NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

RE-ENGINEERING THE UNITED STATES MARINE CORPS' ENLISTED ASSIGNMENT MODEL (EAM)

by

Gary D. Koch Jr.

June 1998

Thesis Advisors: Hemant K. Bhargava
Suresh Sridhar

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE June 1998	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE : RE-ENGINEERING THE UNITED STATES MARINE CORPS' ENLISTED ASSIGNMENT MODEL (EAM)

5. FUNDING NUMBERS

6. AUTHOR(S)
Gary D. Koch Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

In a time of downsizing and budgetary constraints the Manpower division of Headquarters, United States Marine Corps, accomplishes its mission "to put the right Marine in the right place at the right time with the right skills and quality of life" in a variety of ways. Currently, one of the processes that assist the Marine Enlisted Assignments branch is the Enlisted Assignment Model. The current system is not producing the results that are needed and the current managers do not trust the output. This thesis proposes changes to the EAM user interface, data access, and data storage capabilities to enable the Marine Corps to use the latest information technology to more closely mirror the vision as stated above. With the use of Business Process Reengineering, Process Modeling, and Database Design a prototype is developed to address areas of the current system that can be changed. By using these methods to ensure an appropriate interface with optimization techniques, a complete Decision Support System for manpower assignments can be realized. These changes will empower managers to effectively and efficiently manage, not just monitor manpower readiness in order to meet the challenges of the 21st century.

14. SUBJECT TERMS
USMC, Databases, Manpower Assignment, Models, Decision Support Systems, Graphical User Interface

15. NUMBER OF PAGES
149

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFI- CATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

Approved for public release; distribution is unlimited

RE-ENGINEERING THE UNITED STATES MARINE CORPS' ENLISTED ASSIGNMENT MODEL (EAM)

Gary D. Koch Jr.
Major, United States Marine Corps
B.S., The Citadel, 1986

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
June 1998

Author: _____
Gary D. Koch Jr.

Approved by: _____
Hemant Bhargava, Thesis Advisor

_____
Suresh Sridhar, Thesis Advisor

_____
Reuben T. Harris, Chairman
Department of Systems Management

ABSTRACT

In a time of downsizing and budgetary constraints the Manpower division of Headquarters, United States Marine Corps, accomplishes its mission "to put the right Marine in the right place at the right time with the right skills and quality of life" in a variety of ways. Currently, one of the processes that assist the Marine Enlisted Assignments branch is the Enlisted Assignment Model. The current system is not producing the results that are needed and the current managers do not trust the output. This thesis proposes changes to the EAM user interface, data access, and data storage capabilities to enable the Marine Corps to use the latest information technology to more closely mirror the vision as stated above. With the use of Business Process Reengineering, Process Modeling, and Database Design a prototype is developed to address areas of the current system that can be changed. By using these methods to ensure an appropriate interface with optimization techniques, a complete Decision Support System for manpower assignments can be realized. These changes will empower managers to effectively and efficiently manage, not just monitor manpower readiness in order to meet the challenges of the 21st century.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I wish to thank all those that made this thesis possible. Professor Hemant Bhargava, Professor Suresh Sridhar, Kevin Snoap, and Brian Tivnan for their continued input, help, and patience in this process. Without their abundant knowledge and vast research in this subject area there wouldn't be a written form of this thesis, nor would I have met the requirements to graduate.

I stand in awe of the folks at Decision Support Associates, Inc. (DSAI) who have tirelessly devoted a major portion of their lives to EAM. This work pales in comparison to the many years they have spent struggling with this very complex problem, and the ingenious methods with which they have attempted to solve it.

I thank my wife, Carolyn, whom I love more than words can express. She has been the true champion throughout this intensive learning process. Training our six children "in the way they should go" and still managing to "excel them all" and give words of advise and encouragement to me. "Charm is deceitful and beauty is passing, but a woman who fears the Lord, she shall be praised."

Finally I give all of the glory, honor and praise to my Heavenly Father who gave His Son, Jesus Christ to pay the price for my redemption from sin. I live to serve Him.

Coram Deo

Maranantha!

I. INTRODUCTION

In a time of downsizing and budgetary constraints the Manpower division of Headquarters, United States Marine Corps, accomplishes its mission: "to put the right Marine in the right place at the right time with the right skills and quality of life"[Christmas 1996] in a variety of ways. Currently, one of the processes that assist the Marine Enlisted Assignments branch is the Enlisted Assignment Model (EAM). The use of models such as EAM and the understanding of their use are going to become a critical issue in the coming years as we move further into the information age. The current model is complicated and complex and often produces results that are not consistent with current manager projections. This leads to a lack of trust. As a result of this perception the assignments process has been reduced to a time intensive manual process: enlisted monitors, senior non-commissioned officers operating in their respective military occupational specialties, sort through several data fields to match individual Marines to fluid monthly requirements.

The purpose of this thesis is to propose changes to the EAM user interface, data access, and data storage design, by examining the current process of assigning enlisted Marines. By enabling the Marine Corps to use the latest information technology to more closely mirror the vision as stated above, these changes will empower managers to effectively and efficiently manage manpower readiness with greater flexibility to meet the challenges of the 21st century.

A. BACKGROUND

The Manpower and Reserve Affairs (M&RA) Department, Headquarters USMC, is concerned with development, upgrade and maintenance of the M&RA Human Resource Development Process Information Management Systems (HRPDIMS). The Total Force Management Modernization project supports this systems modernization process.

One of the long-term objectives of the Total Force Management Modernization project is to reengineer and re-implement the various manpower planning models using modern computer based modeling technology. This will be done in such a way that allows better interoperability and model reuse, better data and model management, and an improved user and control interface. The current models and algorithms were developed in the 1970s and are owned by Decision Support Applications, Inc. (DSAI).

B. PURPOSE

The current modeling environment for the EAM requires significant contractor involvement for execution, maintenance, and any improvements reflecting changes in policy or new requirements. Therefore, the purpose of this thesis is to reengineer this model. This reengineering involves:

1) a careful study and documentation of the current business process and logic of making enlisted assignment decisions,

2) a proposal of innovations of improvements to the current process in a way that better achieves organizational goals as well as those of the individuals affected by the decisions,

3) demonstration of models and algorithms in a functional prototype Decision Support System (DSS) that includes parts for the model, the data, and user dialog management.

C. SCOPE AND METHODOLOGY

1. Scope

The scope includes:

a) analysis and documentation of the current process and logic of the EAM,

b) reengineer the EAM to facilitate better decisions,

c) assist in the implementation of concepts introduced from the prototype as required by the user.

2. Methodology

The methodology used in this research consists of the following actions:

a) Review relevant information resources including: Pertinent USMC Personnel Orders, EAM documentation, and Standard Operating Procedures (SOP)

b) Conduct an in-depth review of business process reengineering

c) Examine goals of EAM, performance measures, current procedures and perceptions

d) Model existing system and examine activities

e) Design a 'mock' prototype that will be able to incorporate the optimizing mathematical models developed by Capt. Brian Tivnan, Operational Research Student. [Tivnan 1998]

D. ORGANIZATION OF STUDY

Chapter II presents the analysis of the current system as it exists today and as understood by the current users of the system. A critique of each functional area will be included in this chapter. Chapter III discusses the user interface specifically as well as a tour through the current 'mock' prototype. Chapter IV presents an overview of the model, data and design concepts that led to the development of the 'mock' prototype. Chapter V presents the conclusions of this research along with other recommendations for the implementation of the changes that are suggested in this work. Further recommendations for research are also categorized.

II. ANALYSIS OF THE CURRENT SYSTEM

In order to understand the current system the author used techniques and methods from Business Process Reengineering (BPR). Business Process Reengineering is the organizational process required aligning people, processes and technology with strategies to achieve business integration. It can also be thought of as taking a business in its current state and forming an organizational and operational blueprint to redirect skills, policies, information (data), cultural values, organizational structure, processes and incentives towards targeting improvements. [Bui 1996]

A. CURRENT OPERATION AND SYSTEM DESCRIPTION

The Enlisted Assignment Model (EAM) is a large-scale computer-based system designed to make worldwide Marine Corps by-name assignments. It is a flexible system in that it allows the EAM model manager to control all stages of the assignment process through table inputs and/or changes. These inputs and changes are made primarily through the EAM "dictionary", which is a collection of tables containing rules, properties, and other information relevant to the assignment processing. The term dictionary is used throughout this thesis to refer to these tables.

EAM assignments are made monthly through two separate assignment executions or runs. The first run, referred to as the OVERSEAS run, is normally executed at the beginning of the month. The second run, or Continental United States (CONUS) run, is executed during the middle of the month. The model is also capable of making

recommendations for CONUS to CONUS assignments, as well as limited overseas to overseas movement. The timing of the runs can be changed if the model manager determines such a need, as long as the runs are executed with respect to a six-month projection window and keeps in-line with the execution of the Enlisted Staffing Goal Model (ESGM).

1. System Description

The author has used the IDEF 0 modeling approach in Appendix A to formalize the following discussion. Figures 1 and 2 are sections taken from Appendix A created with BPWIN 2.0 and inserted in this section to help understand the following discussion (see Figure 1).

The EAM consists of five primary functions or overlays. The first overlay processes the EAM Personnel File. It determines the "draw" (receiving) Monitor Command Code (MCC) billet on board counts and any special requirements, and selects Marines who are available for reassignment. The second primary overlay uses draw MCC staffing goal data from the first primary overlay, the staffing goals file (taken from the ESGM), and optional user input to generate quotas. The third and fourth primary overlays operate together to assign available Marines to quotas. The fifth primary overlay provides reports, updates files required to process EAM assignments, and is capable of deriving advanced assignments.

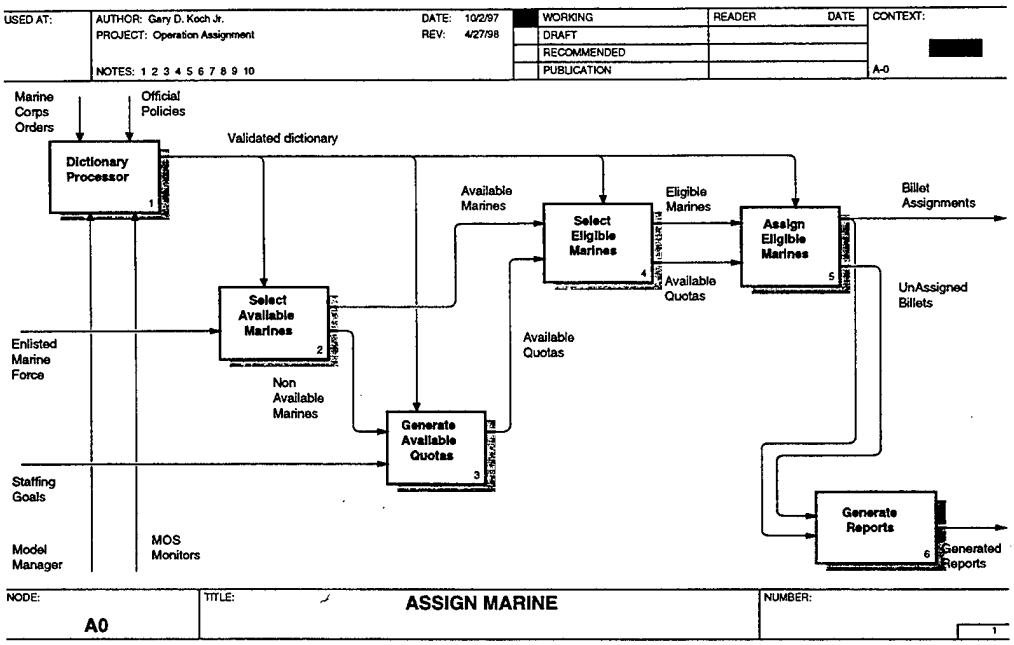


Figure 1: Main Overlays of EAM

2. Operation

Each assignment run proceeds through the following steps (refer to Figure 1):

1) Validate the dictionary

2) Selection of personnel available for reassignment

3) Accumulation of counts of personnel to be charged on-board

4) Generation of quotas by MCC

5) Assignment of personnel to MCC

6) Generation of assignment reports

The first step performed by the model is the selection of available "movers." This is done through a series of true/false tests against Marine Corps Total Forces System

(MCTFS) and Automated Orders Writing Process database information, determining if the individual Marine meets the rules and requirements for assignment. The model manager devises these tests based upon properties derived from the above-mentioned data.

The accumulation of on-board counts is based upon a projection of the status of each Marine and each command at the "fill" time of the run. If the Marine is leaving the command before the fill month he is excluded; likewise, if he is arriving during the fill month he is added and considered "chargeable." It should be noted here that EAM makes recommendations for *future* assignments, but receives and uses personnel information in the form of *current* MCTFS data.

In the quota generation step EAM develops quotas based upon the "picture" of the future status of the commands to be filled. The staffing goal taken from ESGM is the "target." In short, the future on-board population is compared against the staffing goal and the difference between the two becomes the quotas. The sole consideration at this point is the fill requirement, and EAM generates quotas for each command, including those to which it is not allowed to make recommended assignments. For those particular commands, the EAM generated quota become a tool for indicating the number of quotas needed to keep that command at its current staffing goal level (see Figure 2).

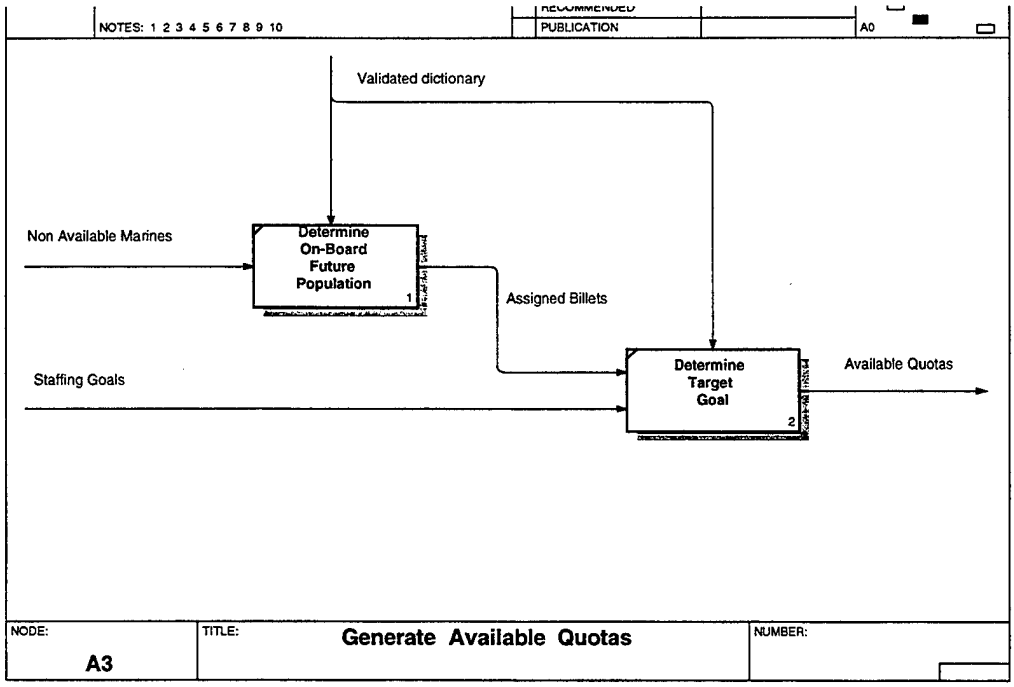


Figure 2: Generating Available Quotas

At this point in the process, the requirements, which are the quotas, have been determined, and there is an available pool of Marines. The EAM will now make assignment recommendations according to the eligibility rules for each quota as defined by mandatory properties in the 'Property Optimization' table of the dictionary. After the model has filled as many of the quotas as possible without violating mandatory rules, it then begins an optimization process. The recommendations are rearranged so as to achieve the most "desirable" set of assignments. These desirable requirements are relative to preferences in several possible categories such as MOS/Grade Substitution Level Minimization, Tour Category Sequence, MCC Preference, and/or Rotation Order.

Along with the orders recommendations, EAM also produces reports displaying relevant assignment information. This information includes the following: quotas,

assignments, summaries, and unassigned Marines who are also available. The model manager and the monitors use these reports.

B. MODEL COMPONENTS DICTIONARY

The following information that is presented concerns the EAM dictionary. This is at the heart of the model. The following narrative will describe briefly the interaction with the Model Manager and the dictionary through a series of tables. The reader can get a more detailed description of the tables and their structure and how the model manager interacts with specific rules and properties in Appendix B.

The Dictionary is called and read into the EAM in the first overlay. There is a specific sequence that the Processor follows in its operation. The sequence of operations during an EAM run is as follows:

1) MCC Definition table read to determine valid MCCs
2) MOS Substitution table read to determine valid MOS/Grade combinations
3) On-board and Available Marines identified
4) Staffing goal file read to determine quotas
5) Properties Optimization table read to determine properties for each MCC
6) From mandatory properties, Marines are assigned to MCCs per MOS Family
7) Assignments optimized from desirable properties
8) Remaining optimizations are determined by model manager

C. LIST OF INPUTS AND OUTPUTS OF THE EAM

Each of the two assignments runs produce a number of files that are either used by the monitors and orders manager, or can be used for analyzing a run. The following is a listing of the input/output files and their extensions [USMC 1992]:

1. Input Files

- MCTFS File (mmsout. txt)
- ESGM File (. eam)

2. Overseas and Conus Output Files

- Quotas File (. qts)
- Quotas Report (. qrp)
- On-Boards File (. obs - sorted or .obu - unsorted)
- Orders File* (. ord - sorted, or .oru - unsorted)
- Assignment Summary Report (. asu)
- Special Billets Report (. sbs)
- Unassigned Available File* (. uau)

3. Post EAM Execution Reports

- Monitor Worksheets
- Solutions Reports (identifies monitor decisions on each recommendation)

One of the jobs run by the programmers after an EAM run produces EAM worksheets for the monitors. The monitors use the sheets when processing the recommendations. When the programmer receives the output from this job, he gives

them to the orders manager. The orders manager will then distribute the worksheets to the monitors for their use. These worksheets are used in conjunction with the EAM recommendations processing. The orders manager downloads the recommendations to the LAN for the monitors to process.

D. SPECIFICATION OF AREAS OF CONCERN, CONSTRAINTS AND ASSUMPTIONS

The operation of the EAM is a complex process. The complexity evolves around the fact that the EAM attempts to accurately model the complicated and demanding process of assigning enlisted Marines worldwide. The underlying fact is that there are many cases that do not fit neatly into all the rules, policies, requirements, and regulations, while still meeting the desires of the individual Marines. Most, if not all, assignments involve positive and negative tradeoffs.

The EAM can be construed as an overly complicated system run by a big computer. This perception can cause managers to shun the model. The key to getting good results from the current system is to understand some critical concepts and to be made fully aware of certain constraints and assumptions. Some of these areas are covered in the following paragraphs. These areas are examined in order to complete our understanding of the current EAM system and are not necessarily areas that need to be 'fixed'.

1. The Target

The first critical concept is the "target" used by the model. The EAM is purely an assignment model and therefore uses staffing goals as the target, not the Table of Organization (T/O) or the Authorized Strength Report (ASR). The resolution of how many Marines by grade and skill are to be assigned to a command is represented by the staffing goal. It is important that appropriate interaction occur with the Staffing Goal Model Manager to ensure that the EAM is targeting correct staffing goals. However, it is even more important to note that even given perfect staffing goals, there are still not enough movable Marines in some grades and skills to fill all projected targets.

2. Accept/Reject Philosophy

The overall goal of the EAM is to make acceptable assignment recommendations to the monitors. An "acceptable" recommendation is "accepted", "modified" or "rejected" by the monitor processing it. The average overall acceptance (accept/modify) rate for an overseas assignment run is only 10%-25%. The average for a CONUS run is 75%-85%. These are "skewed" acceptance numbers. For example, if a small (total population) MOS received only two recommendations and the monitor rejects them both, the zero accept/modify percentage that is reflected is less significant. There are also cases where an individual monitor will become too busy to process his EAM recommendations, or will not be able to process for some reason. He might then reject all of the recommendations, or let them revert to a pending status. [USMC 1992]

3. EAM Careerist Definition

The EAM is capable of making worldwide assignment recommendations for all enlisted Marines. However, some populations are much harder to manage than others are. The "careerist" definition concept is a major factor in determining what individual Marine gets what recommendation for what assignment. This definition is not the same as the Department of Defense careerist definition, nor the same as the career planner careerist definition. The definition is an attempt to measure those Marines likely to be in the Marine Corps past their EAS currently shown in their MCTFS record. In order to project the population at a command the model must "know" which Marines are exiting the Marine Corps, and therefore leaving that unit - a difficult task due to the six-month target approach. "First-term" Marines generally leave the Marine Corps at their EAS; however, this is not true in all cases, leading to erroneous assessments. The current EAM careerist definition serves as a "least common denominator" - the majority of the Marines recommended orders will carry them out. Any Marine who meets this definition will be assigned regardless of his EAS. The ability to establish a minimum time left to EAS upon execution of Permanent Change of Station (PCS) orders exists in the EAM. This capability is found in the property definitions of eligibility or specific billets or commands.

4. Selected Grade

The EAM recommends assignments to fill targets six months into the future. Most of these assignments are for a period of at least three years. Therefore, selection or promotion to the next higher grade is a necessary consideration when selecting Marines to move.

5. MOS and Grade Substitution

In many cases there are no available or eligible Marines to move into vacant staffing goals. There will also be Marines on board that do not match a staffing goal by grade and by MOS. The EAM uses the concept of MOS and grade substitution to solve these problems. The MOS/Grade Substitution Deck in the EAM Dictionary allows the model manager to define acceptable MOS and grade substitutions as a set of levels in the Levels Deck. On grade, on MOS is obviously the most desirable assignment, and is therefore a Level 1 fill. Perhaps the next best Marine for the assignment is one with the same MOS but one grade senior - thus the Marine is on MOS and up one grade and becomes Level 2. The monitor of each respective population provides the needed input for this part of the dictionary.

6. MOS Families

Assessing the assignment process for the entire enlisted population is extremely difficult due to the numbers and size involved. Once it is determined that the staffing

goal process has properly defined the target, or the problems associated with the staffing goal have been noted, it is then acceptable to break up the total assignment task into various smaller parts that are totally unrelated. This is basically the "families" concept. MOS's are grouped into families and then the rules and the processing applies to one family at a time. It should be noted that grade and MOS substitution must stay within families; there does not have to be any MOS substitution within a family.

7. MCTFS Record Input to EAM

Not all Marines are actually processed by the EAM. The very first screen or overlay of the model checks for a valid MCC and MOS in the extract record. These edits are against a table in the EAM dictionary, and any record not passing the edits is immediately removed from further consideration. It is the responsibility of the model manager to ensure that the MOS and MCC tables used in the EAM are up-to-date and reflect the data in the MCTFS tables.

8. Projected On-board Chargeable

The EAM assumes that staffing goals are appropriate for the projected time frame. In order to project the picture of a command in question the model uses the current MCTFS picture of the personnel inventory, and adds or subtracts various kinds of Marines (on out-bound orders from command, on orders in-bound to command). This process results in a total projected on-board count by grade and MOS. This count is then compared to the staffing goals before the actual process of recommendation starts. The

results of this projection effort are displayed in the Quotas Report. The difference between this projected population and the staffing goals causes the EAM to search for "fill" into the remaining openings or "holes". Those Marines currently on board at a command are projected to complete their tours using a calculated date called the "tour completion date". In some cases the date will simply be the rotational tour date (RTD).

9. Availability

The EAM designates every Marine as either available or non-available to leave his present command during the draw month. This characteristic is defined in the dictionary. It includes such things as minimum time on station. In some cases a Marine may be movable, but is not the best choice for assignment. EAM computes availability based on MCTFS data such as RTD, or date current tour began (DCTB) and the minimum tour as defined by the EAM dictionary.

10. Eligibility

The EAM defines prerequisites for each billet or set of billets. These definitions include such things as time back from an unaccompanied tour, and time remaining to EAS. The definitions are always applied to billets. The EAM then searches for Marines who meet the required criteria. In general, there are usually more Marines available to move than are available and eligible for a given billet. It should be noted that the EAM generates a list of Marines that are available but unassigned. A recommended Marine

meets the eligibility requirements according to the EAM dictionary and his MCTFS record, but an assigned available Marine meets only some of the billet requirements.

11. Retrieving MCTFS Data and Reviewing EAM Records

If a problem arises in the models execution, and questions prevail as to what the EAM is reading from the individual records in the MCTFS extract, the output files can be analyzed to determine why a Marine was not chosen for assignment.

12. Draw Case Codes

Draw Case Codes (DCC's) are electronic "flags" that indicate a special circumstance with a Marine's record. This special circumstance would not be visible or found in the MCTFS. An example would be the case of a Marine being passed over for promotion. Each Marine can have up to three DCC's. A particular DCC may affect outcomes for assignment purposes according to relative property definitions in the dictionary.

E. CRITIQUE OF THE CURRENT SYSTEM AND RECOMMENDATIONS

The EAM system that is currently running requires constant intervention by the contractor due to a limited understanding of the model by the model manager. As has been highlighted above, the system requires a great deal of time and effort to thoroughly understand its behavior. The present job of the model manager is occupied with many

other requirements so the time to learn this very convoluted and extremely complicated system is limited.

