxxxx | Home Phone xxxxxxxxxx

City xxxxxxxxxxxxxx | ST xx | ZIP xxxxx | Business Phone xxxxxxxxxxxxxxx

Reserve UIC xxxxx | Training UIC xxxxx

CHAR MODE: REPLACE PAGE 1 COUNT: 1

SOC. SEC. NO. xxxxxxxxx

Birth ate xxxxxx | Sex x | Race x

PEBD xxxxxx | DEP CODE xx | Clearance xxxxxx

Pay Grade xx | Date of Rank xxxxxx | EOS xxxxxx

Precedence No. xxxxxxxx | MOD xxxx | EXRAA xxxxxx

Lineal No. xxxxxxxx | NOBC/PNEC xxxxx

COCI xxxxxx | Subspecialty xxxx

CHAR MODE: REPLACE PAGE 2 COUNT: 1

- - - - - - - DRILL INFORMATION - - - - - - -

SOC. SEC. NO.	TYPE	NUMBER	DATE
XXXXXXXXX	X	XX	XXXXXX
XXXXXXXXX	X	XX	XXXXXX
XXXXXXXXX	X	XX	XXXXXX
XXXXXXXXX	X	XX	XXXXXX
XXXXXXXXX	X	X	XXXXXX

CHAR MODE: REPLACE PAGE 1   COUNT: 0

APPENDIX C

PERSONAL INFORMATION SCREEN

```
; ***** THIS PROGRAM IS THE INTERACTIVE APPLICATIONS
; ***** GENERATOR CODE THAT IS COMPILED TO DEVELOP THE
; ***** PERSONAL SCREEN. IT HAS A FILE TYPE OF .INP -
; ***** THE COMPILED CODE HAS A FILE TYPE OF .FRM.
;Application Title :
PERSONAL INFORMATION
;ORACLE workspace size :

;Block name / Description :
PERSNAME
;Table name :
NAME
;Check for uniqueness before inserting Y/N :
N
;Display/Buffer how many records :
1/25
;Field name :
SSN
;Type of field :
NUMBER
;Length of field / Display length / Query length :
9
;Is this field in the base table Y/N :
Y
;Is this field part of the primary key Y/N :
Y
;Field tc copy primary key from :

;Default value :

;Page :
1
;Line :
4
;Column :
35
;Prompt :
Soc. Sec. No.
;Display prompt above field Y/N :
n
;Allow field to be entered Y/N :
y
;SQL>

;Is field fixed length Y/N :
y
;Auto jump to next field Y/N :
y
;Convert field to upper case Y/N :
n
;Help message :
Enter Member's Social Security Number_ This field is mandatory!
;Lowest value :

;Highest value :
```

```
;Field name :
NAMEF
;Type of field :
CHAR
;Length of field / Display length / Query length :
12
;Is this field in the base table Y/N :
Y
;Is this field part of the primary key Y/N :
N
;Default value :

;Page :
1
;Line :
7
;Column :
10
;Prompt :
First Name
;Display prompt above field Y/N :
y
;Allow field to be entered Y/N :
y
;Allow field to be updated Y/N :
y
;SQL>

;Is field mandatory Y/N :
y
;Is field fixed length Y/N :
n
;Auto jump to next field Y/N :
y
;Convert field to upper case Y/N :
y
;Help message :
Enter Member's First Name - This field is mandatory!
;Lowest value :

;Highest value :

;Field name :
NAMEMI
;Type of field :
CHAR
;Length of field / Display length / Query length :
1
;Is this field in the base table Y/N :
Y
;Is this field part of the primary key Y/N :
N
;Default value :

;Page :
1
;Line :
7
;Column :
23
;Prompt :
MI
;Display prompt above field Y/N :
Y
;Allow field to be entered Y/N :
Y
;Allow field to be updated Y/N :
Y
```

```
;SQL>
;Is field mandatory Y/N :
N
;Is field fixed length Y/N :
Y
;Auto jump to next field Y/N :
Y
;Convert field to upper case Y/N :
Y
;Help message :
Enter Member's Middle Initial-Leave blank if unknown or NMN
;Lowest value :

;Highest value :

;Field name :
NAMELAST
;Type of field :
CHAR
;Length of field / Display length / Query length :
20
;Is this field in the base table Y/N :
Y
;Is this field part of the primary key Y/N :
N
;Default value :

;Page :
1
;Line :
7
;Column :
26
;Prompt :
Last Name
;Display prompt above field Y/N :
y
;Allow field to be entered Y/N :
y
;Allow field to be updated Y/N :
y
;SQL>
;Is field mandatory Y/N :
y
;Is field fixed length Y/N :
n
;Auto jump to next field Y/N :
y
;Convert field to upper case Y/N :
y
;Help message :
Enter Member's Last Name - This Field is Mandatory!
;Lowest value :

;Highest value :

;Field name :
RANK_OR_RATE
;Type of field :
CHAR
;Length of field / Display length / Query length :
5
;Is this field in the base table Y/N :
Y
;Is this field part of the primary key Y/N :
N
;Default value :
```

```
;Page :
1
;Line :
7
;Column :
50
;Prompt :
Rank/Rate
;Display prompt above field Y/N :
y
;Allow field to be entered Y/N :
y
;Allow field to be updated Y/N :
y
;SQL>

;Is field mandatory Y/N :
y
;Is field fixed length Y/N :
n
;Auto jump to next field Y/N :
y
;Convert field to upper case Y/N :
y
;Help message :
Enter Member's Rank or Rate- LCDR, CAPT, ABHCS, BM2, etc.
;Lowest value :

;Highest value :

;Field name :
RCDSFIAG
;Type of field :
CHAR
;Length of field / Display length / Query length :
1
;Is this field in the base table Y/N :
