A SYSTEMS ENGINEERING APPROACH TO

THE DESIGN OF A COTS MANAGEMENT SYSTEM

By

Daniel L. Basil

Research Report submitted to the Faculty of the Virginia Polytechnic and State University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

SYSTEMS ENGINEERING

Approved:

Dr. Harold A. Kursheedt, Jr., Chairman

Dean Benjamin Blanchard

H. Todd Smith

December 1996
Blacksburg, Virginia

LD
5655
V251,
1976
B375
C.2
A COTS MANAGEMENT SYSTEM

By

Daniel L. Basil

Committee Chairman: Dr. Harold A. Kurstedt, Jr.

Systems Engineering

(Abstract)

This project applies the management systems engineering methodology to the design and development of a management tool to be used in managing change of the Commercial-Off-The-Shelf [COTS] software, firmware, and hardware utilized within the Mapping Branch of the National Imagery and Mapping Agency [NIMA] (formerly known as the Defense Mapping Agency [DMA]). The management tool is used to support the management process overlaying the work process for changing COTS products. By using the management systems engineering methodology to understand how COTS products are changed, the functional requirements for a management tool can be derived in a clear and concise manner. Concurrently, the systems approach is applied in the development of the management tool, referred to as the COTS Management System.

The COTS Management System is identified as required by NIMA due to the increasing number of COTS products being used within the Digital Production System [DPS]. The DPS is NIMA’s conglomeration of segmented computer systems combined to provide the tools and functionality necessary to produce the myriad of mapping, charting, and geodesic products in both hard-copy and digital format. NIMA also desires to
implement the COTS Management System in an effort to achieve the Software Engineering Institutes [SEI] Capability Maturity Model [CMM] Level 3.

In the early stages of design, the management systems engineering methodology is used to establish and understand how COTS products are changed (the techniques and practices required to coordinate and control both the insertion and upgrade of COTS software within the DPS). Changing COTS products begins with a change proposal, which can be a request for a new procurement or a request to perform a version upgrade. The COTS product change request then proceeds to an engineering evaluation to determine the necessity and feasibility of the change. If the decision is to move forward, the test and evaluation of the software modification is performed. If the test and evaluation are successful, the COTS change ends with the coordinated installation and verification of the COTS product to the DPS. By applying the management systems engineering methodology to how COTS products are changed, the proper mechanisms and informational requirements for managing the change of COTS products can be derived.

The COTS Management System is designed, utilizing the systems approach, to support the COTS management process. The COTS Management System should be flexible enough to be modified or expanded as its requirements change. By understanding the COTS management process, the COTS Management System can be conceived from a functional framework. The systems engineering approach is applied to identify the functional requirements for the system, develop the detailed design, and support the
system life cycle process from development to the eventual retirement of the system. The system functional requirements provide the necessary framework to design an information system using available hardware and software.

Though the COTS Management System applies specifically to NIMA, the system will be developed in a method to be easily re-used within any similar environment. A management tool of this type could be useful in many environments that use a combination of custom applications and COTS products.
Acknowledgments

I would like to thank all of the individuals who assisted me with this project and my studies at Virginia Polytechnic Institute and State University. I sincerely appreciate the help and guidance that I received from Dr. Harold Kurstedt, Dean Benjamin Blanchard, and Mr. Todd Smith on this effort.

I would doubly like to thank Todd, along with two other good friends, Joe Crossett and Dave Biddinger, for their support and camaraderie throughout the academic trials of this Masters Degree program. I thank Sandy Rose for editing and proof-reading this document, and for her support and friendship throughout this semester. I would additionally like to thank Monique Fauquet and John Brown, two coworkers at the NIMA facility in Reston who are also involved in this process. Monique has worked with me in the development of the COTS change process and the user interface for the database system, and John has worked with me on the logistics end of the COTS change process.

I would finally and most importantly like to acknowledge the memory of my mother, Mrs. Anna Mae Basil, may she rest in peace. Without her example of integrity, compassion, and perseverance, it is doubtful that I would be here today. I attribute anything positive that I have done or will ever do in my life to her.
Table of Contents

Abstract ............................................................................................................. i
Acknowledgments ............................................................................................ v
Table of Contents ............................................................................................ vi
List of Figures .................................................................................................. viii
List of Tables .................................................................................................... ix

CHAPTER 1: INTRODUCTION .............................................................................. 1
  PROJECT OBJECTIVES ............................................................................. 1
  BACKGROUND INFORMATION .............................................................. 2
  PROBLEM STATEMENT .......................................................................... 3
  MAIN PROJECT GOAL ............................................................................. 4
  REASON FOR TOPIC SELECTION .......................................................... 4
  PROJECT OVERVIEW .............................................................................. 5