By examining the process of assignments and how the EAM is supposed to expedite this, one can identify areas that could be improved and make the system more understandable and less complex.

The EAM's primary use is as a support tool for the monitors, to help them make timely and cost effective assignments. The intended goal of the EAM system is that the recommendations produced by each assignment run become PCS orders for individual Marines. In order to maximize the potential for this occurrence, interaction with the monitors must transpire. It is the responsibility of the model manager to be proactive in seeking this interaction. Input from the monitors can greatly enhance the quality of the assignment recommendations. Areas of this system that can be improved to allow greater understanding and as such greater involvement in the assignment process are listed below.

1. User Interface to the System

Although the operation of the model is complex, the monitors should not be concerned with this fact. As the primary users the monitors must understand that their input is crucial. They must also understand that they need only articulate their needs and desires in general terms in attempt to better the recommended assignments for their MOS populations. If the models interface is inadequate, then the model manager cannot run it

effectively using these inputs. Because of this faulty input, the output is not going to be useable and the monitors will not waste their time looking at it.

The process can be less intimidating if the user interface is more friendly and self-explanatory. Current access to properties is through two tables in a datasheet view. There are neither Help menus nor ToolTips to assist the model manager. If the model manager is to 'sell' the system, he must be confident in the system and be able to adequately explain it to the monitors. An example of these interface functions will be implemented in the prototype.

2. Improving Access to Full System Capabilities

The monitors must be made aware of the capabilities of the EAM. It is the responsibility of the model manager to promote these capabilities. Knowledge of the system will allow the monitors the ability, and much more importantly, the inclination to provide effective and concise input. The current model does not have a friendly user interface as mentioned earlier.

Currently the system capabilities are underutilized because of the complexity of the interface and lack of immediate access to useful data. The capabilities of the system are hidden from the user and as a result many of the intermediate steps that might be of interest to manpower planners go unused. Currently the reports that EAM could generate go unused, due to the fact that they are only accessible after a complete run. By creating a easily accessible menu or switch board driven interface the manager could easily access

any information that the monitors may need to know about their family or Occupational Specialty.

By redesigning the system database, these results could be stored in a table that is accessible by both the model manager and the monitors. If this is implemented the monitors can actually view the outputs from their own systems without having to go through volumes of paper.

3. Monitor Productivity and Ability

In some instances the productivity and the ability of the individual assignment monitors will counteract the use of the EAM. A number of the monitors will "work ahead" of the model, assigning Marines in their populations prior to an EAM run. The resulting perception is that they do not "need" the recommendations from the model. This is especially true for those monitors that have small populations. The reliance on EAM therefore is decreased; at which time the model recommendations should be promoted as a projection tool for future assignments. By having the results of the model runs accessible to the monitors as soon as possible, and the ability to make test runs at any time, the monitors can use the system even if they "work ahead".

4. Limiting the Property Tables

As discussed in Appendix B, properties are created and strung together to achieve some purpose in the assignment process. Currently there are over 1000 properties. There is a comment field included with each property but it is not used. As a result, neither the

model manager, nor anyone else understands many of the properties. By allowing Derived Properties to be mixed with other Derived Properties without appropriate comments adds to the confusion and lack of understanding.

In the prototype only Fundamental Properties will be used to build Derived or Logical Properties. The user should be required to make an entry in the comment field to explain the purpose of the property. One further check and balance would be relating the property to a known Marine Corps Order (MCO) or policy to provide an audit trail for inspection purposes.

5. MOS Families

Seventy-three MOS Families currently exist in the EAM. Some of these families are counter-intuitive adding complexity to the model. If the current MOS families could be reduced to the Marine Corps OCC Fields, that number could be reduced to fifty-five.

6. Database Design Issues

An ad hoc database is designed and implemented without the benefit of a conceptual data model. These types of databases can lend themselves to being non-relational. While EAM is built on an Oracle DBMS it was not necessarily designed with the underlying theory and principles of a relational database design. The prototype will take a subset of the current EAM system and show the benefits of this theory. By formally developing the database from a conceptual data model one can prevent

undesirable consequences such as modification anomalies that include deletion and insertion anomalies.

7. Fit vs. Fill

While the current EAM system uses the fundamental properties, derived properties, and properties for optimization as defined by the model manager, it is still basically performing a 'fill' operation. It is not truly optimizing and finding the 'best' Marine for a Job. A true DSS should provide that flexibility. This is being addressed in a separate effort. [Tivnan 1998]

III. DEVELOPMENT OF THE USER INTERFACE

This chapter describes the user interface and how it is used to assist the model manager in his ability to assign Marines efficiently and evaluate his assignments and policies more effectively. By establishing a clear view of the way in which the user interacts with the system, a better understanding of the assignment process will be achieved which will enable the model manager to better explain his model to the assignment monitors and give him a clearer understanding of his results. Because of these areas of improvement the model becomes a better decision support system that has the capability to save the Marine Corps time and money. By designing a simple, logical and streamlined interface the model manager can concentrate his efforts on analyzing his results and not figuring out what type of data entry and manipulation is required.

The discussion of the user interface begins by describing the state transition diagram (see Figure 3) as seen from the users perspective and concludes with a tour of the system from the view of the model manager preparing to make an EAM monthly run.

A. STATE TRANSITION DIAGRAM

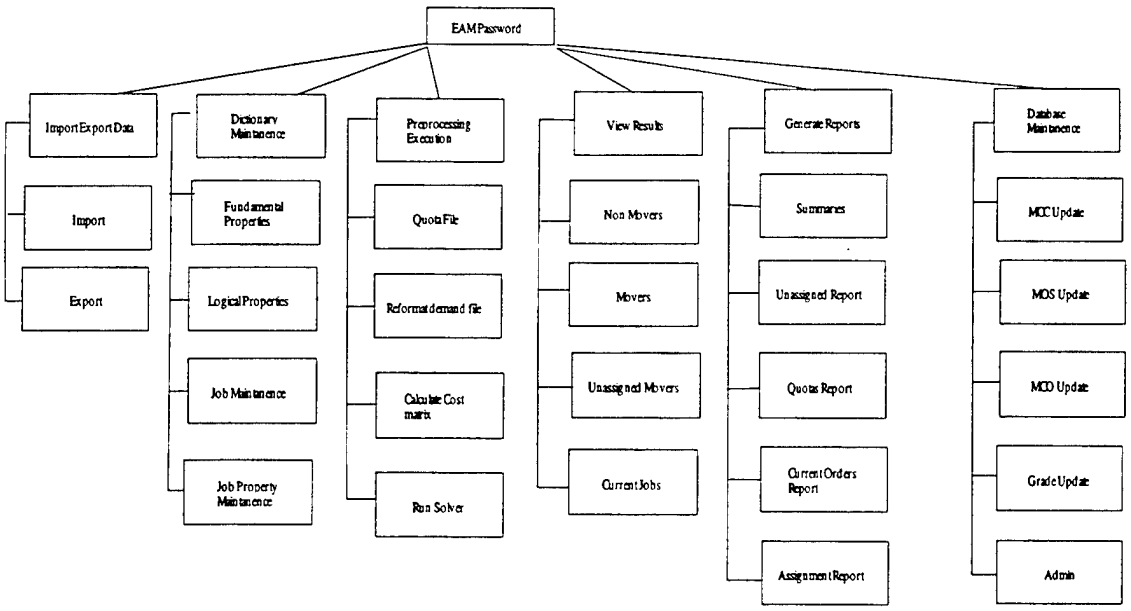


Figure 3: State Transition Diagram

The state transition diagram for the model manager is shown in Figure 3. This diagram shows the user interface between various 'switchboards' and the result generated by the code behind those switchboards. These features of the system should enhance the user friendliness and thus increase the use and understanding of the model.

The following section will be a series of screen shots and commentary that demonstrate the ideas behind the prototype and how a model manager would perform an EAM run using this prototype.

B. DEMONSTRATION OF THE 'MOCK' PROTOTYPE

The first screen that is encountered is the login screen, as shown in Figure 4. The EAM model manager would enter his login name and password. If it is accepted he now becomes an active user and his initials are made available to the rest of the system to maintain the identity of who is using the system and updating properties. A system time stamp is also entered into the Admin table to show the time of login.

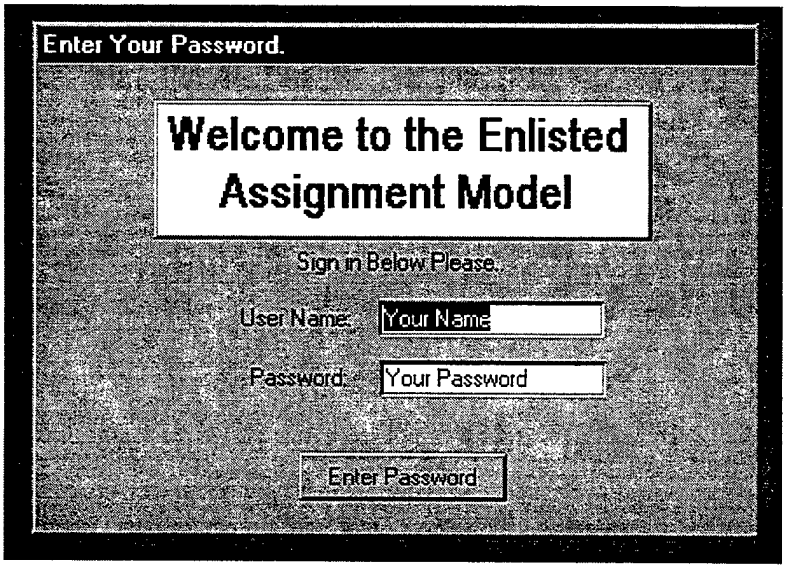


Figure 4: Password Form

1. EAM Switchboard

Once login is successful the user is presented with the main switchboard (see Figure 5). The switchboards were chosen to use as a means of navigating from various sections of the EAM in order to present the user with a fluid transition from the preparation of a run to the execution of the run.

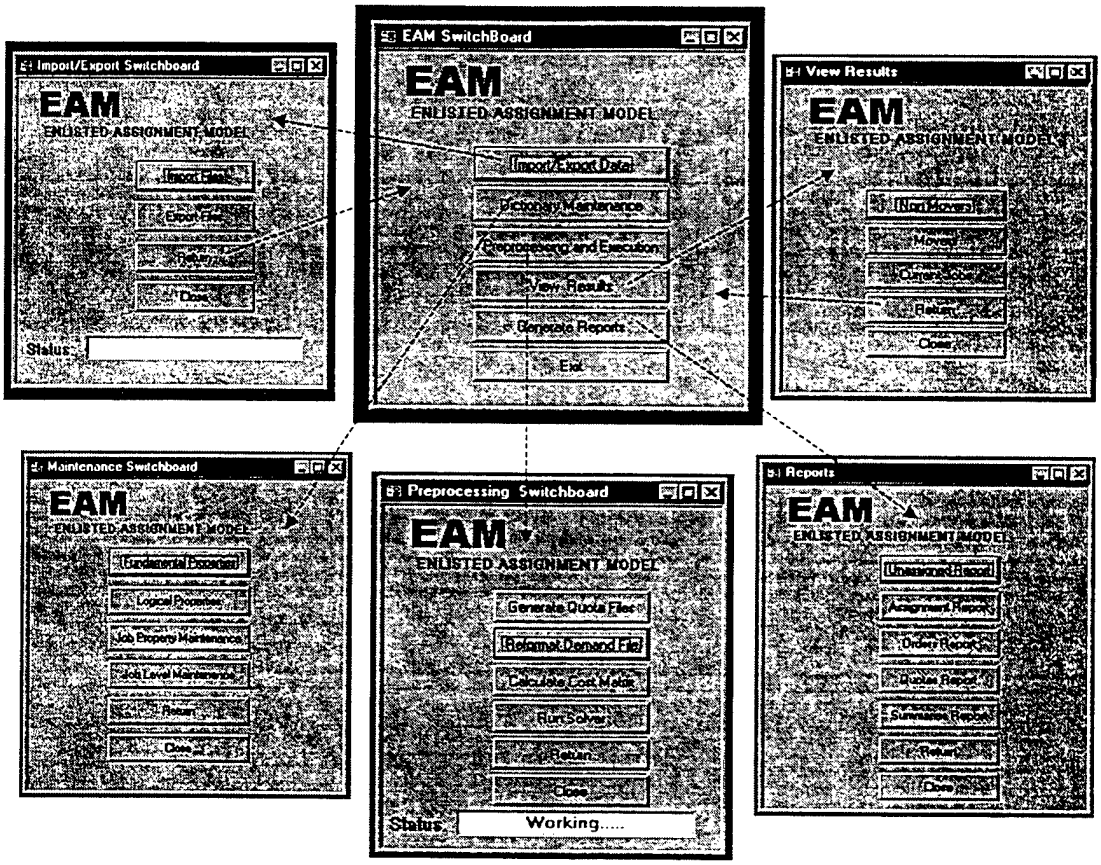


Figure 5: EAM Switchboard Structure

From the main switchboard the buttons are ordered in such a way as to present the user with the way in which the process should be followed. Top to bottom is the normal way in which the process is performed. Code could be introduced to force the user to follow a certain pattern if needed. From this switchboard the user is able to select Five different options. These options are explained below.

2. Import/Export Switchboard

By selecting the Import/Export Data button the user would see the switchboard that is the Import/Export switchboard (see Figure 6).

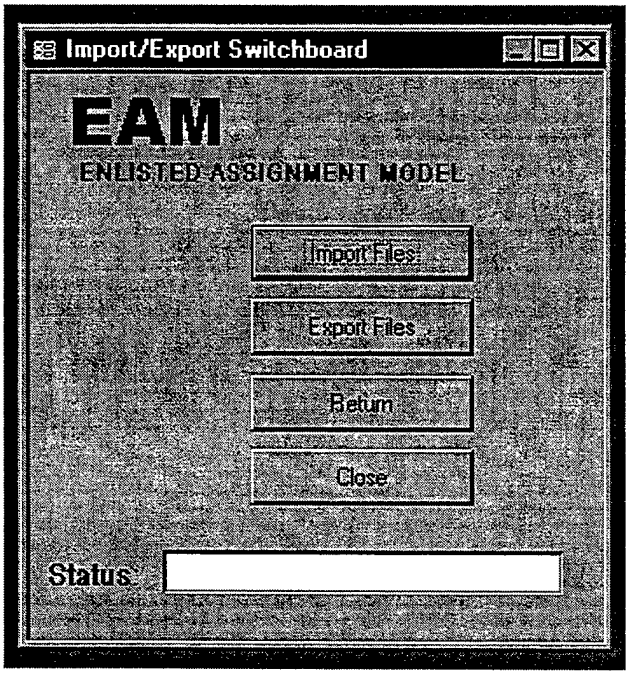


Figure 6: Import/Export Switchboard

This switchboard is used to import the ESGM file and the MCTFS file. The other use is to export the orders file that was generated by the solver. At this time the user is going to import the information to begin an EAM run. The Status window will give the user feedback as to which file is being imported and when the import is complete. The clock function also runs at this time so that the user can see how long this process took. This can be used for future reference. At this time the Marine table and the ESGM table

are created in the database. Once the import is complete the user returns to the EAM Switchboard.

3. Dictionary Maintenance Switchboard

The next step in the process is to perform Dictionary Maintenance. Once this button is pushed the user is presented with the Maintenance Switchboard and the selections that it offers (see Figure 7).

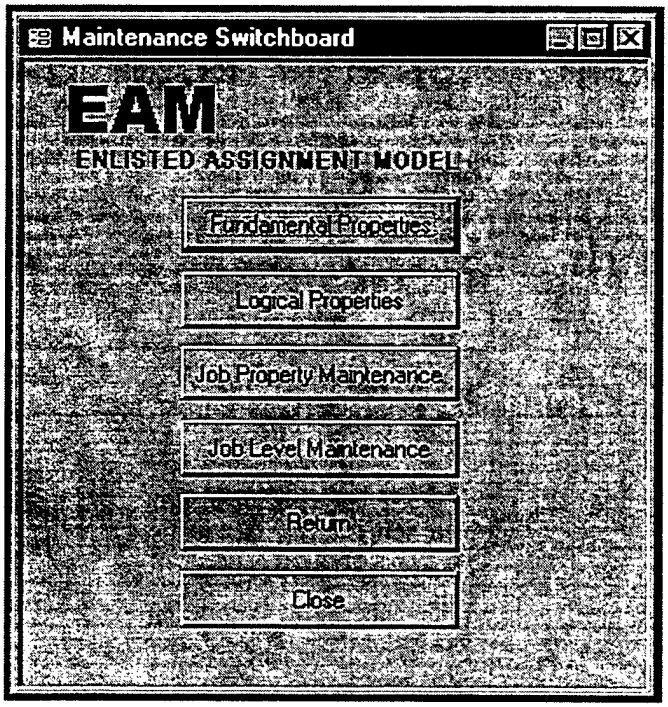


Figure 7: Maintenance Switchboard

From this form the user can perform the required maintenance on the EAM dictionary. Again this form is built with the idea that the user will work his way from the top of the form to the bottom. By pressing on the appropriate button the user is presented

with a more detailed form that will allow the user to manipulate that particular set of data. An explanation of each of these forms is presented below.

- **Fundamental Properties Form**

When the user presses this button he is presented with the Fundamental Property form (see Figure 8). He is able to update or create Fundamental Properties. These properties are based on the data that is available on each Marine that is imported in the MCTFS file.

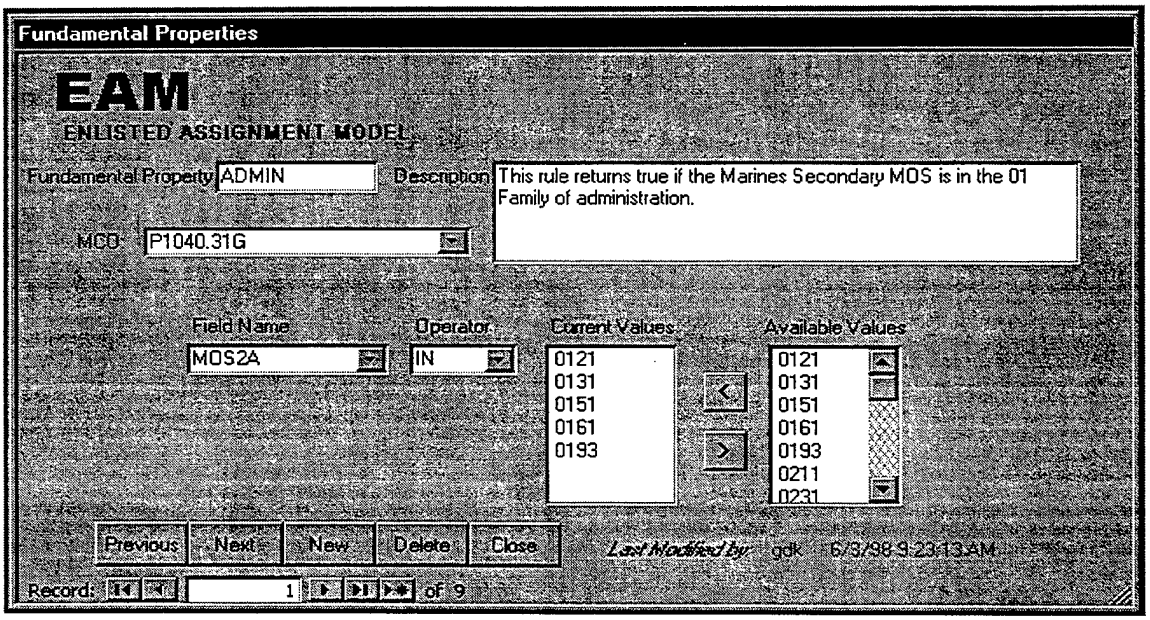


Figure 8: Fundamental Properties Form

Features that are implemented in this form are the mandatory reference to a Marine Corps Order and the identification of the person, date and time that this particular property was updated. All of these features add to the ability of the model manager to continually check his rules and make sure they are up-to-date with current Marine Corps

policy and guidance. This adds a way to maintain a system of checks and balances for the model manager and his superior - something that is not currently done in the current system. The Field Name is a dropdown list that is generated from the input fields' table. Depending on the field name and operator chosen, the form will present the items that can be selected in the value field. This type of input style keeps the user from making erroneous typing errors. The user must also enter a Description of what the property is supposed to be testing. Upon closing this form the Maintenance Switchboard is again opened.

- **Logical Properties Form**

When the user pushes this button he is presented with the Logical Properties form (see Figure 9).

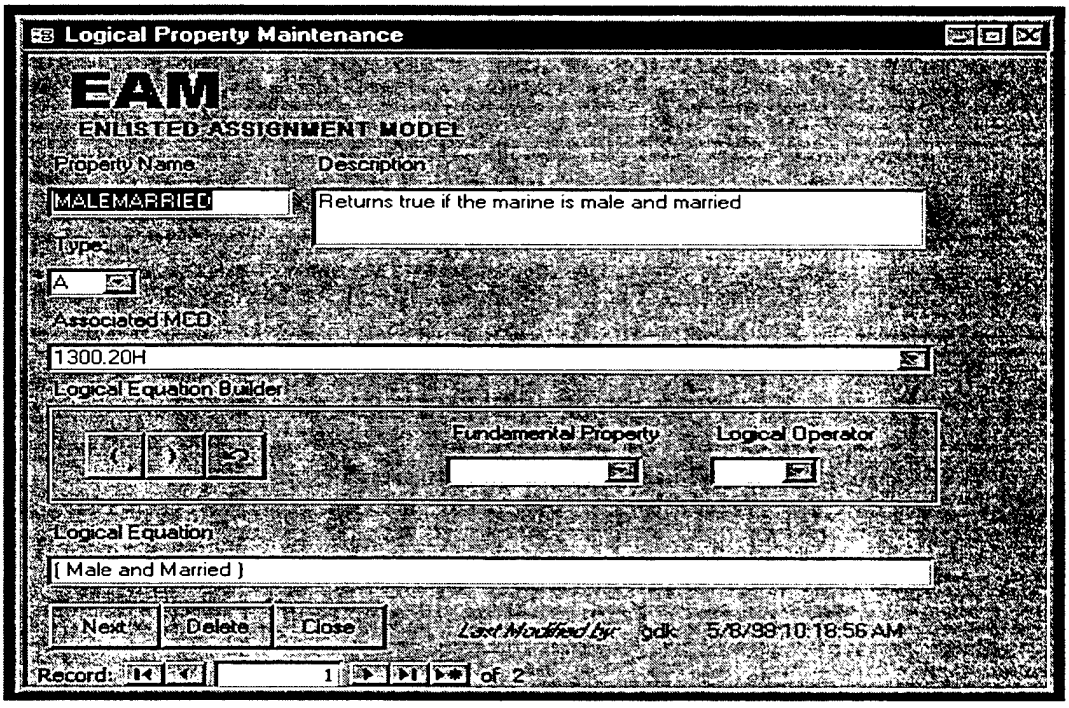


Figure 9: Logical Properties Form

Once Fundamental Properties have been created he is able to create or update Logical Properties. The Logical Properties form is similar to the Fundamental Properties form. In this form the user can 'build' Logical Properties that are based on Fundamental Properties. The difference between the current EAM and this prototype is that the Logical Properties can *only* contain Fundamental Properties. Limiting the input in this fashion force the user to use current Fundamental Properties that he must understand. The same functionality is built into this form as with the Fundamental form which includes the identification of person, date, and time of last update, along with the identification of the appropriate Marine Corps Order.

The way in which the user makes a logical property is by using the push buttons (left parenthesis, right parenthesis, and undo) to construct the appropriate logical equation. This logical equation, when evaluated, will produce a True or False response. The interface is very simple to use and it is expected that the user will have first written out the equation and thought out the implications of what the equation will do to the results of EAM. These properties, along with the Fundamental Properties, can now be associated to a job for further definition of what the criteria for filling that job is.

- **Job Property Level Maintenance Form**

Once the Fundamental and Logical Properties are made or updated then the user is ready to ensure that the Jobs are defined with Mandatory or Desired Properties. Opening the form for Job Maintenance the user is able to define which properties that the Marine *must* satisfy to be assigned the job and what properties that are *desired.* The form is

shown in Figure 10. The concept of defining the Job by properties will allow the ability to differentiate between Marines who could be more qualified for certain jobs than others.