Y
;Is this field part of the primary key Y/N :
N
;Default value :
A
;Page :
1
;Line :
7
;Column :
66
;Prompt :
Status
;Display prompt above field Y/N :
Y
;Allow field to be entered Y/N :
Y
;Allow field to be updated Y/N :
Y
;SQL>

;Is field mandatory Y/N :
Y
;Is field fixed length Y/N :
Y
;Auto jump to next field Y/N :
Y
;Convert field to upper case Y/N :
Y
;Help message :
A- Active, D- Awaiting Deletion
```

```
;Lowest value :
A
;Highest value :
D
;Field name :
STREET
;Type of field :
CHAR
;Length of field / Display length / Query length :
30
;Is this field in the base table Y/N :
Y
;Is this field part of the primary key Y/N :
N
;Default value :

;Page :
1
;Line :
10
;Column :
10
;Prompt :
Street
;Display prompt above field Y/N :
Y
;Allow field to be entered Y/N :
Y
;Allow field to be updated Y/N :
Y
;SQL>

;Is field mandatory Y/N :
Y
;Is field fixed length Y/N :
N
;Auto jump to next field Y/N :
Y
;Convert field to upper case Y/N :
Y
;Help message :
Enter Member's Street Address- This Field is Mandatory!
;Lowest value :

;Highest value :

;Field name :
CITY
;Type of field :
CHAR
;Length of field / Display length / Query length :
18
;Is this field in the base table Y/N :
Y
;Is this field part of the primary key Y/N :
N
;Default value :

;Page :
1
;Line :
13
;Column :
10
;Prompt :
City
;Display prompt above field Y/N :
Y
;Allow field to be entered Y/N :
```

```
Y
;Allow field to be updated Y/N :
Y
;SQL>

;Is field mandatory Y/N :
Y
;Is field fixed length Y/N :
N
;Auto jump to next field Y/N :
Y
;Convert field to upper case Y/N :
Y
;Help message :
Enter Member's City- This field is mandatory!
;Lowest value :

;Highest value :

;Field name :
STATE
;Type of field :
CHAR
;Length of field / Display length / Query length :
2
;Is this field in the base table Y/N :
Y
;Is this field part of the primary key Y/N :
N
;Default value :

;Page :
1
;Line :
13
;Column :
30
;Prompt :
ST
;Display prompt above field Y/N :
Y
;Allow field to be entered Y/N :
Y
;Allow field to be updated Y/N :
Y
;SQL>

;Is field mandatory Y/N :
Y
;Is field fixed length Y/N :
Y
;Auto jump to next field Y/N :
Y
;Convert field to upper case Y/N :
Y
;Help message :
Enter Member's State- This field is mandatory!
;Lowest value :

;Highest value :

;Field name :
ZIP
;Type of field :
NUMBER
;Length of field / Display length / Query length :
9/5
;Is this field in the base table Y/N :
Y
```

```
;Is this field part of the primary key Y/N :
N
;Default value :

;Page :
1
;Line :
13
;Column :
34
;Prompt :
ZIP
;Display prompt above field Y/N :
Y
;Allow field to be entered Y/N :
Y
;Allow field to be updated Y/N :
Y
;SQL>

;Is field mandatory Y/N :
Y
;Is field fixed length Y/N :
Y
;Auto jump to next field Y/N :
Y
;Convert field to upper case Y/N :
N
;Help message :
Enter Member's ZIP Code- This field is Mandatory!
;Lowest value :

;Highest value :

;Field name :
HOMEPHON
;Type of field :
NUMBER
;Length of field / Display length / Query length :
10
;Is this field in the base table Y/N :
Y
;Is this field part of the primary key Y/N :
N
;Default value :

;Page :
1
;Line :
10
;Column :
50
;Prompt :
Home Phone
;Display prompt above field Y/N :
y
;Allow field to be entered Y/N :
y
;Allow field to be updated Y/N :
y
;SQL>

;Is field mandatory Y/N :
n
;Is field fixed length Y/N :
y
;Auto jump to next field Y/N :
y
;Convert field to upper case Y/N :
```

```
n
;Help message :
Enter Member's Home Phone in this Format- 18005551212
;Lowest value :

;Highest value :

;Field name :
BUSPHONE
;Type of field :
NUMBER
;Length of field / Display length / Query length :
15
;Is this field in the base table Y/N :
Y
;Is this field part of the primary key Y/N :
N
;Default value :

;Page :
1
;Line :
13
;Column :
50
;Prompt :
Business Phone
;Display prompt above field Y/N :
y
;Allow field to be entered Y/N :
y
;Allow field to be updated Y/N :
y
;SQL>

;Is field mandatory Y/N :
n
;Is field fixed length Y/N :
n
;Auto jump to next field Y/N :
y
;Convert field to upper case Y/N :
n
;Help message :
Enter Member's Business Phone in this Format-18005551212x2121
;Lowest value :

;Highest value :

;Field name :
RUIC
;Type of field :
NUMBER
;Length of field / Display length / Query length :
5
;Is this field in the base table Y/N :
Y
;Is this field part of the primary key Y/N :
N
;Default value :

;Page :
1
;Line :
16
;Column :
13
;Prompt :
Reserve UIC
```