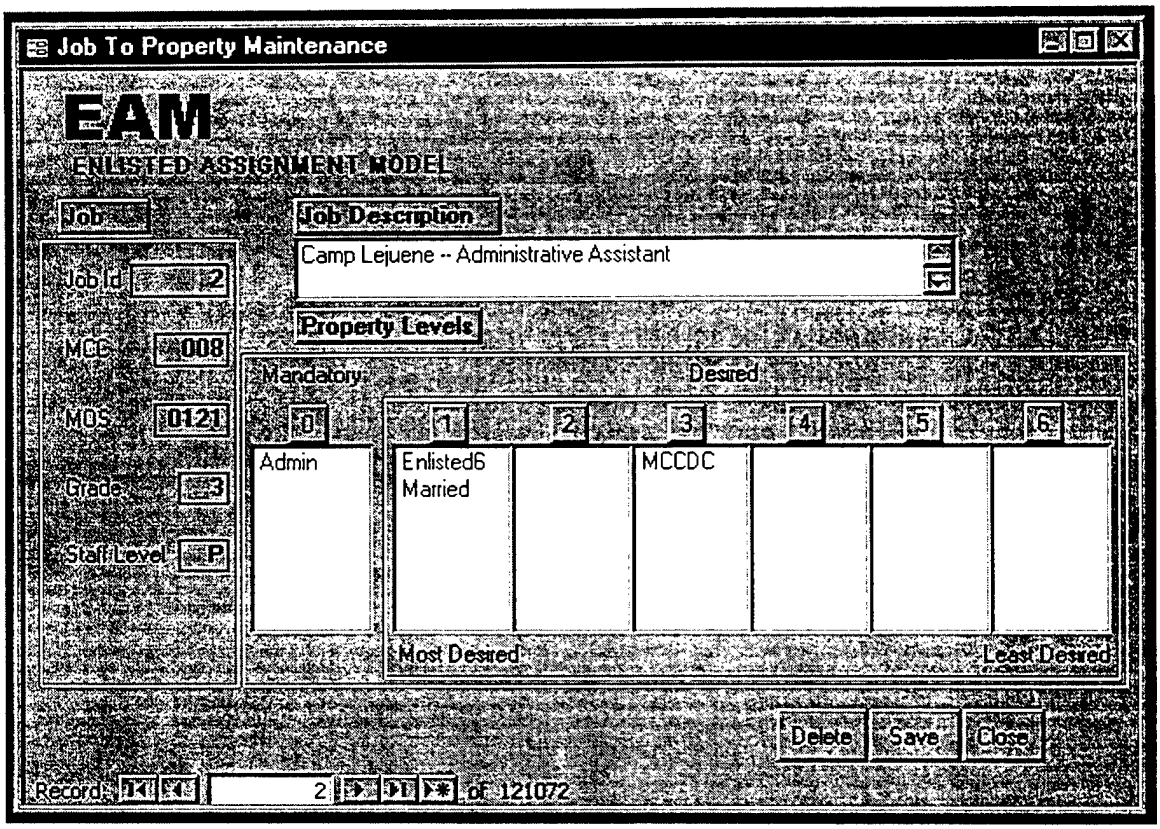


Figure 10: Job Property Level Maintenance Form

A Marine must satisfy all Mandatory properties (in this case one) or else that Marine is dropped from consideration from that job. Properties 1 - 6 are desired properties. In Figure 10 a Marine needs to have gone to Admin School to meet the mandatory requirements of Job 2. If the Marine is married and has enlisted for 6 years he is even more qualified (i.e. more fit). A weighting system for assigning the appropriate

values for these properties is being developed in another thesis that does a similar classification of Schools to Marines in the Recruit Distribution Model [Snoap 1998].

- **Job Property Maintenance Form**

If the user wants to change the properties or add new properties to any of the levels he just needs to click on the level number and the form for maintaining the properties will open to that level (see Figure 11).

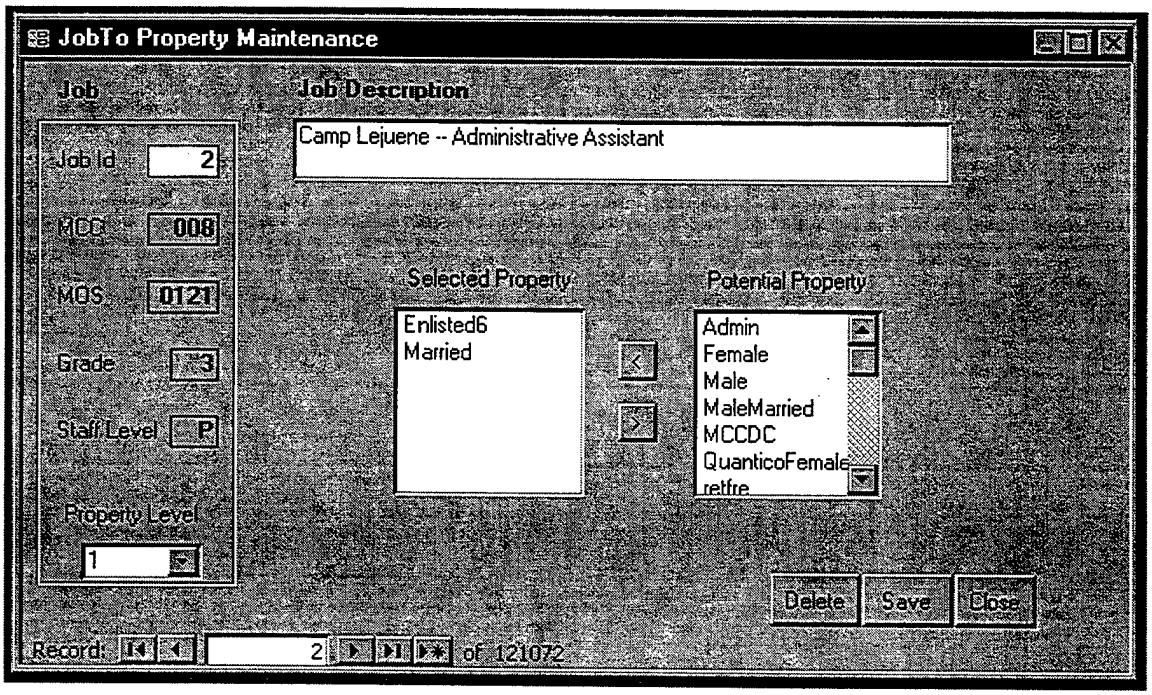


Figure 11: Job Property Maintenance Form

By highlighting the appropriate property the user can now transfer a property from the potential properties to the selected properties or vice versa using the arrow push

buttons. Once all of the maintenance has been performed the user closes the form and returns to the Maintenance Switchboard and is finished with the Dictionary maintenance.

4. Preprocessing Switchboard

Returning to the EAM switchboard the user is now ready to continue with the preprocessing of the MCTFS and ESGM files. Pushing the Preprocessing and Execution button opens the switchboard in Figure 12. This form accesses all of the VBA code that is associated with this form and the code that was discussed earlier. This form is also set up sequentially from top to bottom so that the model manager will need to start from the top and work his way down the form to ensure that all steps are performed.

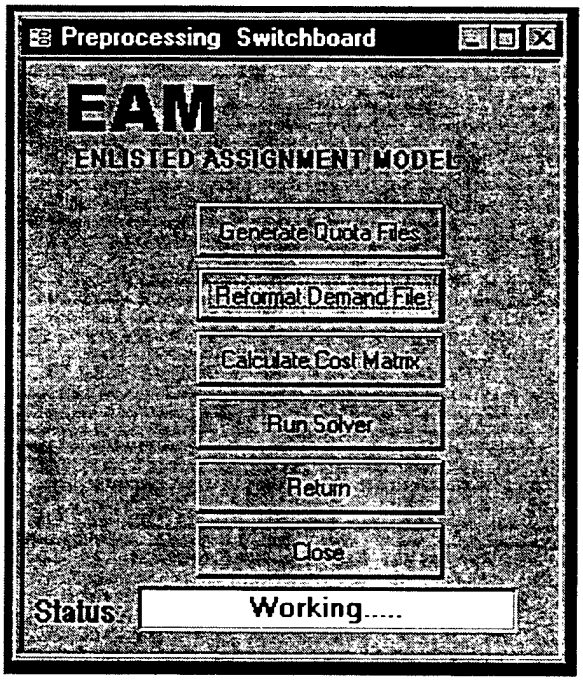


Figure 12: Preprocessing Switchboard

This form also has a status bar in order to ensure that the user is aware of what process is running and what the next step should be.

- **Draw Date Form**

The Generate Quota button will open the form in Figure 13. This form will ask for the draw date and the time on station (TOS) requirements. With this information the Movers, Nonmovers, and Jobs tables will be created. Once they are created the Nonmovers are subtracted from the ESGM file and the demand table is created. These tables are then related as appropriate.

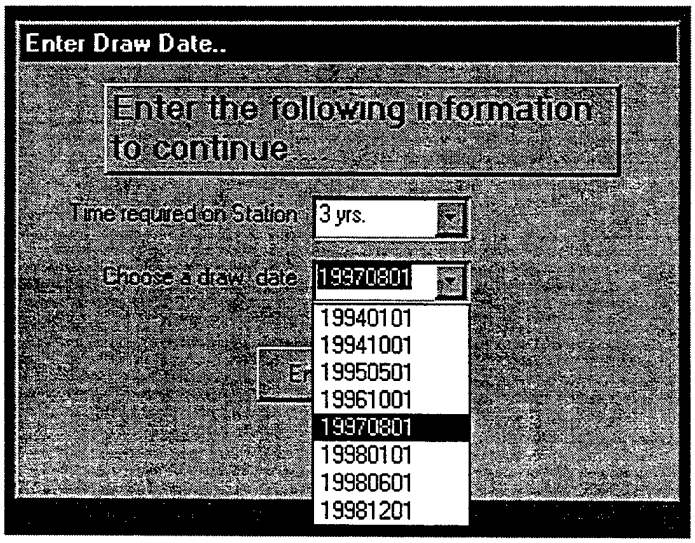


Figure 13: Draw Date Switchboard

This form will close automatically and then the next step is to get the demand table into usable form. By pushing the Reformat Demand File button (see Figure 12), the file is compared with the current jobs identified by ESGM, and quotas are created. The

table is then restructured to ensure that it is ready for the comparisons of the Movers and Jobs to calculate the Job Cost of each Marine.

The next step is selecting the Generate Cost Matrix button. This process is the most time consuming and it also generates the data that will be fed into the solver. Since every job must be compared with each Marine and then a cost for that combination must be calculated and stored, this process can take up to six hours using the 'mock' prototype. In order to reduce the number of variables, the process will automatically drop Marines that are of infinite cost to the Marine Corps for any of the jobs.

The output of this segment is seven text files that will be used by the solver to make assignments. Four of the files are the actual cost matrix and the remaining three are the Marines Id, Jobs Id, and the Demand or quota files. These files can be formatted into any format that is needed by the solver, and are output to whatever directory that the solver will look for them in. At this time the code must be changed by hand. The ability to enter the data into a user-defined directory can easily be implemented.

Once this step has completed then the Run Solver button is selected and the process of assigning the best-qualified Marines to Jobs begins using the cost data generated above. As the solver comes to a solution the information is then read back into the various tables for access by the model managers and ultimately the Marine Monitors.

5. View Result and Report Switchboards

Upon returning to the main switchboard the View Results and Generate Reports buttons are of use to the user in that they allow him examination of the results in a quick and easy fashion (Figures 14 & 15).

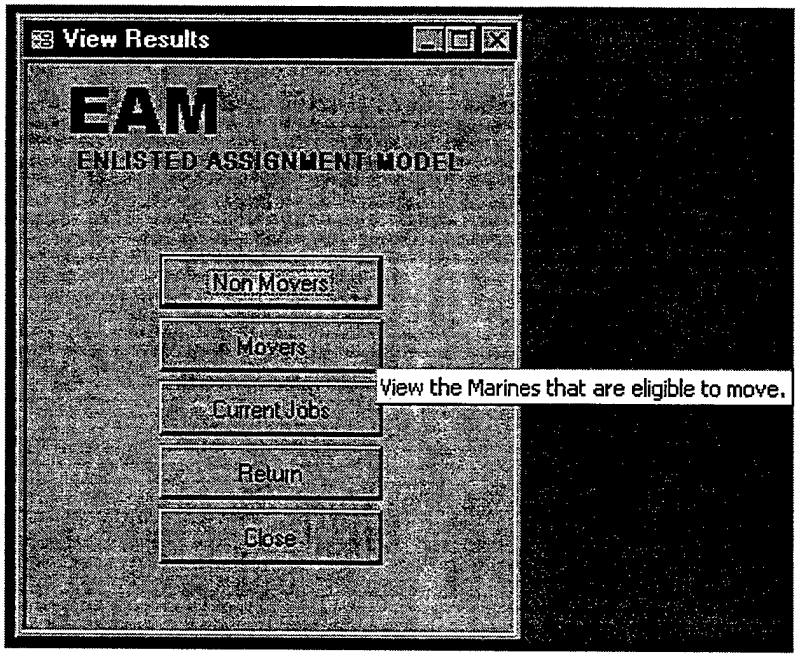


Figure 14: View Results Switchboard

As is shown on these screenshots the use of ToolTips is consistent throughout the prototype to prompt the user and to inform him of what he is about to do. Each of the buttons that are on these forms is connected to an SQL statement that queries the specific tables that contain the data that is being requested. The ability to change the queries is not implemented at this time but would require only a minimal effort to introduce user generated specific queries.

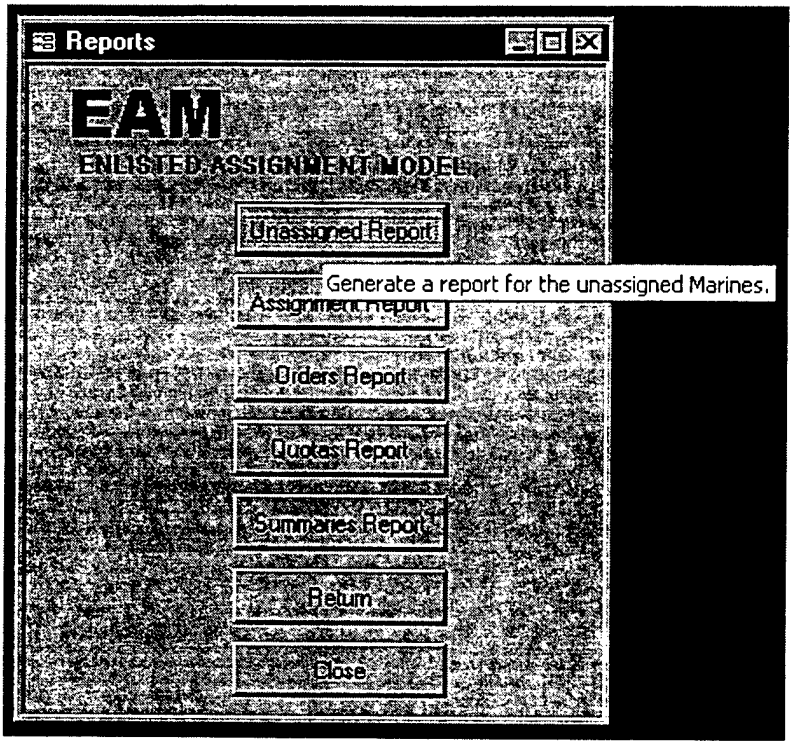


Figure 15: Reports Switchboard

6. Database Maintenance

The final area of interest that is not shown on the EAM switchboard but is represented in the state transition diagram is the area of Database Maintenance. This includes but is not limited to the upkeep of the current MOSs, current MCCs, Grade, and Family (which MOS belongs to it). These forms are shown in Appendix E.

The types of values that the model manager should be able to input into the values field of the Fundamental Properties form (see Figure 8) are important to check in order to prevent erroneous entries. That is what the form in Figure 16 accomplishes. This form lists all of the current input fields of the MCTFS file. These are the attributes of the

Marines. In order to build properties that make sense the Fundamental Property must be built off of one of these values. In order for the system to check for data entry errors, the user is given the option to define the user input so that there can be no mistake as to what needs to be entered.

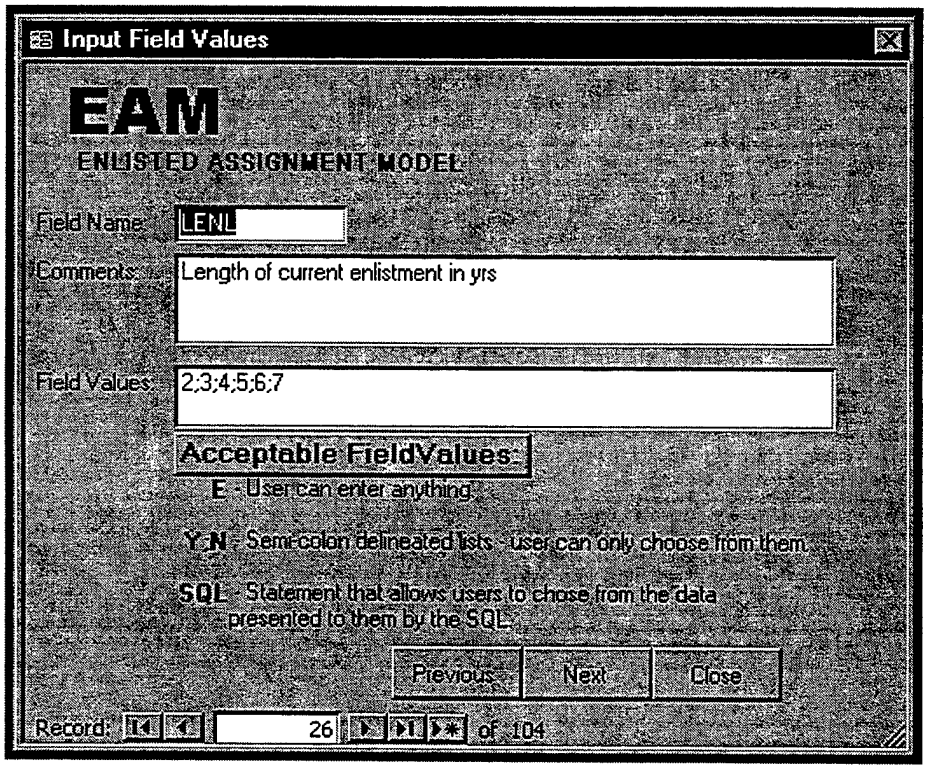


Figure 16: Input Field Values Switchboard

As is shown in the Figure 16, the input field LENL is the length of enlisted service. The only values that the user is able to enter into the field for building a rule are the values 2;3;4;5;6;7. The user can enter an **SQL** statement that would determine the values or the user could just enter an **E** for allowing any entry.

The remaining forms for this section are shown in Appendix E. They are self-explanatory and they are easily tied to any other form through the use of button

commands. An example would be if the user were trying to build a Logical Property and discovers that the Fundamental Property that he intends to use is not properly defined to meet his needs. A button could be easily placed on that particular form to allow him to go directly to the Fundamental Property form without having to back all the out of his current position in the hierarchy.

IV. PROTOTYPE DEVELOPMENT

In designing the prototype the process of Assign Marine was broken down into components that could be more easily understood and therefore offer a better understanding of the assignment process. The new system was designed specifically with the user in mind and to enhance his ability to use the system to improve the assignment process. In order to use this prototype a Pentium computer with at least 32M of RAM, 4G of hard drive space, and Access 97 are required. The prototype currently generates the Job Cost matrix for the solver but is not 'linked' to the solver software as of this writing.

The development of the 'mock' prototype begins with showing the overall EAM environment and breaking out a manageable set of components that will be specifically implemented in development. These components are modeled by the use of a relational DBMS (Access 97) and then links to the visual interfaces are created in Access by the Rapid Application Development (RAD) language Visual Basic for Applications (VBA).

By designing the prototype in this manner it is expected that the benefits of a simple and intuitive user interface will be realized. Also, the ability to perform maintenance, preprocessing, and data access quickly and efficiently will be evident and the streamlining of the actual EAM run will provide a baseline for developing improvements in the next implementation of the production version of EAM.

A. SYSTEM ARCHITECTURE

Figure 17 below shows the overall environment that the EAM system resides in. This simplified view allows the breakdown of specific elements that can be used to develop the 'mock' prototype, which will be utilized to highlight the areas of suggested improvement.

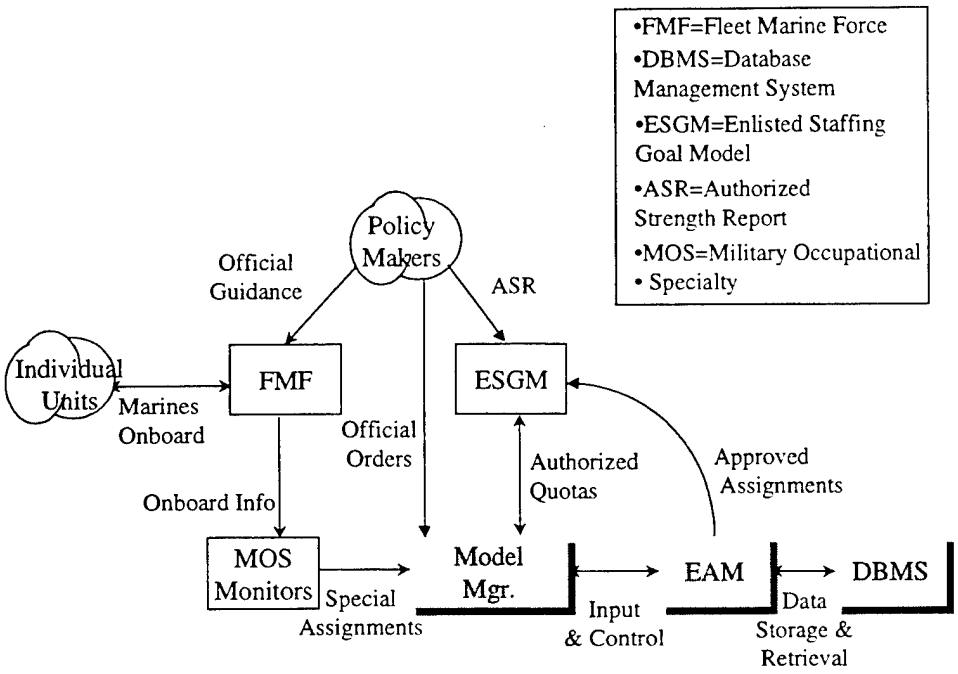


Figure 17: EAM Environment

In Figure 18 components that were emphasized by gray boxes in Figure 17 have been used to give an overview of the 'mock' prototypes system architecture. The highlighted areas are broken down in the following way.

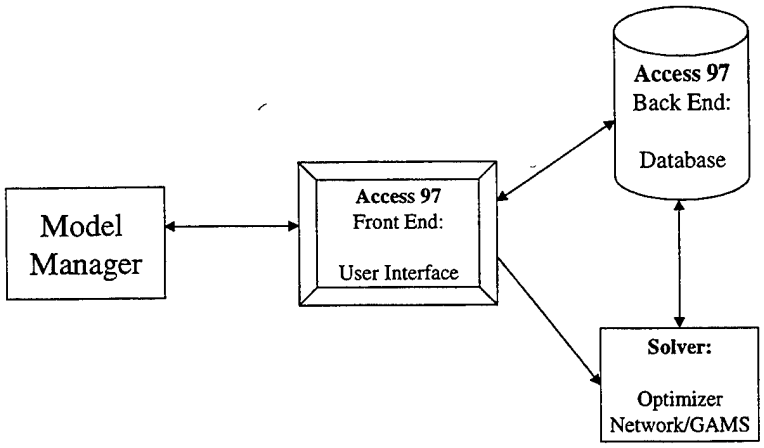


Figure 18: Components of EAM Prototype

B. DATA MODEL DESIGN AND OPERATION

The data model for the EAM is depicted in the Entity Relationship (ER) diagram in Figure 19. All of the entities on this diagram correspond to the entities that will be discussed in this chapter. The primary entities for this database design are the **Marine**, **Job**, **Job Cost**, **Logical Properties** and, **Fundamental Properties**. These entities are created, read, updated, deleted, or archived by a function that has been identified during the process analysis.

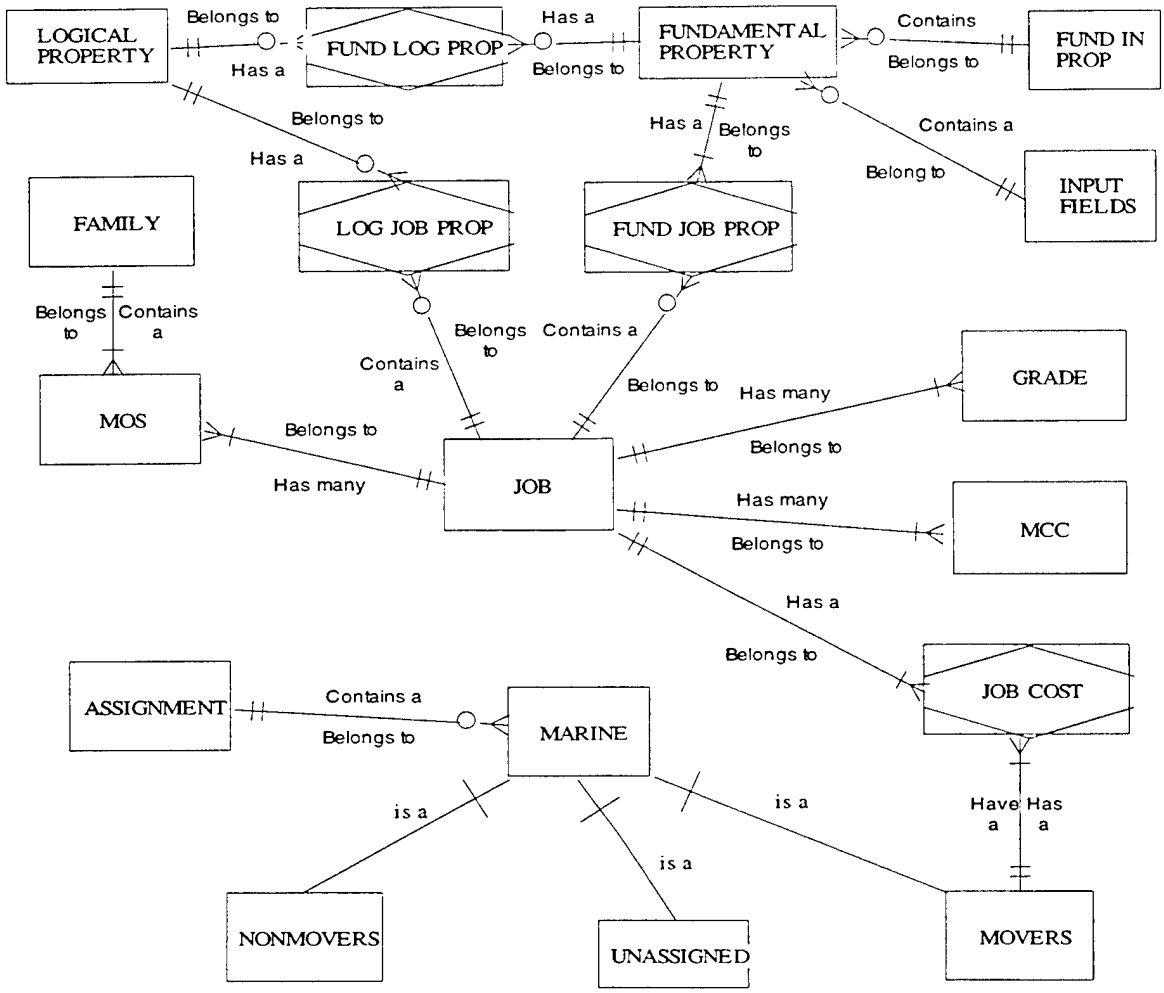


Figure 19: ER Diagram

A Job requires certain Fundamental or Logical Properties to be met in order for a Marine to be qualified to be assigned to that specific Job. The Fundamental and Logical Properties are "built" using the same procedure as the current EAM system. The author has restricted the use of these properties to being associated with the Job only. They are therefore only associated with a specific Job (**ex**. Job 2 requires a MOS of 0311, a grade of E5 and the completion of NCO school; This could be three Fundamental Properties or

one Logical property). If the Marine meets the criteria for a specific job he is assigned a Job Cost. A Job Cost is the combination of how many criteria that he meets for that specific job related to his pay. The more qualified the Marine is for the job the lower cost to the Marine Corps for assigning him that job. Once this Job Cost matrix has been formed then the Jobs, Marines and Costs are exported to a solver to determine the best 'fit'.

C. DATABASE MANAGEMENT SYSTEM

The DBMS used in developing the relational table structure for this model was Access 97. The database tables are an extension of the ER diagram in Figure 19. The full table structures and relationships are depicted in Appendix C. The following sections describe the resulting tables and also the tables that were not depicted above but were a functional subset of those tables that are represented in the ER diagram. These tables provide additional functionality and accessibility to data that otherwise would not be readily available.

1. Tables Derived From the ER Diagram

a. Marine Table (Marines)

This table represents the individual Marine. It stores all of the attributes that are associated with each Marine. These attributes will be used to determine whether or not the Marine is eligible for a certain job. The MCTFS file is imported into this table.

The Marine table has a unique identifying field named MarId. This is an auto-generated number that is used throughout the EAM process to uniquely identify the individual Marines. The Mover (tblMovers), Nonmover (tblNonMovers), and Unassigned (tblUnassignedMovers) tables are all tables that are created at runtime to hold the various types of Marines the are processed by the prototype.

b. Job Table (tblCurrentJobs)

This table represents the individual Jobs that are available to be filled using the current ESGM data. It stores all of the attributes of the unique jobs that are identified by an MCC, MOS, and Grade. The Job table has a unique identifying field named JobId. This is an auto-generated number that is used throughout the EAM process to uniquely identify the individual Jobs in the Marine Corps. At this time a field called StaffLevel is also included to indicate whether the job is an Excepted Command, Priority Command, or a Common Command. The staffing level determines whether or not a command will receive on grade, on MOS assignments or whether a substitute will be allowed. During runtime a Job table is created called tblJobs that specifically addresses only the Jobs that have been identified through the current EAM run. The unique identifying field for that table is the JobId from the table tblCurrentJobs. The additional field in this table that is different from tblCurrentJobs is the Quota field that identifies the number of Marines needed for that specific job on that run.

c. Fundamental Properties Table (tblFundamentalProperties)

Fundamental Properties are properties that are developed from the attributes that exist in the MCTFS data on each individual Marine. The properties will be associated with a Job and tested against a Marine record and will be either True or False dependent upon the Marines attributes in a similar fashion as the current EAM (see Appendix B: Properties Table). This table contains all of the Fundamental Properties that are defined by the model manager. The Fundamental Properties table contains the FundamentalPropName_PK as the unique identifier. The fields associated with this table are: FieldName (obtained from the tblInputfields table), Operator (=, <, >, <>, IN, NOT IN), Value (the criteria for the FieldName to be associated with; input by the user), Description (a used friendly description of what the property is supposed to test) and automated fields DateTime (system generated time stamp), Initials (current users initials), and MCOId (used to identify a property with a specific Marine Corps Order). These fields will keep track of which property was entered into the database and on what basis.

d. Fundamental IN/NOT IN Table (tblFundPropIN)

This table tracks the values that are associated with each IN or NOT IN operator of a specific Fundamental Property. This table contains the composite key of PropName_PK and Value. This combination gives a unique identity to this entry that is associated with the IN and NOT IN operators of the Fundamental Properties Table. This allows multiple values to be selected for these operators as their name suggests.

e. Input Fields Table (tblInputFields)

The Input Fields table is a reference to the current names of the attributes of the Marines that are imported in the MCTFS file designated by FieldName. These are the most current attributes for the Marine and are used to build all of the Fundamental Properties. The other fields in this table are; Comments, for a short description of the FieldNames, and a field called FieldValues. This field is used to limit the user to only the data entries that are valid for that particular FieldName, thus producing accurate data input. The values that are allowed in this field are **E** (for any user input), semi-colon delineated lists (Y;N) or any legal **SQL** statement (**ex**. SELECT * FROM tblMCC).

f. Logical Properties Table (tblLogicalProperties)

The Logical Properties are Fundamental Properties that are combined with logical operators (**and**, **or**, **in**) to achieve a desired result from the available Marines. The properties will be associated with a Job and tested against a Marine record and will be either True or False dependent upon the Marines attributes in a similar fashion as the current EAM (see Appendix B: Properties Table). The Logical Properties table contains the LogicalPropName_PK as the unique identifier. The fields associated with this table are; the LogicalEquation, (a longhand notation of the logical combination of fundamental properties, **ex** ((MALE and SINGLE) or (MALE and ENLISTEDSIX))) which is associated with the property definition, a Description, (user friendly description of what the property is supposed to test) and automated fields DateTime (system generated time

stamp), Initials (current users initials), and MCOId (used to identify a property with a specific Marine Corps Order). These fields will keep track of which property was entered into the database and on what basis.

g. Fundamental Job Table (tblFundJobProp)

This is an associative table that breaks a many-to-many relationship. Since a Job can have many Fundamental Properties and vice-versa this table ensures that a Job is associated properly with a particular Fundamental Property. The table uses a composite key of the FundPropName_FK from the Fundamental Properties table and the JobId of the Job table. There is an additional field Level. The property will be evaluated at the appropriate level. The values that are in the Level field are 0 (mandatory), or 1 (most desired) through 6 (least desired).

h. Logical Job Table (tblLogJobProp)

This is an associative table that breaks a many-to-many relationship. Since a Job can have many Logical Properties and vice-versa this table ensures that the Job is associated properly with a particular Logical Property. This table uses a composite key of the LogPropName_FK from the Logical Properties table and the JobId of the Job table. There is an additional field Level. The property will be evaluated at the appropriate level. The values that are in the Level field are 0 (mandatory), or 1 (most desired) through 6 (least desired).

i. Job Cost Table (tblJob_Marines)

This table is used to collect the calculated cost of a specific Marine filling a specific job and will be referred to as the Job Cost matrix. Every available Marine will be assigned an appropriate cost with each available job. A predetermined cost will be set as a limit to the amount that the Marine Corps is willing to accept for filling a particular job. If the Marine/Job cost exceeds this limit, the solver will not consider that particular Marine/Job combination. This will eliminate some of the variables that will be present in the Job Cost matrix. This table uses a composite key of the MarId of the available Marine and the JobId of the assignable Job. The only other field is the Cost field. This field is used to store the cost of putting a specific Marine in a specific Job.

j. Marine Corps Order Table (tblMCO)

This table contains all Marine Corps Order's (MCO) that are applicable to Manpower Assignment Policy. Orders that are entered into this table should reflect policies and requirements that govern Marine Corps Manpower Management. This information is used during the definition of Fundamental and Logical Properties. This will allow the model manager to identify which properties will need to be changed when a MCO gets changed. An auto-generated OrderId uniquely identifies this table. The MCOrder field contains the referenced number of the MCO and the MCOTitle field contains the subject of the MCO.

k. Assignment Table (tblAssignment)

This table stores all of the necessary attributes that are associated with each *assigned* Marine. The Assignment table has a unique identifying field named MarId. This is the number that was associated with a Marine when he first entered the EAM. The fields for this table contains all of the personal attributes that will be needed to process a Marines Order once he has been assigned.

2. Tables Not Derived From the ER Diagram

a. ESGM Table (tblESGM)

The data for this table is imported from the ESGM file. This data will be used to generate the demand table and eventually be used to develop the Job Cost matrix. The ESGM table uses a composite key of MOS and MCC. The remaining fields represent the grades E2 - E9.

b. MCC Table (tblMCC)

This table contains the MCC's that are currently in the ESGM file. The MCC table has unique identifying field named MCCId_PK. This is an auto-generated number that uniquely identifies this MCC. The other fields include the Description field to give a short description of the MCC, and a TO_Number field that can be used for future use in describing more specifically the MCC.

c. MOS Table (tblMOS)

This table contains the MOS's that are currently in the ESGM file. The MOS field uniquely identifies the MOS table. There is also a Description field that allows a brief description of the MOS.

d. Grade Table (tblGrade)

This table identifies all of the current enlisted grades. The unique identifier is GradeId_PK. It is an integer corresponding to the grades E2 through E9. In the model E1 is treated as an E2. The other field is the Pay field, which is the basic pay of that particular rank of Marine. These values will be used to develop the cost matrix, which will determine the Marines 'cost' for a Job.

e. Family Table (tblOCCFields)

The Family table consists of the OCCField which is the breakdown of MOSs (first two numbers of the MOS determine its OCCField) by Occupational Specialty. This is used to establish a smaller set of sort criteria (55 OCCs vice 844 MOSs). There can be many MOSs in each OCCField. The other remaining field is the Title field for a brief description of the MOS.

D. DATA PREPROCESSING AND PRESENTATION

The tables that were explained in the previous chapter contain all of the appropriate data needed to make assignments. In order to make assignments in an optimal fashion certain preprocessing must be accomplished. The current EAM does all of the preprocessing in a way that fragments the data and keeps it hidden until the run is over and then reports are generated. By using a relational DBMS such as Access 97, the developer can take advantage of the underlying programming language Visual Basic for Applications (VBA) and its Data Access Object (DAO) capabilities to quickly access all aspects of the data which will in turn provide immediate feedback to the model manager via datasheet views and meaningful forms.

Another benefit of using a Rapid Application Development tool such as Access 97 is the ability to experiment with visual interfaces and the real-time feedback and flexibility that these design features offer. The use of VBA programming also allows the developer to exercise more precise control over the data and thus produce solutions that approach a production version result.

The following is a list of the Visual Basic procedures that perform the behind the scenes preprocessing work. This code is accessible from any of the forms that have been designed. The concept of using a friendly graphical user interface to run this code was chosen to provide a simpler view into a complex system in order to attempt to maximize the users understanding of the system and present him with a step by step methodology for maintaining and executing EAM.

This area is divided into two sections; 1) Basic code for the *operation* of EAM and 2) Basic code for the *maintenance* of EAM. It should be stated that this code is just the code that is used to preprocess the information about Marines and Jobs and to develop a Job Cost matrix for the mathematical solver. The functionality of this prototype is a very simplified version of the real system.

As will be seen later, each individual form also contains its own code that is used to provide that particular form with added flexibility upon loading. The author has not chosen to insert that code in the interest of brevity. Anyone who uses Access 97 should become intimately familiar with that part of the user interface design process in order to use this tool to its fullest potential.

1. Operation of EAM

The following descriptions are provided to give the user a good understanding of what functions make up the actual preprocessing code. This code is used to generate the appropriate tables that take the Marines, Staffing Goals, and the Jobs and prepare the Job Cost Matrix for the mathematical solver.

a. Function CreateDemand()

The function CreateDemand is called in order to create a duplicate of the Enlisted Staffing Goals. Once this is accomplished this data becomes a temporary demand table. The Marine Nonmovers are compared against ESGM and subtracted from this demand table. The result is the quotas that need to be filled during this run.

b. Function ReformatDemand()

This function takes the demand file that was created above and reformats it into a usable form. This table becomes the Jobs table. Through the quota field and the Staffing Level field the solver will determine how many Marines a command needs and in what priority it should fill them.

c. Function CreateRelationship()

This function creates a one-to-many relationship between two tables. The function needs to know the name of the relationship, the primary table, the foreign table, and the indexed key. This function is used frequently to establish referential integrity after tables have been created, deleted, and re-created.

d. Function ArrayCost()

This function is used to calculate the cost associated with each Movable Marine as compared to the available jobs from the Jobs table. The prototype currently checks for three attributes and assigns a cost to the Marine. The three attributes currently checked are:

- MOS
- Grade
- Command

2. Maintenance of EAM

The following descriptions are provided to give the user insight into what functions are used to provide some capability for user maintenance of the system and some administrative capabilities. Other functions were created for the prototype but are not covered here. It is recognized and furthermore it is hoped that many of these functions will be expanded on and improved with more user feedback and use of the model.

a. Function Clock()

This function is used to time the preprocessing events in EAM. These times are generated by the system clock and give the model manager feedback on the amount of time that it is taking him to get a result. This is also useful to check out different sorting algorithms.

b. Function SignOut()

This function is used to check which user is currently using EAM and sign them out of the system. If this system is implemented as a multi-user system this will enable the system administrator to allow multiple access.

c. Function FundProps()

This function is used to read the input fields that are requested in the fundamental properties table and ensure that the user is only able to enter the values that

are allowed. The function takes its value from the table input fields in the form of an SQL statement or a semi-colon delineated value list (**ex** Y;N). The manager now has a way to ensure that values for the Fundamental Properties are entered from uncorrupted data.

d. Function DeleteSelected()

This function is used to delete selected values from the fundamental properties that contain an IN or NOT IN statement. Through a pair of list boxes the user can add or delete items. This function performs the delete operations, which remove the values from the Fundamental IN/NOT IN table.

V. CONCLUSIONS

In summary, the author introduced in Chapter I the reason for this thesis and the background behind its inception. In Chapter II the author presented an analysis of the current system as it exists today and as understood by the current users of the system. A critique of each functional area was included in this chapter. In Chapter III the author discussed the redesigned user interface specifically, as well as presented a tour through the current 'mock' prototype. In Chapter IV an overview of the model, data and design concepts were presented in order to show the development background and the technical structure behind the 'mock' prototype.

A. LIMITATIONS IN THE CURRENT 'MOCK' PROTOTYPE

The author has attempted to present a very general picture of what is possible with the current system given the tremendous capabilities of the new visual design technologies and their associated programming code. The prototype is in no way capable of replacing the old system. It is hoped that the ideas of this thesis will generate some renewed desire to generate accurate assignments in a new environment that can be made understandable and explainable.

Through the use of a redesigned user interface it has been shown that some of the most complex elements of the EAM can be made understandable and as such could easily be explained to the MOS monitors. The system is only as capable as those who run it and take the time to understand it, as has been evidenced by the way in which the current

EAM is operated. By making the system easier to understand and the data more accessible it is hoped that the urgency for improving the system will be heightened and the pursuit of accomplishing the Marine Corps mission through this system will be rekindled.

There are currently at least 121,000 unique jobs identified by ESGM. All of them are not required to be filled. They are unique in that they are a combination of MOS, MCC, and Grade. The prototype has been set up as to establish mandatory and desired properties for each job. This is an insurmountable task for one model manager. Suggestions on how to share this burden are illustrated under possible enhancements.

The author is quick to point out that the ability to improve on this work is easily recognized when the full functionality of the current EAM is considered.

B. POSSIBLE ENHANCEMENTS

The following enhancements to this prototype are suggested:

1) The current draw date and TOS requirement screen is not totally developed to incorporate the current format of the date.

2) The current families and the MOSs need to be considered more closely. This prototype will run for the whole segment of all MOSs. The current EAM model recognizes that there is redundancy in some of the solution techniques and thus accounts for that by solving by families and ensuring a quicker runtime. The current EAM contains 73 Families defined specifically for EAM. The Marine Corps has 55 Occupational fields. Both are associated with all of the MOSs.

3) The use of full object oriented code (i.e. C++ or JAVA) could easily be incorporated into the design and implementation process. Using this type of coding will take advantage of the newer memory usage technology that will dramatically improve the preprocessing runtime. The current state of computer hardware should now make it evident that the ability to achieve a solution within a few hours of entering the data is not an insurmountable issue.

4) The current prototype is written strictly to be run from an Access 97 database, but the table structure and the SQL code that currently underlie the database can easily be imported into Oracle and thus the back end data structure can be installed in an Oracle DBMS. Slight modifications to the SQL would be in order as Microsoft SQL is not fully compatible with ANSI 92.

C. FURTHER RESEARCH

The following areas should be considered for further research in this area. The model is currently run by one individual. He is responsible for ensuring that all of the entries are up to date and that the data dictionary is current. With the current move toward web-based applications and the flexibility that they provide it is not inconceivable that this application could be used over a web-based intra- or internet. With certain restrictions built into the pages, the MOS monitors would have real time access to last months run and they would also have input into the next run.

The way in which the prototype is constructed commanders that own the jobs, or their staff, could quite easily enter an area of the database to update the properties for

their jobs. This would alleviate the pressure on the model manager to ensure that he manipulate the model in order to satisfy the customer.

D. SUMMARY

The ultimate goal of the Marine Corps assignment process has been stated quite clearly, " to put the right Marine in the right place at the right time with the right skills and quality of life". The purpose of this thesis has been to examine the current process of doing that and to attempt and to improve the process by which the Marine Corps does business. By examining the current system, developing a RAD prototype to test the newest database and interface designs, it is hoped that some areas for improvement have been recognized. By implementing or at least considering the improvements suggested, and continuing the pursuit of the mission as stated, the current EAM process could be updated into a solid Decision Support System. This DSS would be capable of truly accomplishing the mission at hand and doing it with great savings to the Marine Corps well into the 21st century.

APPENDIX A: IDEF0 DIAGRAMS AND DEFINITIONS

A. PREFACE

Model Name: Enlisted Assignment Model
Definition: The Enlisted Assignment Model is used to identify to the Model Manager the current Marines that are eligible to be reassigned according to the appropriate open billets that exist as per the current Staffing goals that are generated from the Enlisted Staffing Goal Model.
Scope: Develop the Business Process that currently exists for the Enlisted Assignment Model.
Viewpoint: Enlisted Assignment Model Manager
Purpose: Identify the functionality of the current Enlisted Assignment Model specifically identifying the subprocesses that are associated with the main operation of Assign Marine.
Author Name: Gary D. Koch Jr.
Creation Date: 10/2/97

A Structured Analysis and Design Technique (SADT) known as IDEF0 was used to model the activity of assigning the enlisted Marine. This model is just a tool that was used to help the author understand and attempt to describe the relationships of the processes that interact with each other in EAM. It is in no way a complete representation of the entire Enlisted Assignment Process.

An IDEF0 model has a single subject. The subject of this model is "Assign Marine." Ensuring that one has obtained the correct subject is critical to the development of the model. The author has attempted to define the boundaries of the system; what is outside and what is inside the system.

An IDEF0 model has one viewpoint or perspective. This particular model is that of the Enlisted Assignment Model manager. This viewpoint will be used throughout the description of this particular system. Different views would obviously yield different descriptions of the system being modeled.

The IDEF 0 software BPWIN 2.0 was used to develop the top-down diagrams. The diagrams start with a general diagram and are decomposed into more specific diagrams that outline the remaining activities of the system. The diagrams use boxes, which represent functions or activities and arrows that represent interconnections between the boxes. The boxes are numbered alphanumerically and represent the origin and the

path used in the development of the model. A0 is the top-level diagram. This diagram will contain other boxes that will be labeled with a single numeral (1,2,3). When these boxes are decomposed, they will become the A1, A2, A3, etc... diagrams. This process continues through the last level of the model. The arrows identify information or data needed to carry out the functions or activities. An arrow coming into a box from the left, is input data, while and arrow coming out of a box on the right side is output data. An arrow coming from the top shows a control order while an arrow coming in from the bottom shows a physical resource or mechanism required to do a function.

Identifying inputs, controls, mechanisms and then referring to a detailed description in Chapter II of the functioning will explain each box of the model of the activity. The purpose of explaining each box is to describe how each activity functions independently and the interaction required with other activities of the system.

B. NODE A-0: CONTEXT DIAGRAM OF ASSIGN MARINE (FIGURE A-1)

Activity Number:	A-0
Activity Name:	ASSIGN MARINE
Control Name:	Official Policies, Marine Corps Orders
Output Name:	Billet Assignments, Generated Reports
Input Name:	Staffing Goals, Enlisted Marine Force
Mechanism Name:	MOS Monitors, Model Manager
Activity Definition:	This is the main process of the whole EAM model. Through this process an Enlisted Marine will be assigned to the appropriate billet.

C. ARROW DEFINITIONS

Arrow Name	Arrow Definition
Assigned Billets	Billets that are currently held or that have been projected to be held during this particular run.
Available Marines	These are Marines that are first identified by EAM as being available to move.
Available Quotas	This file is compared to the Available Marines file to determine Eligible Marines. It contains the Assignment Quota Records, Assignment Quota Trailer Record, Deployment data Records and Staffing goal Records.
Billet Assignments	These are the assignments that are recommended by EAM. Both those who were assigned and who were not.
Correct Dictionary	Dictionary that can be used to make and EAM run.
Eligible Marines	These Marines are eligible for assignment to a billet that is generated by EAM.
Enlisted Marine Force	This consists of all enlisted Marines and is an extract of the Headquarters Master File. This data is run through an MOS conversion process. This file is also updated by the EAM process output after comparison to the Orders Database.
Generated Reports	Assignment Reports, Assignment Summary, and the Orders File
Marine Corps Orders	Official Marine Corps Orders (MCO)
Model Manager	Person who runs the EAM
MOS Family	The current MOS family being processed.
MOS Monitors	Military Occupational Specialists
Non Available Marines	These Marines do not meet the rules for moving. They become the 'future' pictures of what EAM will see as the currently held jobs.
Official Policies	Official Policies from the offices of Manpower Management
Staffing Goals	The goals that the Marines Corps would like to staff. The Enlisted Staffing Goal Model generates these goals or quotas. (by MCC, by MOS, by Grade)
UnAssigned Billets	These are billets that EAM does not assign.
Validated dictionary	Dictionary that has been validated. Rules that are valid and used for determining availability and eligibility.

Table 1: IDEF0 Arrow Definitions

D. NODE A-0: DECOMPOSITION OF CONTEXT DIAGRAM - ASSIGN MARINE (FIGURE A-2)

Activity Number: A0
Activity Name: ASSIGN MARINE
Input Name: Staffing Goals
Input Name: Enlisted Marine Force
Control Name: Marine Corps Orders
Control Name: Official Policies
Mechanism Name: MOS Monitors
Mechanism Name: Model Manager
Output Name: Billet Assignments

Output Name: Generated Reports
Activity Definition: This is the main process of the whole EAM model. Through this process an Enlisted Marine will be assigned to the appropriate billet.

Activity Number: A1 (Figure A-3)
Activity Name: Dictionary Processor
Control Name: Marine Corps Orders
Control Name: Official Policies
Mechanism Name: Model Manager
Mechanism Name: MOS Monitors
Output Name: Validated dictionary
Activity Definition: This process validates the dictionary. If the dictionary is true this step will also pre-process assorted tables in the EAM.

Activity Number: A1.1
Activity Name: Check Dictionary
Control Name: Official Policies
Control Name: Marine Corps Orders
Mechanism Name: Model Manager
Mechanism Name: MOS Monitors
Output Name: Correct Dictionary
Activity Definition: This process checks to ensure the dictionary is valid in accordance with the rules of EAM.

Activity Number: A1.2
Activity Name: Pre-Process Assorted Tables
Input Name: Correct Dictionary
Output Name: Validated dictionary
Activity Definition: This processes different tables in EAM, mainly concerned with tourtype, tour control factors, and family opmove limits.

Activity Number: A2 (Figure A-4)
Activity Name: Select Available Marines
Input Name: Enlisted Marine Force
Control Name: Validated dictionary
Output Name: Non Available Marines
Output Name: Available Marines
Activity Definition: The process that selects available Marines for assignment. They are classified as movers.
Activity Number: A2.1
Activity Name: Check MOS, GRADE, MCC Substitution
Input Name: Enlisted Marine Force

Control Name:	Validated dictionary
Output Name:	Non Available Marines
Output Name:	Available Marines
Activity Definition:	Checks for invalid MOS MCC GRADE. These Marines are classified as available.