LIST OF REFERENCES

1. Naval Audit Service, Report #D30030, Development of the Aviation Training Support System, 1980.

2. Functional Description for the Reserve Training Support System (RTSS) for Air, by Chief of Naval Reserve (Code 46), 30 March 1981.

3. Functional Description for the Reserve Training Support System (RTSS) for Training Management, by Chief of Naval Reserve (Code 46), 15 November 1983

4. U.S. Department of the Navy, Deputy Chief of Naval Operations (Manpower, Personnel, and Training), Manpower, Personnel, and Training Information Systems (MAPTIS) Program Strategic Plan for Program Objective Memorandum (POM) 87 OPNAV P160-P1-84, October 1984.

5. Chamberlin, Ron. CNAVRES Code 4632, New Orleans, LA, Phone interview, 2 February 1985.

6. Minutes of RTSS(Air) Program Review and Evaluation Conference, New Orleans, 22-23 May 1984.

7. Minutes of ATSS Configuration Advisory Board Meeting, Norfolk, VA, 11-14 September 1984.

8. National Academy of Sciences, Navy Nontactical Automatic Data Processing Policy, Organization and Management, by Committee on the Review of Navy Long-Range ADP Planning, July 1983.

9. Dammeyer, Rod F., "Developing Information Systems to Meet Top Management's Needs," Management Review, v. 72, Feb 1983.