Activity Number:	A2.2
Activity Name:	Determine Assignment Category
Input Name:	Available Marines
Input Name:	Non Available Marines
Control Name:	Validated dictionary
Output Name:	Available Marines
Output Name:	Non Available Marines
Activity Definition:	This process checks a Marines Assignment Category. This is a process that compares the Marines against the Mandatory EAM properties in order to generate an Availables file.

Activity Number:	A2.3
Activity Name:	Determine Availability
Input Name:	Available Marines
Input Name:	Non Available Marines
Control Name:	Validated dictionary
Output Name:	Non Available Marines
Output Name:	Available Marines
Activity Definition:	Further checks are made of the Marines to determine if they are eligible to be reassigned. Further internal controls are added for EAM.

Activity Number:	A3 (Figure A-5)
Activity Name:	Generate Available Quotas
Input Name:	Non Available Marines
Input Name:	Staffing Goals
Control Name:	Validated dictionary
Output Name:	Available Quotas
Activity Definition:	This process performs many things to determine the available quotas that the Marine Corps need to be filled. It extracts quotas for the draw MCC's and deployment Schedules for the deployment MCC's. It also contains the staffing goals.

Activity Number:	A3.1
Activity Name:	Determine On-Board Future Population
Input Name:	Non Available Marines

Control Name:	Validated dictionary
Output Name:	Assigned Billets
Activity Definition:	This process looks at the non movable Marines and formulates a 'future picture' for comparison against the ESGM for determining quotas.

Activity Number:	A3.2
Activity Name:	Determine Target Goal
Input Name:	Assigned Billets
Input Name:	Staffing Goals
Control Name:	Validated dictionary
Output Name:	Available Quotas
Activity Definition:	This process compares the ESGM with the projected Assigned billets to generate the quota's that are required to meet the ESGM.

Activity Number:	A4 (Figure A-6)
Activity Name:	Select Eligible Marines
Input Name:	Available Quotas
Input Name:	Available Marines
Control Name:	Validated dictionary
Output Name:	Available Quotas
Output Name:	Eligible Marines
Activity Definition:	This process compares the available Marines with the available quotas to determine the eligible Marines.

Activity Number:	A4.1
Activity Name:	Determine MOS Family
Input Name:	Available Quotas
Control Name:	Validated dictionary
Output Name:	Available Quotas
Output Name:	MOS Family
Activity Definition:	This process determines the type of run that EAM is running. An OVERSEAS or a CONUS run.

Activity Number:	A4.2
Activity Name:	Determine Eligible Marines
Input Name:	MOS Family
Input Name:	Available Marines
Control Name:	Validated dictionary
Output Name:	Eligible Marines
Activity Definition:	This process takes the available Marines and processes them by family according to the dictionary rules to determine eligibility.

Activity Number:	A5 (Figure A-7)
Activity Name:	Assign Eligible Marines
Input Name:	Eligible Marines
Input Name:	Available Quotas
Control Name:	Validated dictionary
Output Name:	Billet Assignments
Output Name:	UnAssigned Billets
Activity Definition:	This process assigns the eligible Marines. This is solved MOS family by MOS family.

Activity Number:	A5.1
Activity Name:	Achieve Maximum Quota Fill
Input Name:	Eligible Marines
Input Name:	Available Quotas
Control Name:	Validated dictionary
Output Name:	Eligible Marines
Output Name:	Available Quotas
Activity Definition:	This process is subject to the mandatory eligibility constraints determined in Select Eligible Marine Process.

Activity Number:	A5.2
Activity Name:	Max Desirable Assignment Characteristics
Input Name:	Available Quota
Input Name:	Eligible Marines
Control Name:	Validated dictionary
Output Name:	Eligible Marines
Output Name:	Available Quotas
Activity Definition:	This process uses one to six policy optimization algorithms. They are designed to min MOS/Grade substitution, min tour sequence levels, and min desirable property levels, MAX MCC preferences, min PCS mileage costs, max reassignment desirability.

Activity Number:	A5.3
Activity Name:	Max Reassignment Desirability
Input Name:	Eligible Marines
Input Name:	Available Quotas
Control Name:	Validated dictionary
Output Name:	UnAssigned Billets
Output Name:	Billet Assignments
Activity Definition:	This is called the Advanced Assignment Algorithm. It consists of eight separate optimizations. The user may select any number of these optimizations.

Activity Number:	A6 (Figure A-8)
Activity Name:	Generate Reports
Input Name:	UnAssigned Billets
Input Name:	Billet Assignments
Mechanism Name:	Model Manager
Output Name:	Generated Reports
Activity Definition:	This process generates all of EAMs reports. The contractor can control these reports.

Activity Number:	A6.1
Activity Name:	Generate Assignment Reports
Input Name:	UnAssigned Billets
Input Name:	Billet Assignments
Mechanism Name:	Model Manager
Output Name:	Generated Reports
Activity Definition:	This process generates a report that contains the individuals who were recommended for assignment and those who were not. It generates orders that can later be processed if approved.

Activity Number:	A6.2
Activity Name:	Generate Available Billet Reports
Input Name:	UnAssigned Billets
Input Name:	Billet Assignments
Mechanism Name:	Model Manager
Output Name:	Generated Reports
Activity Definition:	This report generates the billets that EAM found available.

Activity Number:	A6.3
Activity Name:	Generate Assignable Marines Report
Input Name:	Billet Assignments
Mechanism Name:	Model Manager
Output Name:	Generated Reports
Activity Definition:	This report generates a list of those Marines that were found assignable by the EAM.

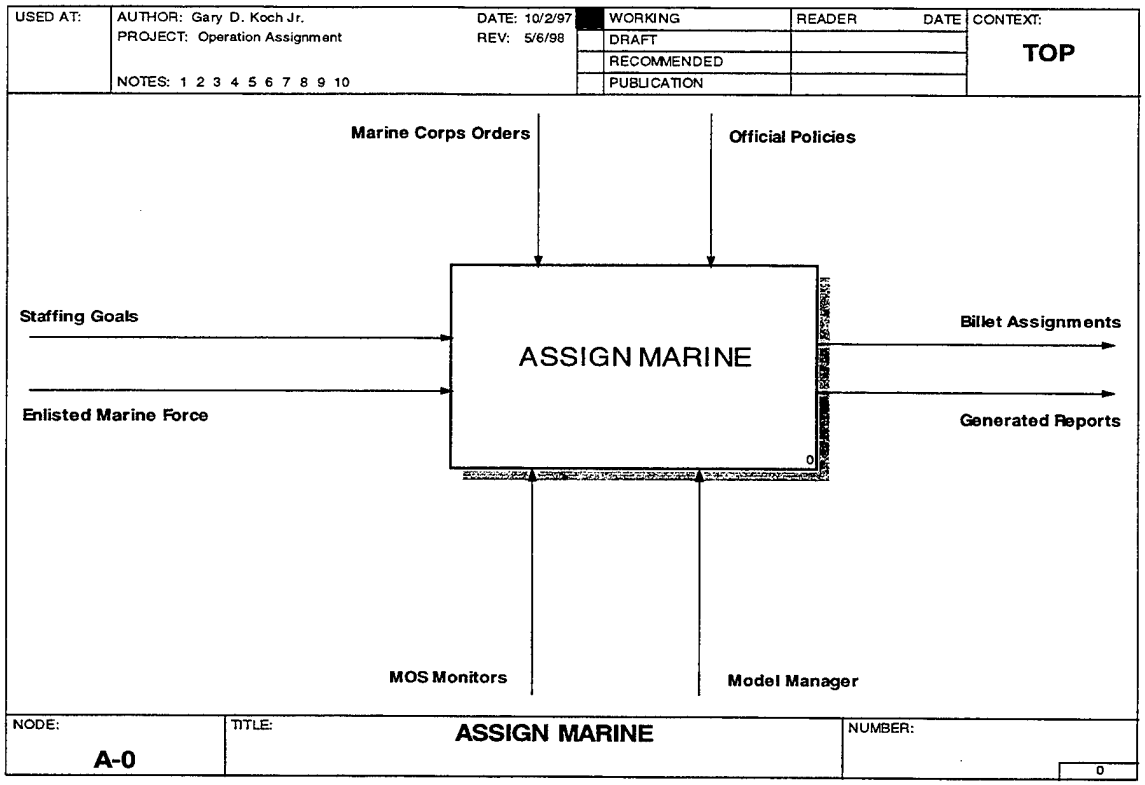


FIGURE A-1. Assign Marine Context Diagram

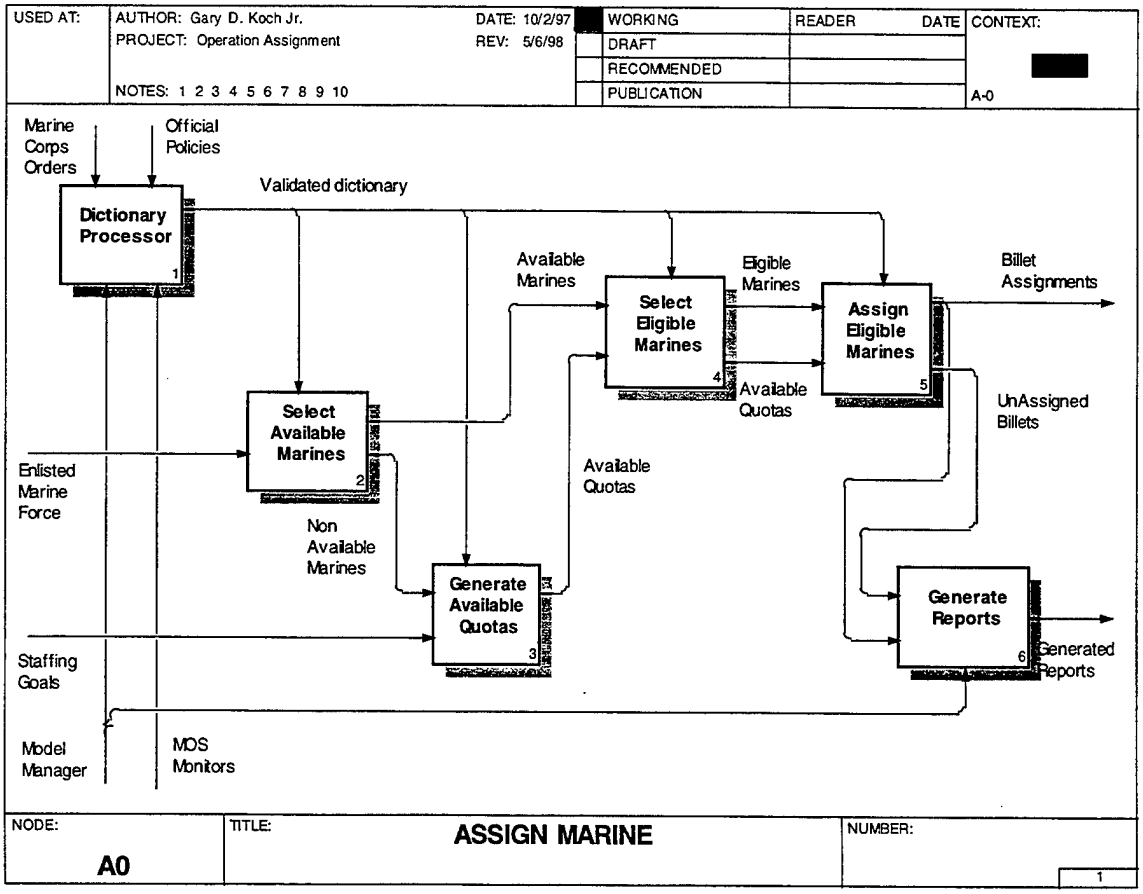


FIGURE A-2. Decomposition of Context Diagram

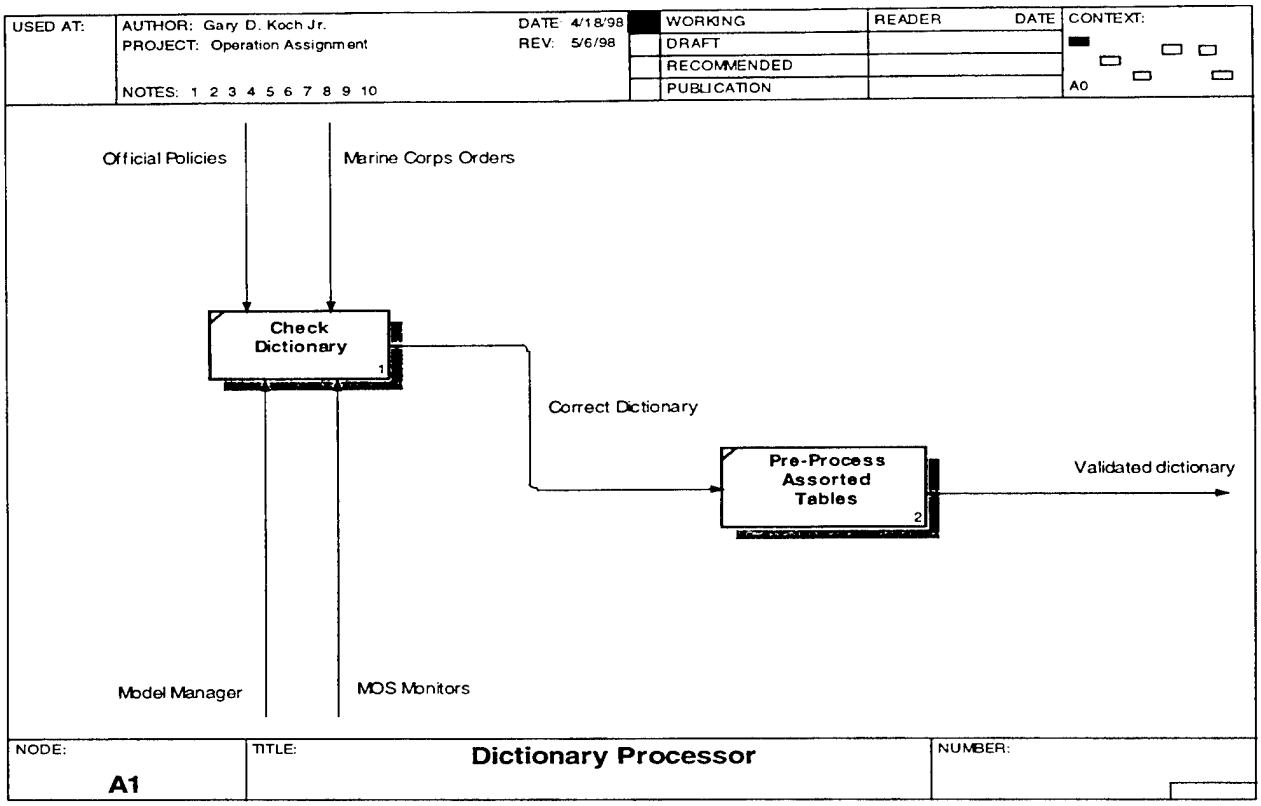


FIGURE A-3. Dictionary Processor

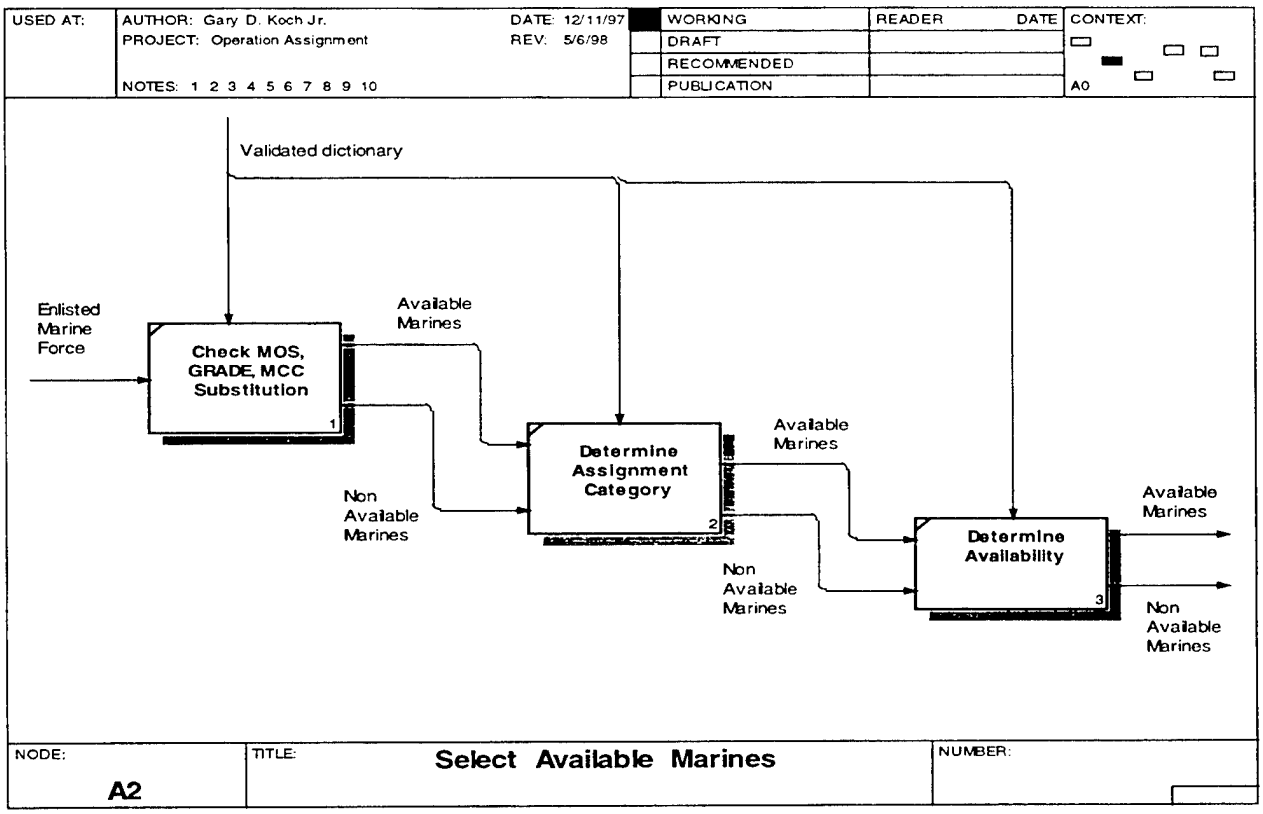


FIGURE A-4. Select Available Marines

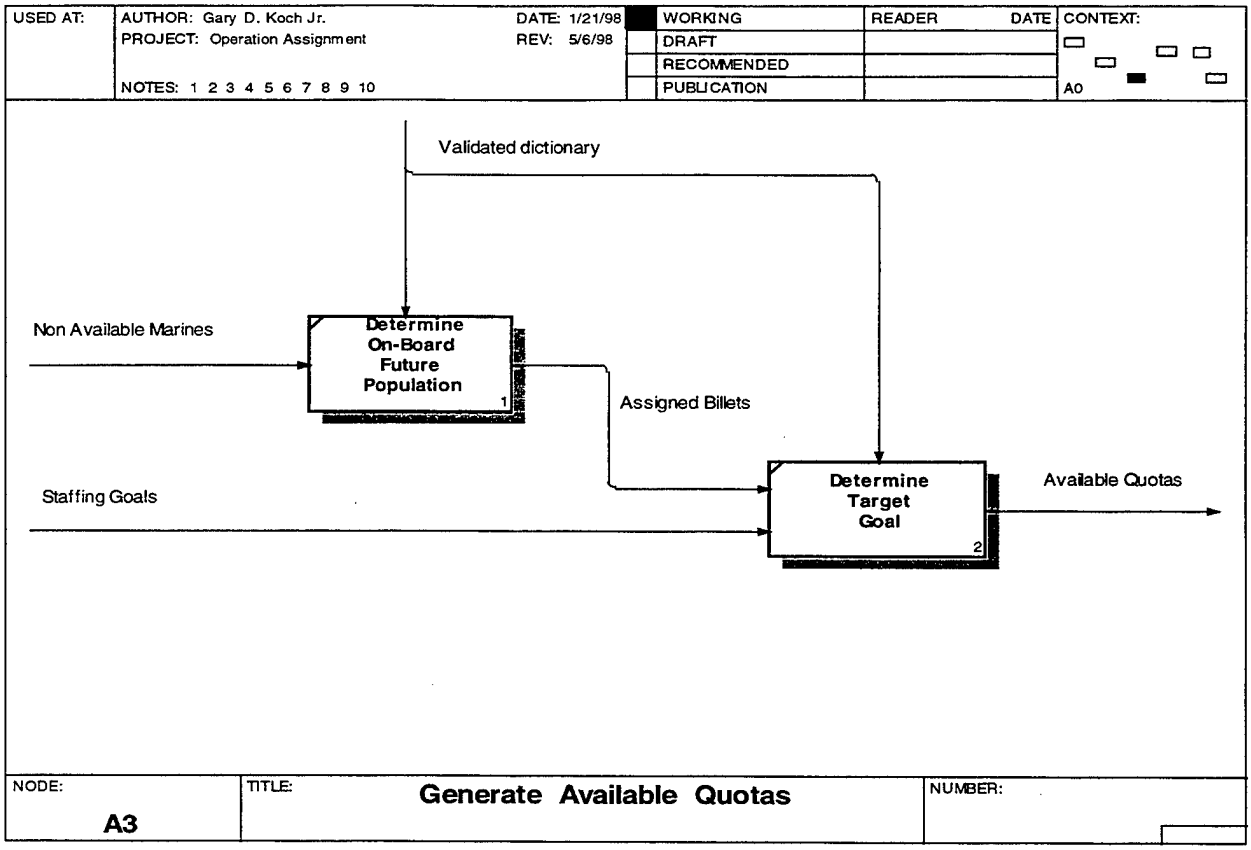


FIGURE A-5. Generate Available Quotas

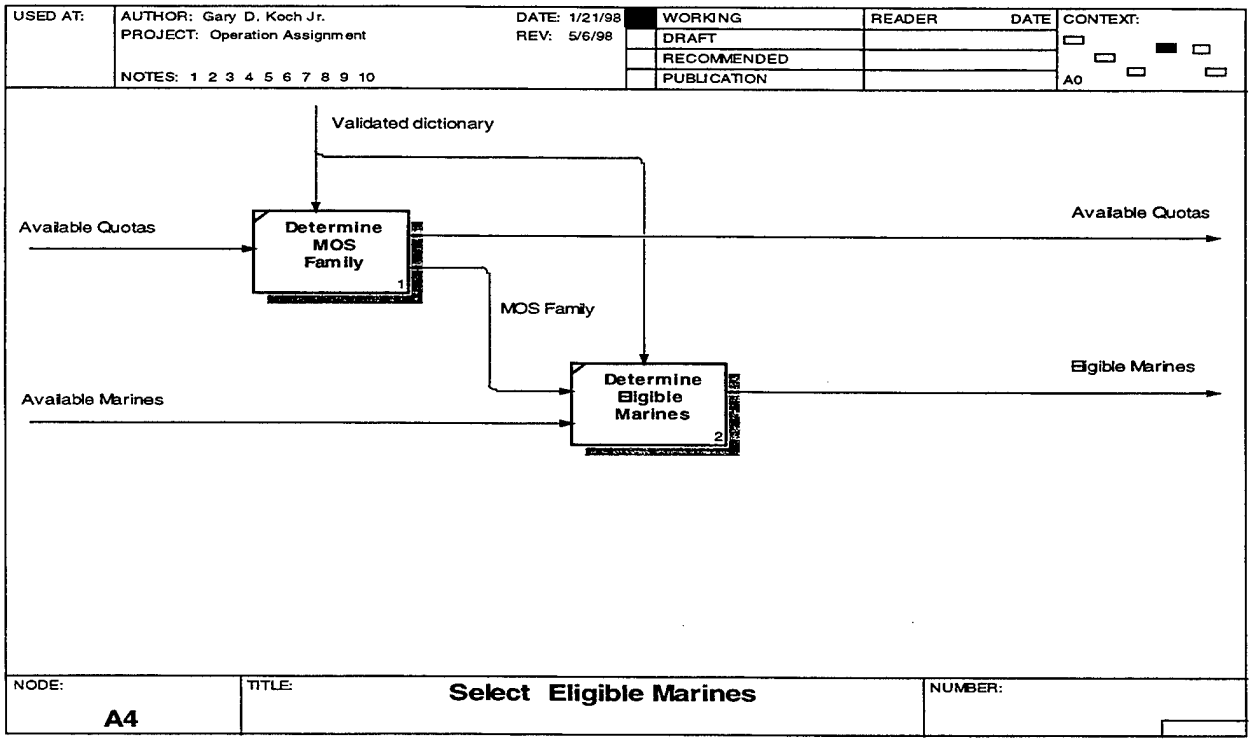


FIGURE A-6. Select Eligible Marines

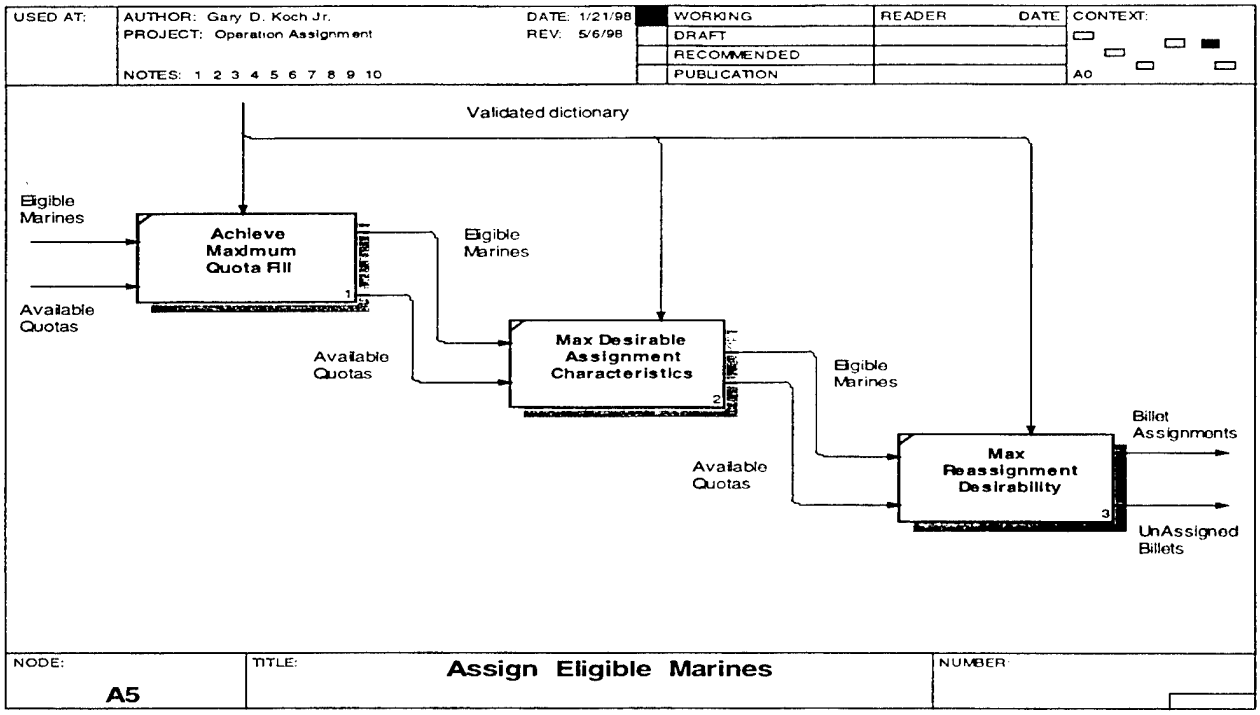


FIGURE A-7. Assign Eligible Marines

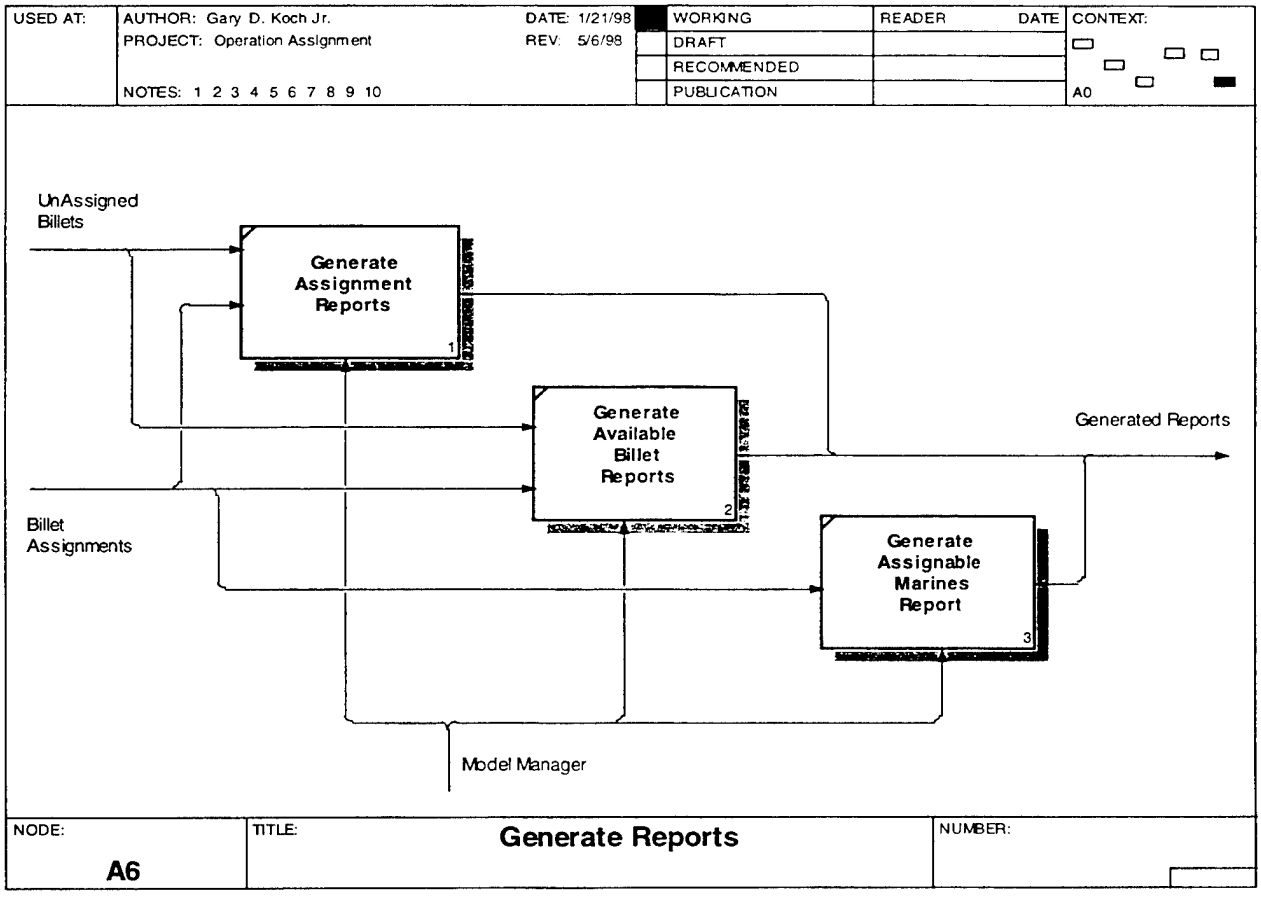


FIGURE A-8. Generate Reports

APPENDIX B: TABLE DEFINITIONS

The EAM Dictionary is the source of flexibility in the model's operation but also the cause of much consternation. It contains all of the assignment rules that the model uses to make its recommendations. The basic model program rarely changes, and the dictionary provides the means of making needed updates and changes. The maintenance of the dictionary is perhaps the most important responsibility of the model manager. The dictionary consists of (16) sixteen tables. The manager has the ability to make changes to any table that has not yet been used in a run. The most convenient means of making changes to a table is to make a new copy of the desired table and then make the corrections. This inability to alter tables previously used in an EAM run serves as a safety precaution. This ensures that the model manager has a copy of the "old" dictionary tables as well as the "new" dictionary tables. Dictionary updates should be made prior to assignment runs to minimize the chance of error. The following pages cover the organization and maintenance of the dictionary [USMC 1992].

A. ORGANIZATION

The dictionary is comprised of many interrelated tables. Each time a table is updated, a new dictionary must be created to incorporate the new table.

1. Properties Table

Virtually any information contained in the Headquarters Master File (HMF) can be manipulated for EAM purposes. MCTFS data is screened by the model and can be set up to enhance assignment concerns, or preclude certain assignment types that have been deemed undesirable. The MCTFS data is "strung" together to form "properties". These "properties" clearly define assignment rules for the model. A property is considered a single or complex set of true/false tests that compare data items from the EAM personnel extract file with each other or with the user-supplied constants. A complex set is a set of single tests strung together by logical operators. There are (2) two types of properties within the Properties table: Fundamental properties and Derived properties. Fundamental properties are properties defined directly by the raw MCTFS data input. No other properties can be used to define a fundamental property. Properties defined by other properties are called derived properties. These properties have other, often-fundamental properties, as their components. For example, to create a property to identify all male Marines with the MOS of 0311, the manager would create three properties. The first two properties titled 'MALE' and 'MOS0311' would read: MALE SEX EQ M and MOS0311 PMOS EQ 0311. 'Sex' and 'PMOS' are input fields off of the MCTFS extract. The third property, a derived property, would tie these two fundamental properties together. Titled 0311MALE, it would read 0311MALE AD MALE MOS0311. 'AD' and 'EQ' stand for 'and' and 'equal' respectively.

New properties can be created and given any name so long as the length does not exceed (8) eight characters. Existing properties can be strung together to give them new meaning and form new properties. There are (23) twenty-three mandatory classification properties that must be defined in order for the model to run. Other properties have grown from this list, or have been created to meet the monitor needs over the years. There are now over (1000) one thousand properties currently defined.

2. Property Optimization Table

As stated earlier, the EAM classifies Marines first, and then classifies MCC's in order to find a match. The Property Optimization table is where properties are utilized to provide specific assignment criteria to a particular MCC or tour type. These criteria can relate to the entire Marine Corps, an MOS family, a particular MOS within a family, and even a grade within an MOS. The specificity of the criteria is of great importance. The more specific criteria overrule the less specific criteria as applied in the same case. If a quota satisfies specifications for more than one levels set, the quota will be associated with the most specific (i.e., having the largest specificity rank) levels set. Specificity ranking weights, by and within data type, are as follows:

	MCC	GRADE	MOS	SBI
Specific	18	2	6	27
Range	9	1	3	
All	0	0	0	

Table 2: Specificity Criteria

Within each data type, ranking weight increases with specificity. For example, a weight of 18 is given for specifying an MCC, 9 for specifying a tour category ("range" of MCC's), or 0 for accepting any MCC. If a Marine satisfies all three requirements, the EAM will assign him to a particular MCC because that is more specific. The model manager must ensure that the users' intent is not overruled by the more specific prerequisite. Levels within the table specify mandatory and desirable characteristics for Marines to be assigned to quotas and an MCC. Level numbers groups these prerequisites in sets. Level numbers are integers from 0 to 6, with level 0 prerequisites being mandatory. It should be noted that "mandatory" here does not refer to the (23) twenty-three mandatory properties discussed earlier. Levels 1-6 prerequisites are desirable, but not mandatory. LEVEL 1 is the most desirable level, LEVEL 2 the next most desirable, and so on to LEVEL 6 which is the least desirable. To satisfy the prerequisites for a level, data items on a Marine's personnel record must satisfy all properties listed for that level. If a property or group of properties is listed on level 0, the mandatory level, then a Marine must fully meet all the prerequisites to be assigned to that MCC or tour group.

The following is an example that further explains the above:

A quota is generated for a 0369 Staff Sergeant at MCC 121. If the most specific entry in this table reads:

Spec Index	MCC Spec	MOS Spec	SBI	Low Grd	High Grd	Cost Center	Lvl	Prop1	Prop2
25	121	0369		2	4		0	AVLDNRC	SNCOGRAD
25	121	0369		2	4		1	SINGLE	

Table 3: Specificity vs. Level

With the above stipulations, a Marine with a MOS of 0369 would have to satisfy the LEVEL 0 properties.

Therefore, he would have to be available for a dependent not restricted CONUS tour and be a graduate of the SNCO academy. If he does not satisfy both requirements, he is ineligible and is dropped from consideration for assignment to this MCC. After finding Marines that meet these mandatory requirements, the EAM would attempt to "optimize" the assignment and proceed to LEVEL 1, which contains a desirable property that requires the Marine to be single. The model will then attempt to find a single Marine, but it is not bound to do so.

If the property "NOFILL" is listed as a prerequisite, no recommended assignment will be made to that particular MCC or tour type. This property is used frequently to avoid making recommendations to deployed (UDP) units. Another method of precluding assignments is to use the properties "CAREERST" and "FRSTTERM" in LEVEL 0 as mandatory properties. Since nobody can satisfy both requirements, no recommendations are made. In summary, to effect the assignment to a particular MCC or tour type, the Property Optimization table is interfaced. Any property can be applied as mandatory or desirable prerequisites. The level of specificity will determine the priority of fill for the EAM. If property A is applied to a tour type for a specific MOS, but property B is applied to the same MOS in a specific MCC within that tour type, property B will take priority. The Property Optimization table is updated monthly in accordance with the deployment schedule, which will be covered next. As the units enter their "lock-on"

periods, the EAM is precluded from making assignments to that unit until it returns from the UDP.

The EAM makes assignment recommendations with respect to a (6) six-month projection. Due to this fact it is necessary to track the deployment schedule in order to determine staffing availability of units participating in the Unit Deployment Cycle. Marine Expeditionary Units (MEU's) do not receive EAM recommendations (7) seven months prior to deployment ("lock-on" plus one month). All other non-MEU designated units who deploy are precluded at the (4) four-month mark.

MMEA-12 maintains and updates the deployment schedule and provides the model manager with a copy. A spreadsheet has been set up in order for the EAM manager to effectively track the UDP cycle. The deployment schedule is incorporated into EAM through the Properties Optimization table of the dictionary. Updates are made once a month at a minimum and are the most common dictionary change.

In determining the units to preclude from EAM assignment recommendations, the model manager will use a straight-line column method. The following steps are used to select those units to receive the "NOFILL" property designator in the LEVELS portion of the dictionary (see Table 4):

1) Print out the updated version of the deployment schedule, review it and draw a straight-line column under the month that coincides with the run title.

2) If the line falls on a respective unit's lock-on period, a "*" in that unit's row, or it's actual deployment cycle, a "D" in that unit's row, that unit should be precluded from EAM assignment recommendation.

3) Those units who fall in the category of the above are precluded by then performing a Property Optimization table property restriction modification, placing the "NOFILL" property in the mandatory line for that particular unit. The "NOFILL" property is an allocation property that blocks all inbound assignment recommendations.

4) Likewise, if a previously precluded unit is no longer deployed, or not designated under the straight-line method, it should taken off the precluded status by again modifying the Property Optimization table of the dictionary.

5) The following example applies:

	1997													1998										
	J	F	M	A	M	J	J	A	S	O	N	D		J	F	M	A	M	J	J	A	S	O	N	D
V31											*	*		*	*	D	D	D	D	D	D	*			
V32	*	D	D	D	D	D	D	*																	
V33	D	D	D	D	D	*																			

Table 4: Deployment Matrix

In the above example, assume the first run of the month is the overseas run for November 97. A line column is drawn under November. Of the three units listed, one is entering the lock-on period. The following dictionary change listed in Table 5 would be made in the Property Optimization table:

Spec Index	MCC Spec	MOS Spec	SBI	Low Grd	High Grd	Cost Center	Lvl	Prop1	Prop2
25	V31			2	9		0	NOFILL	

Table 5: Property Optimization Table

This essentially blocks V31 from receiving any EAM recommendations until it is removed. When adding the 'NOFILL' property to the mandatory property level for V31, do not delete any other properties defined to V31. This will enable you to simply remove the 'NOFILL' property once V31 comes off of deployment and maintain the previous property settings. Using the same example, it is noted that V31 drops off the "NOFILL" listing in October 1998. They have been back from deployment for a month, and thus are eligible to receive EAM recommendations. The 'NOFILL' property can now be removed from V31's listing in the Property Optimization table.

3. MOS Substitution

This portion of the dictionary, divided into the MOS Family and MOS Substitution tables, also provides a great deal of flexibility in the assignment process. Obviously, the best match for any quota is "on grade, on MOS", but the EAM cannot always find an exact match. By allowing the model to find an acceptable, defined substitution within another grade or another MOS, a larger number of "movers" is provided for the monitors to choose from. Recently, all MOS/Grade substitutions were removed from the model based on duplication of functionality between the ESGM and EAM models. The ESGM model evaluates the ASR and attempts to generate staffing goals on grade, on MOS. If it can't, it allows for substitutions. Previously, EAM would evaluate the staffing goal and attempt to generate an on MOS/on grade assignment. If it could not, it too would allow for substitutions. In this process, the ESGM might see an Authorized Strength Report (ASR) for an E7 / 0151, and because of shortages, generate a

staffing goal for an E6/0151. Now the EAM sees a staffing goal for an E6 / 0151, but since it can't find any E6 / 0151 movers, it assigns an E5 / 0151 instead. Therefore, a sergeant will end up filling a gunnery sergeant's billet.

The first sub-table in this table is the MOS Family Table. MOS's are grouped into distinct families; the grade and MOS substitutions must stay within families. There are currently (73) seventy-three families defined. The second section is called the MOS Substitution Table. Here, the acceptable grade and MOS substitutions are listed in order of priority. There are several possible configurations detailed in the Decision Systems Automated Services (DSAI) users' guide, but the substitutions are minimized to such an extent by monitor input that the basic format currently reflected in the dictionary is sufficient. The reasoning behind this minimization lies in the fact that the monitors tend to restrict the allowable substitutions due to the specialization factor. The belief here is that the critical nature of today's assignments requires that billets be filled on grade, on skill. The Monitor's Guide provides an example of the consequences of overly restrictive substitution criteria. It should be defined in the Family Organization Table in order to be utilized in the MOS Substitution Deck.

4. MCC Definition Table

This section of the dictionary is perhaps the most frequently used. In this table, MCC's are defined as to a tour type, dependents status, tour sequence group, priority of fill, cost center, future tour control factor, and geo-location code. All new MCC's must be listed and/or defined in this section before they can be considered by the EAM.

Similar MCC's can be grouped together to form a tour sequence group. For example, "/CB" is a tour group that contains the various CONUS Barracks locations. Tour categories can be beneficial when assigning and utilizing related criteria. Rather than process every MCC within a similar group, the criteria can be applied to the tour category and all the attached MCC's are affected. Tour categories can be created by the model manager if the need arise.

Tour types must be assigned to each MCC or tour type. Legal tour type entries:

"O" - Overseas
"C" - CONUS
"X" - Non-filled (EAM will never assign to MCC's with this tour type)

Dependent Status refers as to whether dependents are allowed. Legal statuses type entries:

"U" - Dependents unrestricted
"R" - Dependents restricted

Each MCC must be specified with a priority from 1-10. This priority number controls the order in which the EAM distributes personnel shortages among quotas. A priority of 1 will be filled before that of 2, and so on. If not all quotas within a priority can be filled, the EAM will "fair share" the shortage among all quotas of that priority. The cost center name associated with an MCC must be defined in the Cost table of the dictionary. Cost centers are used in the model to optimize PCS mileage costs. If a MCC is defined in the MCC table, it must be assigned a Future Tour Control Factor (FTCF) and a geo-location code.

The standard tour length at a CONUS barracks is (36) thirty-six months. It has been deemed that billet MOS (BMOS) 8152 (guards) should be restricted to a (24) twenty-four month tour in order to get them to the FMF before separation. Grades E2-E4 of BMOS 8152 have been designated a (24) twenty-four month tour in the Tour Length Classification Table. This is more specific than a FTCF code and is given priority by the model. Marines in the grade of E2-E4 serving in a BMOS of 8152 at an MCC contained in the tour category "/CB" (CONUS Barracks) will be selected by the EAM at the two year mark of his tour. All others will remain for a third year in accordance with the less specific criteria of the FTCF Deck.

When the decision is made to assign a new MCC, the process is relatively simple for the correlative EAM updates. The new MCC must first be defined in the MCC Definition Table. The model manager has to decide what tour type, if any, to put the MCC in. This decision is made with input from the respective monitor(s). Other decisions that must be made concern the following: tour category assigned to (CONUS or Overseas, dependents restricted or unrestricted), fill priority, associated cost center. The final step in entering the new MCC into the dictionary is defining it in the Tour Control Factor Deck, and also in the GEO/MCC Deck. Geographic location codes (geo codes) are used in the advanced assignment and derivation process. The model attempts to make advanced assignments in CONUS for the return of Marines going overseas. Recommendations are based on detaching MCC location and dependent location. Originally mandated by Congress as an economical tool, the process is not presently used to a great degree. This fact is true due to the emphasis placed upon the actual

requirement, the "needs of the Marine Corps", which overrides cost concerns when assigning personnel. There are currently (16) sixteen geo codes defined.

5. Tour Sequence Table

Tour sequencing is a part of the optimization process and the Tour Sequence Table allows the manager to determine whether Marines will be assigned from one tour category to another. Tour categories and tour sequences are weighted according to priority or are precluded. For instance, it is currently stipulated that no "back-to-back" B-billets be authorized. The tour sequence from the various B-billets is zeroed out when matched with gaining tour categories. In many instances, the FMF tour categories are given higher priority. However, the tour sequence from FMF to FMF should probably be a lower priority. The associated disadvantage of tour sequencing is that the rules apply to all MOS's. There is not a means to specify a tour sequence for an MOS or grade so the rules determine the best sequence for the majority of the MOS's. As an example:

	/BF	/CB	/CD	/CE	/CF	/CH	/CI	/CN
/CF	0	4	4	0	0	0	2	2

Table 6: Tour Category Priorities

Tour category /CF is the sequence "from", while tour categories /BF to /CN are the gaining or sequencing "to" tour categories. A tour category with a priority of "2" receives consideration before a category 4. Tour categories of "0" are precluded from sequencing.

6. State/County Code

State/County codes are used to associate cost centers with a Marines first dependent's state and county location, or his current state location. This table is seldom interfaced or updated.

7. Advanced Assignment Table

This table is not currently used, but if desired, it can be utilized to make follow-on assignments for Marines upon their tour completion at future MCC.

8. Cost Table

The Cost Table is used during Cost Optimization, if cost is being optimized. Whether or not cost optimization is utilized, all Cost Centers and its respective distance from all other cost centers must be identified here to be valid. To add a Cost Center, open a new copy and in the Edit menu option, select 'Insert Cost Center.' The table will automatically be formatted when the new cost center's name is entered. The model manager must then enter the distances between the new and old centers.