10. U.S. Department of the Navy, Commander, Naval Reserve Force Analysis of Commander, Naval Reserve Force Information Systems Management Requirements, by Naval Reserve CINCPACFLT Detachment 420, January 1984.

11. U.S. Department of the Navy, Deputy Chief of Naval Operations (Manpower, Personnel, and Training), Manpower, Personnel, and Training Information Systems (MAPTIS) Program End User Computing Guidelines OPNAV P160-G2-84, September 1984.

12. Gore, Marvin and Stubbe, John, Elements of Systems Analysis, Wm. C. Brown Co., 1983.

APPENDIX D

PROTOTYPE INDICES

Index Name	Table Name	Column Name	Index Type
NAME_RUIC	NAME	RUIC	NON UNIQUE
NAME_ZIP	NAME	ZIP	NON UNIQUE
NAME_SSN	NAME	SSN	UNIQUE
SSN_LANG	LANGUAGE	SSN	NON UNIQUE
LANG_SSN	LANGUAGE	LANG	NON UNIQUE
SSN_EDUCATION	EDUCATION	SSN	NON UNIQUE
NAME_TRUIC	NAME	TRUIC	NON UNIQUE
CLR_SSN	CLEARANCE	SSN	NON UNIQUE
DRILL_SSN	DRILLS	SSN	NON UNIQUE
DRILL_NUM	DRILLS	NUMDRILLS	NON UNIQUE

```
;Field name :
;Block name / Description :
          - - - - - - - - PERSONAL INFORMATION - - - - - - - -
%END
```

```
;Line :
21
;Column :
20
;Prompt :
COCI
;Display prompt above field Y/N :
Y
;Allow field to be entered Y/N :
Y
;Allow field to be updated Y/N :
Y
;SQL>

;Is field mandatory Y/N :
N
;Is field fixed length Y/N :
N
;Auto jump to next field Y/N :
Y
;Convert field to upper case Y/N :
N
;Help message :
Enter Member's COCI
;Lowest value :

;Highest value :

;Field name :
SUBCD
;Type of field :
CHAR
;Length of field / Display length / Query length :
5
;Is this field in the base table Y/N :
Y
;Is this field part of the primary key Y/N :
N
;Default value :

;Page :
2
;Line :
21
;Column :
58
;Prompt :
Subspecialty
;Display prompt above field Y/N :
y
;Allow field to be entered Y/N :
y
;Allow field to be updated Y/N :
y
;SQL>

;Is field mandatory Y/N :
n
;Is field fixed length Y/N :
n
;Auto jump to next field Y/N :
y
;Convert field to upper case Y/N :
n
;Help message :
Enter member's Subspecialty Code
;Lowest value :

;Highest value :
```

```
N
;Is field fixed length Y/N :
N
;Auto jump to next field Y/N :
Y
;Convert field to upper case Y/N :
N
;Help message :
Enter Member's Lineal Number
;Lowest value :

;Highest value :

;Field name :
PNOBC_OR_PNEC
;Type of field :
CHAR
;Length of field / Display length / Query length :
4
;Is this field in the base table Y/N :
Y
;Is this field part of the primary key Y/N :
N
;Default value :

;Page :
2
;Line :
18
;Column :
58
;Prompt :
NOBC/PNEC
;Display prompt above field Y/N :
Y
;Allow field to be entered Y/N :
Y
;Allow field to be updated Y/N :
Y
;SQL>

;Is field mandatory Y/N :
N
;Is field fixed length Y/N :
N
;Auto jump to next field Y/N :
Y
;Convert field to upper case Y/N :
N
;Help message :
Enter Member's Primary NOBC or NEC
;Lowest value :

;Highest value :

;Field name :
COCI
;Type of field :
CHAR
;Length of field / Display length / Query length :
9
;Is this field in the base table Y/N :
Y
;Is this field part of the primary key Y/N :
N
;Default value :

;Page :
2
```

```
;Type of field :
NUMBER
;Length of field / Display length / Query length :
6
;Is this field in the base table Y/N :
Y
;Is this field part of the primary key Y/N :
N
;Default value :

;Page :
2
;Line :
15
;Column :
58
;Prompt :
EXRAA
;Display prompt above field Y/N :
Y
;Allow field to be entered Y/N :
Y
;Allow field to be updated Y/N :
Y
;SQL>

;Is field mandatory Y/N :
N
;Is field fixed length Y/N :
Y
;Auto jump to next field Y/N :
Y
;Convert field to upper case Y/N :
N
;Help message :
Enter Member's EXRAA- YYMMDD
;Lowest value :

;Highest value :

;Field name :
LINEAL
;Type of field :
NUMBER
;Length of field / Display length / Query length :
8
;Is this field in the base table Y/N :
Y
;Is this field part of the primary key Y/N :
N
;Default value :

;Page :
2
;Line :
13
;Column :
20
;Prompt :
Lineal No.
;Display prompt above field Y/N :
y
;Allow field to be entered Y/N :
Y