9. Exceptional Marine Classification Table

This table is used to further identify exceptional Marines and to fine tune assignments. The manager must enter the classification type (C = classification E = exception), MCC Spec, RUC Spec, MOS Type (B = billet, P = primary, and T = training), MOS Spec, Low Grade, High Grade, and the properties associated with the entry. For

example, if the ESGM has identified that a 8512 B-billet at MCC 009 is to be filled by a 4066 sergeant, and it is known that all other 4066's at MCC 009 are filling primary billets, the following entry would be entered:

Class Type	MCC Spec	RUC Spec	MOS Type	MOS Spec	Low Grade	High Grade	Prop 1	Prop 2....
C	009		P	4066	2	9	XXX	XXX
E	009		B	4066	5	5	YYY	YYY

Table 7: Exceptional Marine Classification Entry

APPENDIX C: BASIC CODE

A. CODE FOR OPERATION OF EAM

1. Function CreateDemand()

```
Public Function CreateDemand()
   'This function will step through the table Demand
   'and create a demand file.  This function uses a query on the
   'tble non movers and sequentially steps through the Marines and
   'determines which job that they are filling.  Once this is done
   'the finished table is the actual demand for this particular run.

   Dim db As Database          'EAM database
   Dim rst As Recordset        'tblDemand recordset
   Dim rst1 As Recordset       'tblNonMovers recordset
   Dim strGrad As String       'Keeps track of Marines grade.
   Dim strMCCOld As String     'Saves previous record MCC.
   Dim strMOSOld As String     'Saves previous record MOS.
   Dim strBM As String         'Used for BookMark.
   Dim booIn As Boolean        'Set true if next Marine MCC is the same as
                               'current MCC being checked (sorting logic).
   Dim booDup As Boolean       'Set true if same MCC but different grade (sorting logic).
   Dim varRetVal As Variant    'Status bar
   Dim lngJ As Long            'record count
   Dim lngI As Long            'counter

   '--------------------------------------------------------------------------------
   Forms!frmPreprocessingSwitchboard!txtMsg = "Working....."
   Set db = CurrentDb()
   Set rst = db.OpenRecordset("tblDemand", dbOpenTable)          'Open tblDemand
   Set rst1 = db.OpenRecordset("qryAllMarines", dbOpenSnapshot)
   booIn = False

   rst1.MoveLast               'Ensure snapshot loaded
   rst1.MoveFirst
   lngI = 0
```

```
lngJ = rst1.RecordCount
varRetVal = SysCmd(acSysCmdInitMeter, "Updating Demand File.....", lngJ)


Do Until rst1.EOF          'Loop through NonMovers only once.
rst.MoveLast               'Ensure Table loaded
rst.MoveFirst
strBM = rst.Bookmark       'Set Bookmark at first MCC group.

    Do Until rst.EOF       'Loop through MCC MOS jobs to determine quotas

        booDup = False     'set for tracking last MCC of BMK group

        'Check demand against Nonmovers to determine quotas.
        If rst1!PMOS = "0000" Then     'If no PMOS use BMOS
          If rst!MCC = rst1!MCC And rst!MOS = rst1!BMOS Then
            strGrad = "E" & rst1!grd   'Correcting grade column
            strMOSOld = rst1!BMOS      'keep track of this MOS for other grades
            strMCCOld = rst1!MCC       'keep track of this MCC for other grades

          'Update the demand file.
            rst.Edit
            If rst1!grd = 2 Then
             rst![E2] = (rst(strGrad) - rst1!Duplicates)
            ElseIf rst1!grd = 3 Then
             rst![E3] = (rst(strGrad) - rst1!Duplicates)
            ElseIf rst1!grd = 4 Then
             rst![E4] = (rst(strGrad) - rst1!Duplicates)
            ElseIf rst1!grd = 5 Then
             rst![E5] = (rst(strGrad) - rst1!Duplicates)
            ElseIf rst1!grd = 6 Then
             rst![E6] = (rst(strGrad) - rst1!Duplicates)
            ElseIf rst1!grd = 7 Then
             rst![E7] = (rst(strGrad) - rst1!Duplicates)
            ElseIf rst1!grd = 8 Then
             rst![E8] = (rst(strGrad) - rst1!Duplicates)
            ElseIf rst1!grd = 9 Then
             rst![E9] = (rst(strGrad) - rst1!Duplicates)
            End If
            rst.Update

            DoEvents
            varRetVal = SysCmd(acSysCmdUpdateMeter, lngI)
```

```
        lngI = lngI + 1

        rst1.MoveNext
        If rst1.EOF Then
            rst1.MovePrevious
            GoTo LastLine          'end of Non-Mover file -- Exit
        End If
        booDup = True

    End If
Else
    If rst!MCC = rst1!MCC And rst!MOS = rst1!PMOS Then
        strGrad = "E" & rst1!grd    'Correcting grade column
        strMOSOld = rst1!PMOS       'keep track of this MOS for other grades
        strMCCOld = rst1!MCC        'keep track of this MCC for other grades

        'Update the demand file.
        rst.Edit
        If rst1!grd = 2 Then
         rst![E2] = (rst(strGrad) - rst1!Duplicates)
        ElseIf rst1!grd = 3 Then
         rst![E3] = (rst(strGrad) - rst1!Duplicates)
        ElseIf rst1!grd = 4 Then
         rst![E4] = (rst(strGrad) - rst1!Duplicates)
        ElseIf rst1!grd = 5 Then
         rst![E5] = (rst(strGrad) - rst1!Duplicates)
        ElseIf rst1!grd = 6 Then
         rst![E6] = (rst(strGrad) - rst1!Duplicates)
        ElseIf rst1!grd = 7 Then
         rst![E7] = (rst(strGrad) - rst1!Duplicates)
        ElseIf rst1!grd = 8 Then
         rst![E8] = (rst(strGrad) - rst1!Duplicates)
        ElseIf rst1!grd = 9 Then
         rst![E9] = (rst(strGrad) - rst1!Duplicates)
        End If
        rst.Update

        DoEvents
        varRetVal = SysCmd(acSysCmdUpdateMeter, lngI)
        lngI = lngI + 1

        rst1.MoveNext
        If rst1.EOF Then
```

```
            rst1.MovePrevious
            GoTo LastLine        'end of Non-Mover file -- Exit
          End If
          booDup = True

        End If
      End If

' If we have moved out of the current MCC group the Marine could
' be assigned to a job that is not in this ESGM file.
If rst![MCC] <> rst1![MCC] Then

  'If we just updated then there are no more Marines to check for this MCC.
  If booDup = True Then
    booDup = False
  Else              'If didn't update then record for future use.
    rst.MovePrevious
    If rst.BOF Then    'Check if at the frmBeginning of the file.
      rst.MoveNext
    End If
    'Record rst1.Record this Marine will be looked at later.

    DoEvents
    varRetVal = SysCmd(acSysCmdUpdateMeter, lngI)
    lngI = lngI + 1
    rst1.MoveNext
  End If
  'I am in the next MCC group - reset bookmark.
  If rst![MCC] = rst1![MCC] Then
    rst.Bookmark = strBM
    booIn = True
  Else            'No Marines yet - keep searching MCCs until match with
                  'with the next Marine being looked at.
    Do Until rst![MCC] = rst1![MCC]
      rst.MoveNext
      If rst.EOF And Not rst1.EOF Then

        DoEvents
        varRetVal = SysCmd(acSysCmdUpdateMeter, lngI)
        lngI = lngI + 1

        rst1.MoveNext
        If rst1.EOF Then
```

```
                rst1.MovePrevious
                GoTo LastLine          'end of Nonmover file -- Exit
              End If
              rst.Bookmark = strBM
            End If
          Loop
          strBM = rst.Bookmark
          booIn = True
        End If

      End If

      'This checks for also if I am at the end of an MCC group
      If rst![MCC] <> rst1![MCC] Then

        DoEvents
        varRetVal = SysCmd(acSysCmdUpdateMeter, lngI)
        lngI = lngI + 1

        rst1.MoveNext
        If rst![MCC] <> rst1![MCC] Then
          rst.MoveNext
          strBM = rst.Bookmark
          rst.Bookmark = strBM
          rst1.MovePrevious
        End If
      End If

      'Checking to see if same MOS in the MCC with different grade quotas.
      If Not booIn Then
        If rst1!PMOS = "0000" Then          'Is PMOS blank?  If it is use BMOS.
          If booDup And strMCCOld = rst1!MCC And strMOSOld = rst1!BMOS
Then
            strMCCOld = rst!MCC
          Else
            rst.MoveNext
          End If
        Else
          If booDup And strMCCOld = rst1!MCC And strMOSOld = rst1!PMOS
Then
            strMCCOld = rst!MCC
          Else
            rst.MoveNext
```

```
            End If
          End If
        Else
          booIn = False
        End If

      Loop
      If rst1.EOF Then
        Exit Do 'shouldn't get this far but just in case...
      Else
        DoEvents
        varRetVal = SysCmd(acSysCmdUpdateMeter, lngI)
        lngI = lngI + 1

        rst1.MoveNext
      End If
  Loop
LastLine:
  Forms!frmPreprocessingSwitchboard!txtMsg = "Demand File Complete!"
  rst.Close
  rst1.Close
  varRetVal = SysCmd(acSysCmdRemoveMeter)
  Set db = Nothing
End Function
```

2. Function ReformatDemand()

```
Public Function ReformatDemand()
  Dim db As Database
  Dim rstDemand As Recordset
  Dim rstJob As Recordset
  Dim i As Integer
  Dim strGrade As String
  Dim strCommand As String
  Dim sngRandom As Single
  Dim lngI As Long
  Dim lngJ As Long
  Dim varRetVal As Variant

  Randomize
  Set db = CurrentDb()
```

```
'Create a Demand recordset with just the data we need.
Set rstDemand = db.OpenRecordset("qryDemand", dbOpenSnapshot)
'Create a Job recordset for the quota file.
Set rstJob = db.OpenRecordset("qryJobsFill", dbOpenDynaset)

rstDemand.MoveLast
rstJob.MoveLast

rstDemand.MoveFirst
rstJob.MoveFirst
lngI = 0
lngJ = rstDemand.RecordCount
varRetVal = SysCmd(acSysCmdInitMeter, "Reformating Demand File.....", lngJ)

Do Until rstDemand.EOF

   DoEvents
   varRetVal = SysCmd(acSysCmdUpdateMeter, lngI)
   lngI = lngI + 1

   For i = 2 To 9
      strGrade = "E" & i
      'Debug.Print strGrade, rstDemand(strGrade)
      sngRandom = Rnd()
      If sngRandom < 0.1 Then
         strCommand = "E"
      ElseIf (sngRandom < 0.35 And sngRandom >= 0.1) Then
         strCommand = "P"
      Else
         strCommand = "O"
      End If

      If rstDemand(strGrade) > 0 Then
         rstJob.Edit
         rstJob![Quota] = rstDemand(strGrade)
         rstJob![StaffLevel] = strCommand
         rstJob.Update
         'Debug.Print strGrade, rstJob![MOS], rstDemand![MOS]
         'If i = 9 Then
         rstJob.MoveNext
         'End If
      Else
```

```
            rstJob.Delete
            rstJob.MoveNext
            'Exit For
          End If
        Next i
        rstDemand.MoveNext
    Loop
    rstDemand.Close
    rstJob.Close
    varRetVal = SysCmd(acSysCmdRemoveMeter)
    Set db = Nothing
    Forms!frmPreprocessingSwitchboard!txtMsg = "Reformat Complete!"
End Function
```

3. Function CreateRelationship()

```
Public Function CreateRelationship(strRel As String, strPTab As String, strFTab As
String, strFld As String) As Boolean
    Dim db As Database
    Dim rel As Relation
    Dim rel1 As Relation
    Dim fld1 As Field
    Dim fld As Field

    On Error GoTo CreateRelationship_Err

    Set db = CurrentDb()
    Set rel = db.CreateRelation(strRel, strPTab, strFTab, dbRelationUpdateCascade)

    ' Create field in Relation object.
    Set fld = rel.CreateField(strFld)

    ' Specify field name in foreign table.
    fld.ForeignName = strFld

    ' Append Field object to Fields collection of Relation object.
    rel.Fields.Append fld

    ' Append Relation object to Relations collection.
    db.Relations.Append rel
```

```
    db.Relations.Refresh

    Set db = Nothing

CreateRelationship_Exit:
    Exit Function

CreateRelationship_Err:
    Select Case Err.Number
        Case glrcErrObjectExists
            db.Relations.Delete rel.Name
            Resume
        Case Else
            MsgBox "Error: " & Err.Description & _
            " ( " & Err.Number & " ) "
            CreateRelationship = False
            Resume CreateRelationship_Exit
    End Select

End Function
```

4. Function ArrayCost()

```
Public Function ArrayCost()
    'This function will be used to calculate the cost associated with each
        'Movable Marine as compared to the available jobs from the demand table.

        Dim db As Database
        Dim rstMarine As Recordset
        Dim rstJob As Recordset
        Dim rstJM As Recordset
        Dim rstGrade As Recordset
        Dim intCost As Long
        Dim intPayJ As Long
        Dim intPayM As Long
        Dim strPMOS As String
        Dim strMOS As String
        Dim varGrade As Variant
        Dim varMar As Variant
        Dim varJob As Variant
```

```
    Dim varCost As Variant
    Dim intCount As Integer
    Dim intCountJ As Integer
    Dim intI As Integer
    Dim intJ As Integer
    Dim lngBeg As Long
    Dim lngEnd As Long

    'lngBeg = Timer

    Set db = CurrentDb()
    'Create a Marine Movers recordset with just the data we need.
    Set rstMarine = db.OpenRecordset("qryArrayMovers", dbOpenSnapshot,
dbForwardOnly)
    'Create a Job recordset form the demand file.
    Set rstJob = db.OpenRecordset("qryJobsFill", dbOpenSnapshot, dbForwardOnly)
    'Create a Grade recordset for calculating cost
    Set rstGrade = db.OpenRecordset("GRADE", dbOpenSnapshot, dbForwardOnly)
    'Create a Job_Marine cost table to develop the cost matrix.
    Set rstJM = db.OpenRecordset("tblJob_Marine", dbOpenTable)
    'rstGrade.MoveFirst
    varMar = rstMarine.GetRows(7805)
    varJob = rstJob.GetRows(16200)
    varGrade = rstGrade.GetRows(9)
    rstMarine.Close
    rstJob.Close

    'Ensure we have records.
    intCount = UBound(varMar, 2) + 1
    intCountJ = UBound(varJob, 2) + 1

    'Static CostTable(0 To 7800, 0 To 14000) As Integer

    For intI = 0 To intCount - 1        'Do Until rstMarine.EOF
    DoEvents                            'rstJob.MoveFirst
      If varMar(2, intI) <> 1 Then      'check for E1 if so treat as E2
        intPayM = varGrade(2, (varMar(2, intI)) - 2)
      Else
        intPayM = 1038
      End If
      For intJ = 0 To intCountJ - 1                 'Do Until rstJob.EOF
        intPayJ = varGrade(2, (varJob(3, intJ)) - 2)
```

```
If varJob(4, intJ) = "E" Then            'cost of excepted staffing command
   intCost = -7500
   If varMar(2, intI) <> varJob(3, intJ) Then
      intCost = 1000000000
   End If
   If varMar(1, intI) <> varJob(2, intJ) Then
      intCost = 1000000000
   End If
ElseIf varJob(4, intJ) = "P" Then        'cost of priority staffing command
   intCost = -5000
   If varMar(2, intI) <> varJob(3, intJ) Then
      If varMar(2, intI) - varJob(3, intJ) = -1 Then
         intCost = intCost + 1 * ((intPayJ) - (intPayM))
      ElseIf varMar(2, intI) - varJob(3, intJ) = 1 Then
         intCost = intCost + 2 * ((intPayM) - (intPayJ))
      Else
         intCost = 1000000000
      End If
   End If
Else                                     'cost of a regular job.
   intCost = -2500
   If varMar(2, intI) <> varJob(3, intJ) Then
      If varMar(2, intI) - varJob(3, intJ) = -1 Then
         intCost = intCost + 1 * ((intPayJ) - (intPayM))
      ElseIf varMar(2, intI) - varJob(3, intJ) = 1 Then
         intCost = intCost + 2 * ((intPayM) - (intPayJ))
      Else
         intCost = 1000000000
      End If
   End If
   If varMar(1, intI) <> varJob(2, intJ) Then
      strPMOS = Left(varMar(1, intI), 2)
      strMOS = Left(varJob(2, intJ), 2)
      If strPMOS = strMOS Then
         intCost = intCost + 1 * ((intPayJ) - (intPayM))
      Else
         intCost = 1000000000
      End If
   End If
End If
   If intCost < 900000000 Then
      'CostTable(intI, intJ) = intCost
      rstJM.AddNew
```

```
                rstJM!MarId = varMar(0, intI)
                rstJM!JobId = varJob(0, intJ)
                rstJM!cost = intCost
                rstJM.Update
                rstJM.Move 0, rstJM.LastModified
              End If
                'Debug.Print rstJM!MarId, rstJM!JobId, rstJM!cost
                'rstJob.MoveNext
              intCost = 0
        Next intJ
        If intI = 1800 Then
          rstJM.Close
          DoCmd.TransferText acExportFixed, "spcCost", "tblJob_Marine",
"E:\Eam\text\cost.txt"
          Forms!frmPreprocessingSwitchboard!txtMsg = "Cost text exported!"
          DoEvents
          db.Execute "DELETE * FROM tblJob_Marine;"
          Set rstJM = db.OpenRecordset("tblJob_Marine", dbOpenTable)
        ElseIf intI = 3800 Then
          rstJM.Close
          DoCmd.TransferText acExportFixed, "spcCost", "tblJob_Marine",
"E:\Eam\text\cost1.txt"
          Forms!frmPreprocessingSwitchboard!txtMsg = "Cost1 text exported!"
          DoEvents
          db.Execute "DELETE * FROM tblJob_Marine;"
          Set rstJM = db.OpenRecordset("tblJob_Marine", dbOpenTable)
        ElseIf intI = 5800 Then
          rstJM.Close
          DoCmd.TransferText acExportFixed, "spcCost", "tblJob_Marine",
"E:\Eam\text\cost2.txt"
          Forms!frmPreprocessingSwitchboard!txtMsg = "Cost2 text exported!"
          DoEvents
          db.Execute "DELETE * FROM tblJob_Marine;"
          Set rstJM = db.OpenRecordset("tblJob_Marine", dbOpenTable)
        'ElseIf intI = 5800 Then
         '  rstJM.Close
          ' DoCmd.TransferText acExportFixed, "spcCost", "tblJob_Marine",
"E:\Eam\text\cost3.txt"
          'Forms!frmPreprocessingSwitchboard!txtMsg = "Cost3 text exported!"
          'DoEvents
          'db.Execute "DELETE * FROM tblJob_Marine;"
          'Set rstJM = db.OpenRecordset("tblJob_Marine", dbOpenTable)
        'ElseIf intI = 7803 Then
```

```
        'Set rstJM = db.OpenRecordset("tblJob_Marine", dbOpenTable)
      End If

    Next intI

    rstJM.Close
    rstGrade.Close
   Set db = Nothing

End Function
```

B. CODE FOR MAINTENANCE ON EAM

1. Function Clock()

```
Public Function Clock(lngStart As Long, lngEnd As Long)
  Dim lngTotal As Long
  Dim lngHour As Long
  Dim lngMin As Long
  Dim lngSec As Long
  Dim varTime As Variant

  lngTotal = lngEnd - lngStart
  lngHour = lngTotal \ 3600
  lngMin = (lngTotal - lngHour * 3600) \ 60
  lngSec = (lngTotal - (lngMin * 60) - (lngHour * 3600)) \ 1
  'varTime = Format(lngTotal, "ttttt")
  MsgBox "This took " & lngHour & ": " & lngMin & ": " & lngSec & " to
accomplish...", vbOKOnly, "Elapsed Operation Time"
End Function
```

2. Function SignOut()

```
Public Function SignOut()
Dim db As Database
Dim rstAdmin As Recordset
Dim strFind As String
```

```
Set db = CurrentDb()
Set rstAdmin = db.OpenRecordset("tblAdmin", dbOpenDynaset)


strFind = "[InUse] = True"

With rstAdmin
   .MoveFirst
   .FindFirst strFind
   .Edit
   ![InUse] = False
   .Update
End With

MsgBox "Thank you for using the Enlisted Assignment Model.", vbOKOnly, "EAM"

Set db = Nothing
End Function
```

3. Function FundProps()

```
Public Function FundProps() As String

Dim db As Database
Dim rstFieldValues As Recordset
Dim strField As String
Dim strChar As String
Dim strFundProps As String

Set db = CurrentDb()
Set rstFieldValues = db.OpenRecordset("tblInputFields", dbOpenDynaset)

strField = "[FIELDNAME] = """ & Forms!frmFundamentalProperties!cmbFieldName &
"""
With rstFieldValues
   .MoveFirst
   .FindFirst strField
   strChar = Left(![FieldValues], 1)
   If strChar = "S" Then
      Forms!frmFundamentalProperties!cmbValue.RowSourceType = "Table/Query"
      FundProps = ![FieldValues]
```

```
  ElseIf strChar = "E" Then
    Forms!frmFundamentalProperties!cmbValue.RowSourceType = ""
  Else
    Forms!frmFundamentalProperties!cmbValue.RowSourceType = "Value List"
    FundProps = ![FieldValues]
  End If

End With

Set db = Nothing
End Function
```

4. Function DeleteSelected()

```
Function DeleteSelected(frm As Form) As Integer
  Dim db As Database
  Dim rstFieldValues As Recordset
  Dim strField As String
  Dim strField1 As String
  Dim ctlSource As Control
  'Dim ctlDest As Control
  Dim strItems As String
  Dim strItems1 As String
  Dim intCurrentRow As Integer

  Set db = CurrentDb()
  Set rstFieldValues = db.OpenRecordset("tblFundPropIN", dbOpenDynaset)
  Set ctlSource = frm!lstValue
  'Set ctlDest = frm!lstDestination
  For intCurrentRow = 0 To ctlSource.ListCount - 1
    If ctlSource.Selected(intCurrentRow) Then
      strItems = ctlSource.Column(0, intCurrentRow)

      strField = "[PropName_PK] = """ & frm![txtFundamentalPropName_PK] & """"
      strField1 = "[Value] = """ & strItems & """"
      rstFieldValues.FindFirst strField
      rstFieldValues.FindFirst strField1
      rstFieldValues.Delete
    End If
  Next intCurrentRow
```

```
'Reset destination control's RowSource property.
ctlSource.RowSource = ""
ctlSource.RowSource = "SELECT [tblFundPropIN].[Value] FROM tblFundPropIN " & _
"WHERE [tblFundPropIN].[PropName_PK] = """ &
frm![txtFundamentalPropName_PK] & """"

End Function
```

APPENDIX D: DATA DEFINITION AND TABLE STRUCTURE

A. TABLE: GRADE

Properties

Date Created:	4/7/98 8:50:27 PM	Def. Updatable:	True
Description:	Listing of grades and pay.	Last Updated:	5/9/98 9:54:48 PM
OrderByOn:	False	RecordCount:	8

Columns

Name	Type	Size
GradeId_PK	Number (Long)	4
Grade	Text	2
Pay	Number (Long)	4

Relationships

GRADEtblCurrentJobs

GRADE		tblCurrentJobs
GradeId_PK	1 ∞	GradeId_PK

Attributes:	Enforced, Cascade Updates
Description:	One-To-Many

B. TABLE: MARINES

Properties

Date Created:	4/17/98 12:31:47 PM	Def. Updatable:	True
Description:	All the Marines from text file.	Last Updated:	5/9/98 9:54:49 PM
OrderByOn:	True	RecordCount:	0

Columns

Name	Type	Size
MID(SSN)	Text	9
INI(FM)	Text	2
LNAME	Text	10

DULIM	Text	1
ECC-FLAG	Text	1
SEX	Text	1
MCC	Text	3
RUC	Text	5
FORMMCC	Text	3
FORMRUC	Text	5
FUTRMCC	Text	3
CURLOC	Text	2
AFADBD	Text	6
DOB	Text	6
EDD	Text	6
EDA	Text	6
DCTB	Number (Long)	4
RTD	Text	8
DAUS(DNR	Text	6
DAUS(DRS	Text	6
EASA	Text	8
LENL	Text	1
CLF	Text	1
SEC	Text	1
GRD	Text	1
GRD-DOR	Text	6
SCAT	Text	1
ORDERS	Text	1
DPREF1	Text	3
DPREF2	Text	3
DPREF3	Text	3
SGRD	Text	1
CIT	Text	2
CEDL	Text	1
BMOS	Text	4
PMOS	Text	4
MOS1A	Text	4
MOS2A	Text	4
TCF	Text	2
CURDUS	Text	1
HOR	Text	2
GCT	Text	3
SSCHOOL1	Text	3
SSCHOOL2	Text	3
SSCHOOL3	Text	3
SSCHOOL4	Text	3
SSCHOOL5	Text	3
SSCHOOL6	Text	3
REL1	Text	2
DEPLOC	Text	2
DRWCASE1	Text	1
DRWCASE2	Text	1
DRWCASE3	Text	1
PEN	Text	8
DCS1	Text	3
ITD	Text	6
AGLC	Text	3
AGLC-EDA	Text	6
DMCC	Text	3
TSC	Text	2
LMCC	Text	3
NODEP	Text	1
DSC	Text	1
DRD	Text	6
GLC	Text	3
GEODCTB	Text	6
PTCD	Text	6
ATCD	Text	6
DGLC	Text	3

GEOFLAG	Text	10
SRBP	Text	1
SADM	Text	1
MARSTA	Text	1
COMP	Text	2
ADT	Text	3
CSEC1	Text	1
S/ORDF	Text	7
IMOS	Text	4
blank	Text	2
S/DOP	Text	8
S/SBI	Text	5
S/FMCC	Text	3
S/EDD	Text	8
S/EDA	Text	8
S/FDS	Text	1
S/MAC	Text	1
S/MAD	Text	8
S/TFAC	Text	2
S/ADVASN	Text	3
S/AA-EDA	Text	8
S/AA-FLG	Text	1
MarId	Number (Long)	4

C. TABLE: TBLADMIN

Properties

Date Created:	4/27/98 4:45:19 PM	Def. Updatable:	True
Description:	Contains the data on all users of the model.	Last Updated:	5/9/98 9:54:50 PM
OrderByOn:	False	RecordCount:	2

Columns

Name	Type	Size
AdminId	Number (Long)	4
FName	Text	15
LName	Text	50
Initials	Text	3
Email	Text	50
Password	Text	15
LastUsed	Date/Time	8
InUse	Yes/No	1

D. TABLE: TBLCURRENTJOBS

Properties

Date Created:	4/21/98 1:30:09 PM	Def. Updatable:	True

Description:	Current Jobs in the Marine Corps by MOS,MCC,Grade.	Last Updated:	5/9/98 11:14:46 PM
OrderByOn:	False	RecordCount:	121072

Columns

Name	Type	Size
JobId	Number (Long)	4
MCC	Text	3
MOS	Text	4
GradeId_PK	Number (Long)	4
StaffLevel	Text	1
Description	Text	50

Relationships

GRADEtblCurrentJobs

GRADE		tblCurrentJobs
GradeId_PK	1 ∞	GradeId_PK

Attributes:	Enforced, Cascade Updates
Description:	One-To-Many

tblCurrentJobstblFundJobProp

tblCurrentJobs		tblFundJobProp
JobId	1 ∞	JobId

Attributes:	Enforced, Cascade Updates
Attributes:	One-To-Many

tblCurrentJobstblLogJobProp

tblCurrentJobs		tblLogJobProp
JobId	1 ∞	JobId

Attributes:	Enforced, Cascade Updates
Attributes:	One-To-Many

E. TABLE: TBLCURRENTJOBS

tblMCCtblCurrentJobs

tblMCC		tblCurrentJobs
MCC	1 ∞	MCC

Attributes:	Enforced, Cascade Updates
Attributes:	One-To-Many

tblMOStblCurrentJobs

tblMOS			tblCurrentJobs
MOS	1	∞	MOS

Attributes:	Enforced, Cascade Updates
Attributes:	One-To-Many

F. TABLE: TBLESGM

Properties

Date Created:	4/13/98 3:32:55 PM	Def. Updatable:	True
Description:	The ESGM for all of the Marine Corps.	Last Updated:	4/24/98 9:41:01 AM
OrderByOn:	True	RecordCount:	0

Columns

Name	Type	Size
MOS	Text	4
MCC	Text	3
E9	Number (Long)	4
E8	Number (Long)	4
E7	Number (Long)	4
E6	Number (Long)	4
E5	Number (Long)	4
E4	Number (Long)	4
E3	Number (Long)	4
E2	Number (Long)	4

G. TABLE: TBLFUNDAMENTALPROPERTIES

Properties

Date Created:	4/7/98 7:42:52 PM	Def. Updatable:	True
Description:	Fundamental Properties	Last Updated:	5/9/98 9:55:15 PM
OrderByOn:	True	RecordCount:	9

Columns

Name	Type	Size
FundamentalPropName_PK	Text	10
FieldName	Text	25
Operator	Text	8
Value	Text	30
Description	Text	255
DateTime	Date/Time	8
Initials	Text	4
MCOId	Number (Long)	4
Type	Text	1

Relationships

FUND_PROPSLOG_FUND_PROPS

tblFundamentalProperti			tblLog_Fund_Prop
FundamentalPropName_P	1	∞	FundamentalPropName_FK

Attributes: Enforced, Cascade Updates, Cascade Deletes
Description: One-To-Many

OPERATORSFUND_PROPS

OPERATORS			tblFundamentalProperti
Operator	1	∞	Operator

Attributes: Enforced, Cascade Updates
Attributes: One-To-Many

H. TABLE: TBLFUNDAMENTALPROPERTIES

tblFundamentalPropertiestblFundFamilyProp

tblFundamentalProperti			tblFundFamilyProp
FundamentalPropName_P	1	∞	FundPropName_FK

Attributes: Enforced, Cascade Updates
Attributes: One-To-Many

tblFundamentalPropertiestblFundJobProp

tblFundamentalProperti			tblFundJobProp
FundamentalPropName_P	1	∞	FundPropName_FK

Attributes: Enforced, Cascade Updates
Attributes: One-To-Many

tblFundamentalPropertiestblFundPropIN

tblFundamentalProperti			tblFundPropIN
FundamentalPropName_P	1	∞	PropName_PK

Attributes:	Enforced, Cascade Updates
Attributes:	One-To-Many

tblInputFieldstblFundamentalProperties

tblInputFields			tblFundamentalProperti
FIELDNAME	1	∞	FieldName

Attributes:	Enforced, Cascade Updates
Attributes:	One-To-Many

tblMCOtblFundamentalProperties

tblMCO			tblFundamentalProperti
OrderId	1	∞	MCOId

Attributes:	Enforced, Cascade Updates
Attributes:	One-To-Many

I. TABLE: TBLFUNDJOBPROP

Properties

Date Created:	5/4/98 8:37:01 PM	Def. Updatable:	True
Description:	The Association of Fundamental properties with jobs.	Last Updated:	5/9/98 9:54:49 PM
OrderByOn:	False	RecordCount:	17

Columns

Name	Type	Size
FundPropName_FK	Text	50
JobId	Number (Long)	4
Level	Number (Long)	4

Relationships

tblCurrentJobstblFundJobProp

tblCurrentJobs			tblFundJobProp
JobId	1	∞	JobId

Attributes:	Enforced, Cascade Updates
Description:	One-To-Many

tblFundamentalPropertiestblFundJobProp

tblFundamentalProperti			tblFundJobProp
FundamentalPropName_P	1	∞	FundPropName_FK

Attributes:	Enforced, Cascade Updates
Attributes:	One-To-Many

J. TABLE: TBLFUNDPROPIN

Properties

Date Created:	4/28/98 10:42:14 PM	Def. Updatable:	True
Description:	Fundamental properties that contain the in or not-in	Last Updated:	5/9/98 9:55:15 PM
OrderByOn:	False	RecordCount:	11

Columns

Name	Type	Size
PropName_PK	Text	50
Value	Text	10

Relationships

tblFundamentalPropertiestblFundPropIN

tblFundamentalProperti			tblFundPropIN
FundamentalPropName_P	1	∞	PropName_PK

Attributes:	Enforced, Cascade Updates
Description:	One-To-Many

K. TABLE: TBLINPUTFIELDS

Properties

Date Created:	4/10/98 9:45:48 PM	Def. Updatable:	True
Description:	Current Marine input fields.	Last Updated:	5/15/98 12:05:20 PM
OrderBy:	tblInputFields.STARTINGBYTE	OrderByOn:	True

RecordCount: 104

Columns

Name	Type	Size
FIELDNAME	Text	8
STARTINGBYTE	Number (Integer)	2
FIELDLENGTH	Number (Integer)	2
FIELDFORMAT	Text	12
COMMENTS	Text	150
FieldValues	Text	150

Relationships

tblInputFieldstblFundamentalProperties

tblInputFields			tblFundamentalProperti
FIELDNAME	1	∞	FieldName

Attributes: Enforced, Cascade Updates
Description: One-To-Many

L. TABLE: TBLJOB_MARINE

Properties

Date Created:	4/17/98 4:44:03 PM	Def. Updatable:	True
Description:	Intersection of Marines with	Last Updated:	4/24/98 9:13:22 AM
OrderByOn:	False	RecordCount:	0

Columns

Name	Type	Size
MarId	Number (Long)	4
period	Text	1
JobId	Number (Long)	4
Cost	Number (Long)	4

M. TABLE: TBLJOBS

Properties

Date Created:	4/16/98 1:35:38 PM	Def. Updatable:	True
Description:	Current Jobs and quotas.	Last Updated:	5/9/98 9:55:15 PM

OrderByOn:	True	RecordCount:	0

Columns

Name	Type	Size
MCC	Text	3
MOS	Text	4
GradeId_PK	Number (Long)	4
Quota	Number (Long)	4
StaffLevel	Text	1
JobId	Number (Long)	4

N. TABLE: TBLLOGFAMILYPROP

Properties

Date Created:	5/8/98 9:54:41 AM	Def. Updatable:	True
Description:	The association of logical properties to jobs.	Last Updated:	5/9/98 9:55:16 PM
OrderByOn:	False	RecordCount:	0

Columns

Name	Type	Size
LogicalPropName_FK	Text	50
OCCField	Text	2
Level	Number (Long)	4

Relationships

tblLogicalPropertiestblLogFamilyProp

tblLogicalProperties			tblLogFamilyProp
LogicalPropName_PK	1	∞	LogicalPropName_FK

Attributes:	Enforced, Cascade Updates
Description:	One-To-Many

tblOCCFieldstblLogFamilyProp

tblOCCFields			tblLogFamilyProp
OCCField	1	∞	OCCField

Attributes:	Enforced, Cascade Updates
Attributes:	One-To-Many

O. TABLE: TBLLOGICALPROPERTIES

Properties

Date Created:	4/7/98 7:43:21 PM	Def. Updatable:	True
Description:	Logical Properties, a logical combination of fundamental properties.	Last Updated:	5/9/98 9:55:15 PM
OrderByOn:	False	RecordCount:	2

Columns

Name	Type	Size
LogicalPropName_PK	Text	15
LogicalEquation	Text	75
Description	Text	255
DateTime	Date/Time	8
Initials	Text	4
MCOId	Number (Long)	4
Type	Text	1

Relationships

tblLogicalPropertiestblLog_Fund_Prop

tblLogicalProperties			tblLog_Fund_Prop
LogicalPropName_PK	1	∞	LogicalPropName_FK

Attributes:	Enforced, Cascade Updates
Description:	One-To-Many

tblLogicalPropertiestblLogFamilyProp

tblLogicalProperties			tblLogFamilyProp
LogicalPropName_PK	1	∞	LogicalPropName_FK

Attributes:	Enforced, Cascade Updates
Attributes:	One-To-Many

P. TABLE: TBLLOGICALPROPERTIES

tblLogicalPropertiestblLogJobProp

tblLogicalProperties			tblLogJobProp
LogicalPropName_PK	1	∞	LogicalPropName_FK

Attributes:	Enforced, Cascade Updates
Attributes:	One-To-Many

tblMCOtblLogicalProperties

tblMCO		tblLogicalProperties
OrderId	1 ∞	MCOId

Attributes:	Enforced, Cascade Updates
Attributes:	One-To-Many

Q. TABLE: TBLLOGJOBPROP

Properties

Date Created:	5/4/98 8:47:20 PM	Def. Updatable:	True
Description:	The association of logical properties to jobs.	Last Updated:	5/9/98 10:22:23 PM
OrderByOn:	False	RecordCount:	6

Columns

Name	Type	Size
LogicalPropName_FK	Text	50
JobId	Number (Long)	4
Level	Number (Long)	4

Relationships

tblCurrentJobstblLogJobProp

tblCurrentJobs		tblLogJobProp
JobId	1 ∞	JobId

Attributes:	Enforced, Cascade Updates
Description:	One-To-Many

tblLogicalPropertiestblLogJobProp

tblLogicalProperties		tblLogJobProp
LogicalPropName_PK	1 ∞	LogicalPropName_FK

Attributes:	Enforced, Cascade Updates
Attributes:	One-To-Many

R. TABLE: TBLMCC

Properties

Date Created:	4/7/98 8:42:30 PM	Def. Updatable:	True
Description:	Current Monitor Command	Last Updated:	5/9/98 9:55:15 PM
OrderByOn:	True	RecordCount:	1081

Columns

Name	Type	Size
MCCId_PK	Number (Long)	4
Description	Text	50
TO_Number	Text	10
MCC	Text	3

Relationships

tblMCCtblCurrentJobs

tblMCC		tblCurrentJobs
MCC	1 ∞	MCC

Attributes:	Enforced, Cascade Updates
Description:	One-To-Many

S. TABLE: TBLMCO

Properties

Date Created:	4/18/98 3:09:22 PM	Def. Updatable:	True
Description:	Table for Marine Corps Orders dealing with Manpower issues.	Last Updated:	5/9/98 9:55:15 PM
OrderByOn:	False	RecordCount:	12

Columns

Name	Type	Size
OrderId	Number (Long)	4
MCOrder	Text	50
MCOTitle	Text	100

Relationships

tblMCOtblFundamentalProperties

tblMCO			tblFundamentalProperti
OrderId	1	∞	MCOId

Attributes:	Enforced, Cascade Updates
Description:	One-To-Many

tblMCOtblLogicalProperties

tblMCO			tblLogicalProperties
OrderId	1	∞	MCOId

Attributes:	Enforced, Cascade Updates
Attributes:	One-To-Many

T. TABLE: TBLMOS

Properties

Date Created:	4/7/98 7:44:08 PM	Def. Updatable:	True
Description:	Military Occupational Specialties.	Last Updated:	5/9/98 9:55:15 PM
OrderByOn:	True	RecordCount:	272

Columns

Name	Type	Size
MOS	Text	4
Description	Text	50
Family	Text	2

Relationships

tblMOStblCurrentJobs

tblMOS			tblCurrentJobs
MOS	1	∞	MOS

Attributes:	Enforced, Cascade Updates
Description:	One-To-Many

tblOCCFieldstblMOS

tblOCCFields			tblMOS
OCCField	1	∞	Family

Attributes:	Enforced, Cascade Updates
Attributes:	One-To-Many

U. TABLE: TBLMOVERS

Properties

Date Created:	4/18/98 9:10:25 PM	Def. Updatable:	True
Description:	Marines that are available to move.	Last Updated:	5/9/98 9:55:15 PM
OrderByOn:	False	RecordCount:	0

Columns

Name	Type	Size
MarId	Number (Long)	4
MCC	Text	3
BMOS	Text	4
PMOS	Text	4
GRD	Number (Long)	4
SEX	Text	1
LNAME	Text	10

V. TABLE: TBLNONMOVERS

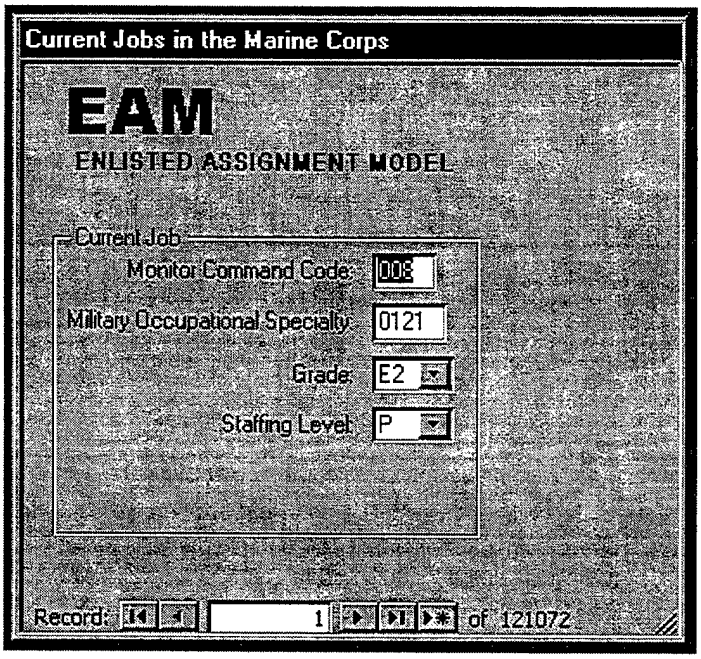
Properties

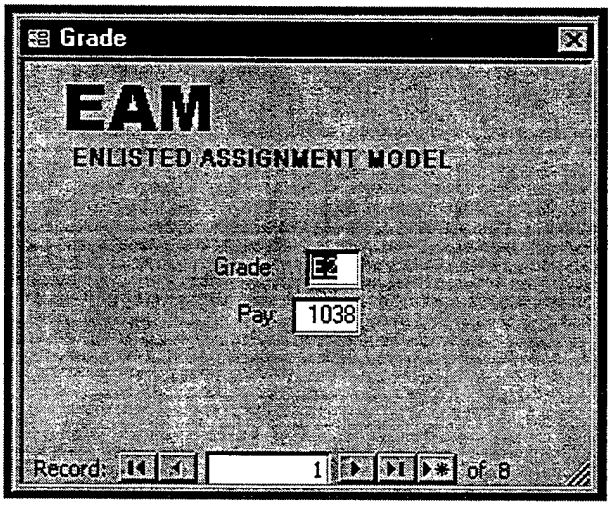
Date Created:	4/18/98 9:15:04 PM	Def. Updatable:	True
Description:	Marines that are not available to move.	Last Updated:	5/9/98 9:55:15 PM
OrderByOn:	False	RecordCount:	0

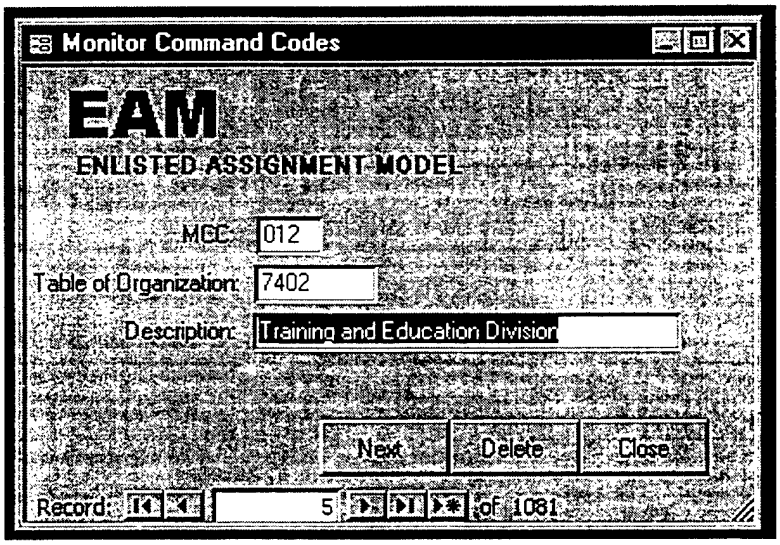
Columns

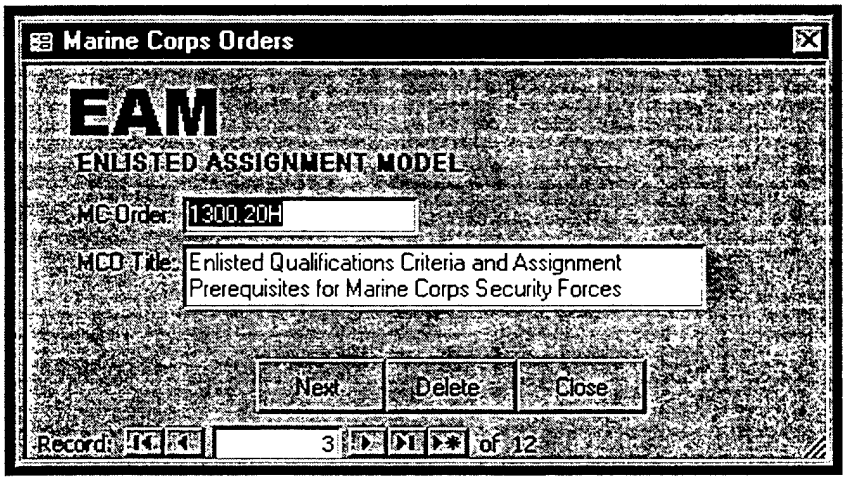
Name	Type	Size
MarId	Number (Long)	4
MCC	Text	3
BMOS	Text	4
PMOS	Text	4
GRD	Text	1
LNAME	Text	10

APPENDIX E: DATABASE MAINTENANCE INTERFACES






Monitor Command Codes
EAM
ENLISTED ASSIGNMENT MODEL
MCC: 012
Table of Organization: 7402
Description: Training and Education Division
Next
Delete
Close
Record: 5 of 1081


Marine Corps Orders
EAM
ENLISTED ASSIGNMENT MODEL
MC Order: 1300.20H
MCO Title: Enlisted Qualifications Criteria and Assignment Prerequisites for Marine Corps Security Forces
Next
Delete
Close
Record: 3 of 12

APPENDIX F: LIST OF ACRONYMS

AFADBD	Armed Forces Active Duty Base Date
ASR	Authorized Strength Report
BMOS	Billet MOS
BPR	Business Process Engineering
CEDL	Current Education Level
CIT	Citizenship Status
CONUS	Continental United States
CURDUS	Current Duty Status
CURLOC	Current Location state or country
DAO	Data Access Object
DAUS(DNR	Date Arrived US (Not Restricted)
DAUS(DRS	Date Arrived US (Dep Restricted)
DBMS	Database Management System
DCC	Draw Case Code
DCTB	Date Current Tour Began
DMCC	Deployment MCC
DOB	Date of Birth
DPREF1	Duty Preference 1
DPREF2	Duty Preference 2
DPREF3	Duty Preference 3
DRD	Deployment Return Date
DSAI	Decision Support Associates, Inc
DSC	Deployment Status Code
DSS	Decision Support System
DULIM	Duty Limitations
EAM	Enlisted Assignment Model
EAS	Expiration of active service
ECC	Expiration of current contract
EDA	Estimated Date of Arrival
EDD	Estimated Date of Departure
ER	Entity Relationship
ESGM	Enlisted Staffing Goal Model
FORMMCC	Former MCC
FORMRUC	Former RUC
FUTRMCC	Future MCC
GCT	GCT Score
GLC	Geographic Location Code
GRD	Present Grade
GRD-DOR	Present Grade Date of Rank
HOR	Home of record
HRPDIMS	Human Resource Process Development Information Management Systems

LENL	Length of Current Enlistment
LMCC	Last MCC (before former)
LNAME	Last Name
M & R A	Manpower and Reserve Affairs
MARSTA	Marital status
MCC	Monitor Command Code
MCO	Marine Corps Order
MCTFS	Marine Corps Total Force System
MOS	Military Occupational Specialty
MOS1A	First additional MOS
MOS2A	Second additional MOS
PCS	Permanent Change of Station
PMOS	Primary MOS
RAD	Rapid Application Development
RDM	Recruit Distribution model
RTD	Rotation Tour Date
RUC	Reporting Unit Code
SADT	Structured Analysis and Design Technique
SGRD	Selected Grade
SOP	Standard Operating Procedure
SSN	Social Security Number
SRBP	Selective Reenlistment Bonus Program
T/O	Table of Organization
TCF	Tour Control Factor
TOS	Time on Station
TSC	Tour Sequence Code
USMC	United States Marine Corps
VBA	Visual Basic for Applications

LIST OF REFERENCES

Bui, Tung X., class notes IS 4200 "Systems Analysis and Design", Business Process Re-Engineering: Executive Programme, Naval Postgraduate School, Monterey, CA, November 25, 1996.

Christmas, G., Deputy Chief of Staff for Manpower & Reserve Affairs, USMC, "Memorandum on the Manpower Process," 1996.

Herrick, Craig L., "A Survey of Software for Decision Analysis", Master's Thesis, Naval Postgraduate School, Monterey, CA, March 1997.

Snoap, Kevin J., "Business Process Re-Engineering of the USMC Recruit Distribution Model", Proposed Master's Thesis, Naval Postgraduate School, Monterey, CA, September 1998.

Tivnan, Brian F., "Optimizing United States Marines Corps Enlisted Assignments", Proposed Master's Thesis, Naval Postgraduate School, Monterey, CA, September 1998.

USMC, Manpower Management Information Systems Branch, Manpower and Reserve Affairs Department, "Turnover File: EAM Manager," 1992.

BIBLIOGRAPHY

Bhargava, Hemant, "Decision Support Systems IS4185 Lecture Notes", Naval Postgraduate School, 1996.

Cornell, Gary, Visual Basic 5 From the Ground Up, McGraw-Hill, Berkely, CA, 1997.

Craig, John Clark, and Webb, Jeff, Microsoft Visual Basic 5.0 Developers Workshop, 4th Edition, Microsoft Press, Redmond, WA, 1997.

Davis, Michael, D, Applied Decision Support, chapter eight "The User Interface", pp 93-110, Prentice-Hall, Englewood Cliffs, NJ, 1988.

Decision Systems Associates, Inc., "Users Guide, Enlisted Assignment Model (EAM) Version 2.02", May 1991.

Decision Systems Associates, Inc., "U.S. Marine Corps PREPAS Models Conversion Software and Hardware Evaluation," October 1992.

Early, Steve H., "Decision Net: A Database Approach", Master's Thesis, Naval Postgraduate School, Monterey, CA, September 1996.

Herrick, Craig L., "A Survey of Software for Decision Analysis", Master's Thesis, Naval Postgraduate School, Monterey, CA, March 1997.

Hoffer, J.,George, J., and Valacich, J., "Modern Systems Analysis and Design", The Benjamin/Cummings Publishing Company, Inc., 1996

Kamel, Magdi, "Database Design", IS4183 class notes, Naval Postgraduate School Monterey, CA, December 1996.

Klingman, D., and Phillips, N., "Topological and Computational Aspects Of Preemptive Multi-criteria Military Personnel Assignment Problems," *Management Science*, vol. 30, no. 11, pp. 1362-1375, 1984.

Kroenke, David M.,Database Processing - Fundamentals, Design, and Implementation, 5th Edition, Prentice-Hall, Englewood Cliffs, NJ, 1995.

Levindon, Les, and Robins, Kevin, ed., Cyborg Wars: The Military Information Society, Free Association Books, London, 1989.

Litwin, Paul, and Getz, Ken, with Gilbert, Mike and Reddick, Greg, Microsoft Access 95 Developers Handbook, 2nd Edition, Sybex, Alameda, CA, 1996.

Logic Works, Inc., "Logic Works BPWin Tutorial Guide," 1994 - 1995.

Malhotra, Yogesh, "Business Process Redesign: An Overview", (http://www.brint.com/papers/bpr.htm), 1996.

MCO 1300.31A "Enlisted Classification and Assignment Documents", Headquarters Marine Corps, Washington, DC, 3 Mar 1992.

MCO 5320.12C "Staffing Precedents for Officer and Enlisted Billets", Headquarters Marine Corps, Washington, DC, 27 May 1994.

MCO P1300.8R "Marine Corps Personnel Assignment Policy", Headquarters Marine Corps, Washington, DC, 4 Oct 1994.

USMC, Manpower Management Information Systems Branch, Manpower and Reserve Affairs Department, "Statement of Work for the USMC Manpower Model Modernization: Reengineering of the Enlisted Assignment Model and Recruit Distribution Model," 1997.

USMC, Manpower Management Information Systems Branch, Manpower and Reserve Affairs Department, "Turnover File: EAM Manager," 1992.

Van Crevald, Martin, The Training of Officers: From Military Professionalism to Irrelevance, The Free Press, New York, 1990.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center.....2
   8725 John J. Kingman Road, Ste 0944
   Fort Belvior, VA 22060-6218

2. Dudley Knox Library.....2
   Naval Postgraduate School
   411 Dyer Road
   Monterey, CA 93943-5101

3. Director, Training and Education.....1
   MCCDC (Code C46)
   1019 Elliot Road
   Quantico, VA 22134-5027

4. Director, Marine Corps Research Center.....2
   MCCDC (Code C40RC))
   2040 Broadway Street
   Quantico, VA 22134-5107

5. Director, Studies and Analysis Division 1
   MCCDC (Code C45)
   3300 Russell Road
   Quantico, VA 22134-5130

6. Marine Corps Representative.....1
   Naval Postgraduate School
   Code 037, Bldg. 234, HA-220
   699 Dyer Road
   Monterey, CA 93943-5000

7. Marine Corps Tactical Systems Support Activity.....1
   Technical Advisory Branch
   ATTN: Major Cummiskey
   Box 555171
   Camp Pendleton, CA 92055-5080

8. Hemant Bhargava (Code SM/BH).....2
   Naval Postgraduate School
   Systems Management Department
   Monterey, CA 93943-5000

9. Suresh Sridhar (Code SM/SR).....1
   Naval Postgraduate School
   Systems Management Department
   Monterey, CA 93943-5000

10. Major Gary D. Koch Jr.....1
    2083 Redwood Crest
    Vista, CA 92038

11. Gary D. Koch Sr.....1
    4547 Vista Dr.
    Phoenix, AZ 85032