;Allow field to be updated Y/N :
Y
;SQL>

;Is field mandatory Y/N :
```

```
20
;Prompt :
Precedence No.
;Display prompt above field Y/N :
Y
;Allow field to be entered Y/N :
Y
;Allow field to be updated Y/N :
Y
;SQL>

;Is field mandatory Y/N :
N
;Is field fixed length Y/N :
N
;Auto jump to next field Y/N :
Y
;Convert field to upper case Y/N :
N
;Help message :
Enter Member's Precedence Number
;Lowest value :

;Highest value :

;Field name :
MOD
;Type of field :
NUMBER
;Length of field / Display length / Query length :
1
;Is this field in the base table Y/N :
Y
;Is this field part of the primary key Y/N :
N
;Default value :

;Page :
2
;Line :
15
;Column :
40
;Prompt :
MOD
;Display prompt above field Y/N :
Y
;Allow field to be entered Y/N :
Y
;Allow field to be updated Y/N :
Y
;SQL>

;Is field mandatory Y/N :
N
;Is field fixed length Y/N :
Y
;Auto jump to next field Y/N :
Y
;Convert field to upper case Y/N :
N
;Help message :
Enter Member's MOD
;Lowest value :

;Highest value :

;Field name :
EXRAA
```

```
;Auto jump to next field Y/N :
Y
;Convert field to upper case Y/N :
N
;Help message :
Enter Member's Date of Rank or Rate- YYMMDD
;Lowest value :

;Highest value :

;Field name :
EOS
;Type of field :
NUMBER
;Length of field / Display length / Query length :
6
;Is this field in the base table Y/N :
Y
;Is this field part of the primary key Y/N :
N
;Default value :

;Page :
2
;Line :
12
;Column :
58
;Prompt :
EOS
;Display prompt above field Y/N :
Y
;Allow field to be entered Y/N :
Y
;Allow field to be updated Y/N :
Y
;SQL>

;Is field mandatory Y/N :
N
;Is field fixed length Y/N :
Y
;Auto jump to next field Y/N :
Y
;Convert field to upper case Y/N :
N
;Help message :
Enter Member's Expiration Of Service
;Lowest value :

;Highest value :

;Field name :
PRECD
;Type of field :
NUMBER
;Length of field / Display length / Query length :
8
;Is this field in the base table Y/N :
Y
;Is this field part of the primary key Y/N :
N
;Default value :

;Page :
2
;Line :
15
;Column :
```

```
1
;Is this field in the base table Y/N :
Y
;Is this field part of the primary key Y/N :
N
;Default value :

;Page :
2
;Line :
12
;Column :
20
;Prompt :
Pay Grade
;Display prompt above field Y/N :
Y
;Allow field to be entered Y/N :
Y
;Allow field to be updated Y/N :
Y
;SQL>

;Is field mandatory Y/N :
N
;Is field fixed length Y/N :
Y
;Auto jump to next field Y/N :
Y
;Convert field to upper case Y/N :
N
;Help message :
Enter Member's Pay Grade from Appendix
;Lowest value :

;Highest value :

;Field name :
DOR
;Type of field :
NUMBER
;Length of field / Display length / Query length :
6
;Is this field in the base table Y/N :
Y
;Is this field part of the primary key Y/N :
N
;Default value :

;Page :
2
;Line :
12
;Column :
40
;Prompt :
Date of Rank
;Display prompt above field Y/N :
Y
;Allow field to be entered Y/N :
Y
;Allow field to be updated Y/N :
Y
;SQL>

;Is field mandatory Y/N :
N
;Is field fixed length Y/N :
Y
```

```
;Display prompt above field Y/N :
Y
;Allow field to be entered Y/N :
Y
;Allow field to be updated Y/N :
Y
;SQL>

;Is field mandatory Y/N :
N
;Is field fixed length Y/N :
N
;Auto jump to next field Y/N :
Y
;Convert field to upper case Y/N :
N
;Help message :
Enter Member's Dependency Code
;Lowest value :

;Highest value :

;Field name :
CLRACCESS
;Type of field :
CHAR
;Length of field / Display length / Query length :
6
;Is this field in the base table Y/N :
Y
;Is this field part of the primary key Y/N :
N
;Default value :

;Page :
2
;Line :
9
;Column :
58
;Prompt :
Clearance
;Display prompt above field Y/N :
Y
;Allow field to be entered Y/N :
Y
;Allow field to be updated Y/N :
Y
;SQL>

;Is field mandatory Y/N :
N
;Is field fixed length Y/N :
Y
;Auto jump to next field Y/N :
Y
;Convert field to upper case Y/N :
Y
;Help message :
UNCLAS,CONFID,SECRET,TOPSEC
;Lowest value :

;Highest value :

;Field name :
PAYGRADE
;Type of field :
CHAR
;Length of field / Display length / Query length :
```

```
Y
;Help message :
C- Caucasian, B- Black, O- Oriental, H- Hispanic
;Lowest value :
B
;Highest value :
O
;Field name :
PEBD
;Type of field :
NUMBER
;Length of field / Display length / Query length :
6
;Is this field in the base table Y/N :
Y
;Is this field part of the primary key Y/N :
N
;Default value :

;Page :
2
;Line :
9
;Column :
20
;Prompt :
PEBD
;Display prompt above field Y/N :
Y
;Allow field to be entered Y/N :
Y
;Allow field to be updated Y/N :
Y
;SQL>

;Is field mandatory Y/N :
N
;Is field fixed length Y/N :
Y
;Auto jump to next field Y/N :
Y
;Convert field to upper case Y/N :
N
;Help message :
Enter Member's Pay Entry Base Date
;Lowest value :

;Highest value :

;Field name :
DEPEN
;Type of field :
CHAR
;Length of field / Display length / Query length :
2
;Is this field in the base table Y/N :
Y
;Is this field part of the primary key Y/N :
N
;Default value :

;Page :
2
;Line :
9
;Column :
40
;Prompt :
Dep Code
```

```
;Is this field part of the primary key Y/N :
N
;Default value :

;Page :
2
;Line :
6
;Column :
40
;Prompt :
Sex
;Display prompt above field Y/N :
Y
;Allow field to be entered Y/N :
Y
;Allow field to be updated Y/N :
Y
;SQL>

;Is field mandatory Y/N :
N
;Is field fixed length Y/N :
Y
;Auto jump to next field Y/N :
Y
;Convert field to upper case Y/N :
Y
;Help message :
M- Male  F-Female
;Lowest value :
F
;Highest value :
M
;Field name :
RACE
;Type of field :
CHAR
;Length of field / Display length / Query length :
1
;Is this field in the base table Y/N :
Y
;Is this field part of the primary key Y/N :
N
;Default value :

;Page :
2
;Line :
6
;Column :
60
;Prompt :
Race
;Display prompt above field Y/N :
Y
;Allow field to be entered Y/N :
Y
;Allow field to be updated Y/N :
Y
;SQL>

;Is field mandatory Y/N :
N
;Is field fixed length Y/N :
Y
;Auto jump to next field Y/N :
Y
;Convert field to upper case Y/N :
```

```
9/9/0
;Is this field in the base table Y/N :
N
;Default value :

;Page :
2
;Line :
3
;Column :
36
;Prompt :
Soc. Sec. No.
;Display prompt above field Y/N :
n
;Allow field to be entered Y/N :
N
;SQL>

;Field name :
DOB
;Type of field :
NUMBER
;Length of field / Display length / Query length :
6
;Is this field in the base table Y/N :
Y
;Is this field part of the primary key Y/N :
N
;Default value :

;Page :
2
;Line :
6
;Column :
20
;Prompt :
Birth Date
;Display prompt above field Y/N :
Y
;Allow field to be entered Y/N :
Y
;Allow field to be updated Y/N :
Y
;SQL>

;Is field mandatory Y/N :
N
;Is field fixed length Y/N :
Y
;Auto jump to next field Y/N :
Y
;Convert field to upper case Y/N :
N
;Help message :
Enter Member's Birth Date in this Format- YYMMDD
;Lowest value :

;Highest value :

;Field name :
SEX
;Type of field :
CHAR
;Length of field / Display length / Query length :
1
;Is this field in the base table Y/N :
Y
```

```
;Display prompt above field Y/N :
Y
;Allow field to be entered Y/N :
Y
;Allow field to be updated Y/N :
Y
;SQL>

;Is field mandatory Y/N :
N
;Is field fixed length Y/N :
Y
;Auto jump to next field Y/N :
Y
;Convert field to upper case Y/N :
N
;Help message :
Enter Member's Reserve Unit Identification Code
;Lowest value :

;Highest value :

;Field name :
TRUIC
;Type of field :
NUMBER
;Length of field / Display length / Query length :
5
;Is this field in the base table Y/N :
Y
;Is this field part of the primary key Y/N :
N
;Default value :

;Page :
1
;Line :
16
;Column :
54
;Prompt :
Training UIC
;Display prompt above field Y/N :
Y
;Allow field to be entered Y/N :
Y
;Allow field to be updated Y/N :
Y
;SQL>

;Is field mandatory Y/N :
N
;Is field fixed length Y/N :
Y
;Auto jump to next field Y/N :
Y
;Convert field to upper case Y/N :
N
;Help message :
Enter Member's Training RUIC if it is Different from RUIC
;Lowest value :

;Highest value :

;Field name :
SSN
;Type of field :
NUMBER
;Length of field / Display length / Query length :
```

13. "Tools to Rejuvenate Your Old Systems," EDP Analyzer, v. 22, April 1984.

14. Page, Marcus W., "Crossing the Data Bridge", Government Computing News, December 1984.

15. "A Simple Guide To Five Normal Forms In Relational Database Theory," Computing Practices, v. 26, February 1983.

16. Kroenke, David, Database Processing, Science Research Associates, Inc., 1983.

17. Weiss, Harvey M., "The ORACLE Database Management System," Mini-Micro Systems, v. 13, August 1980.

18. Chamberlain, D. D., et. al., "SEQUEL 2: A Unified Approach to Data Definition, Manipulation, and Control," IBM Journal of Research and Development, November 1976.

19. Relational Software, Inc., ORACLE Terminal User's Guide, version 3.1, 10 March 1983.

20. Senn, James A., Information Systems In Management, Wadsworth Publishing Co., 1982.

21. ORACLE, Inc., ORACLE UFI Reference Manual, version 4.0, August 1984.

22. ORACLE, Inc., ORACLE Database Administrator's Guide, version 4.0, August 1984.

23. McKim, Commander Gerald. CNAVRES Code 45, New Orleans, LA. Interview, 16 August 1984.

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.	LCDR Michael Winslow VAW-120 Norfolk, Virginia 23511	2
4.	LT Howard Seeger Director, Strategic System Project Office Washington, D. C. 20376	2
5.	Department Chairman, Code 54 Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943	1
6.	CDR Ronald Kurth, Code 52KH Department of Computer Science Naval Postgraduate School Monterey, California 93943	3
7.	Adjunct Professor Michael Spencer, Code 54XQ Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943	3
8.	Associate Professor Thomas Swenson, Code 54ZW Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943	2
9.	Computer Technology Curriculum Office Code 37 Naval Postgraduate School Monterey, California 93943	1
10.	Commander Naval Reserve Forces Code 46 New Orleans, Louisiana 70146	2

END

FILMED

8-85

DTIC