CHAPTER 2: CONCEPTUAL SYSTEM DESIGN ................................................... 8
  CONCEPTUAL SYSTEM DESIGN OVERVIEW ........................................... 8
  DEFINITION OF NEED ........................................................................... 10
  SYSTEM OPERATIONAL CONCEPT ....................................................... 12
  FEASIBILITY STUDY ............................................................................. 13
  SYSTEM OPERATIONAL REQUIREMENTS ........................................... 20
    Mission Definition ............................................................................... 20
    Performance and Physical Parameters .............................................. 21
    Use Requirements ............................................................................. 24
    Operational Deployment and Distribution ..................................... 25
    Operational Life Cycle .................................................................. 26
    Effectiveness Factors ..................................................................... 27
    Environment .................................................................................... 28
  SYSTEM MAINTENANCE CONCEPT ....................................................... 28
    Anticipated Levels of Maintenance Support ................................... 28
    Responsibilities for Support ............................................................ 32
    Repair Policy ................................................................................... 33
    Effectiveness Responsiveness ........................................................ 33
    Maintenance Environment .............................................................. 34
  SYSTEM SPECIFICATION .................................................................. 34
  TECHNICAL PERFORMANCE MEASURES ............................................ 37

CHAPTER 3: APPLICATION OF MANAGEMENT SYSTEMS ENGINEERING ....... 40
  BUILDING MANAGEMENT TOOLS ........................................................... 41
  Defining the COTS Change Process ...................................................... 43
  Work Breakdown Structure for the COTS Change Process ............... 44
  Application of the Management System Analysis ............................. 64
  A HOLISTIC VIEW OF MANAGEMENT TOOLS ....................................... 68
  The Conceptual Model for Management Tools and Guides ............... 68

vi
List of Figures

Figure 1-1 The Systems Engineering Life Cycle (Overview) ..................................................... 7
Figure 2-1 Conceptual System Design Phase (Overview). {Adapted from Blanchard and Fabrycky, p.35} ..................................................... 9
Figure 2-2 Feasibility Study (Network Application). ................................................................. 16
Figure 2-3 Feasibility Study (Client Application). ................................................................. 17
Figure 2-4 Mission Use Profile for a Typical Day. ................................................................. 25
Figure 2-5 System Operational Deployment. ........................................................................... 26
Figure 2-6 Maintenance Concept (Overview). ........................................................................ 29
Figure 3-1 Expanded Management System Model. {Taken from Kurstedt, p.201} ............. 42
Figure 3-2 The COTS Change Process. .................................................................................. 45
Figure 3-3 Function 1: New Product Requests. ................................................................. 47
Figure 3-4 Function 2: Product Evaluation. ................................................................. 51
Figure 3-5 Function 3: Product Testing. ................................................................. 56
Figure 3-6 Function 4: Product Installation. ........................................................................ 61
Figure 3-7 Management System Analysis (Functions for Building Management Tools). {Taken from Kurstedt, p.246} ......................................................... 65
Figure 3-8 Management System Analysis of the COTS Change Process. ......................................................... 67
Figure 3-9 Conceptual Model for Management Tools and Guides. {Adapted from Kurstedt, p.380} ......................................................... 69
Figure 3-10 Establishing Relationships Among Data Objects. ............................................. 71
Figure 4-1 Preliminary System Design Phase (Overview). {Adapted from Blanchard and Fabrycky, p.56} ......................................................... 74
Figure 4-2 COTS Management System Top-Level Flow. ...................................................... 77
Figure 4-3 First-Level Operational Flow Diagram. ............................................................. 79
Figure 4-4 Second-Level Operational Flow Diagram. ......................................................... 81
Figure 4-5 Third-Level Operational Flow Diagram. ........................................................... 83
Figure 4-6 First-Level Maintenance Flow Diagram. ............................................................ 85
Figure 4-7 Second-Level Maintenance Flow Diagram. ......................................................... 87
Figure 4-8 Third-Level Maintenance Flow Diagram. ........................................................... 89
Figure 4-9 Allocation of Functional Requirements. ............................................................ 91
Figure 4-10 Using MS Access for the Client and Server. .................................................... 94
Figure 4-11 Client-Server Relationship. ............................................................................. 96
Figure 5-1 Detail Design and Development Phase (Overview). {Adapted from Blanchard and Fabrycky, p.83} ......................................................... 98
Figure 5-2 Client-Server Configuration. ............................................................................. 100
Figure 5-3 COTS Management System - Database Design (Overview). ............................ 103
Figure 6-1 Test and Evaluation Process (Overview). {Taken from Blanchard and Fabrycky, p.101} ......................................................... 114

viii
List of Tables

TABLE 2-1 INITIAL FEASIBLE DESIGN ................................................................. 19
TABLE 2-2 PARTIAL SYSTEM SPECIFICATION ..................................................... 35
TABLE 2-3 TECHNICAL PERFORMANCE MEASURES .......................................... 39
TABLE 4-1 SOFTWARE FUNCTIONAL MODULES ................................................ 92
TABLE 5-1 ACTION ASSIGNMENTS TABLE COLUMNS ....................................... 104
TABLE 5-2 ACTION RESPONSE TABLE COLUMNS ........................................... 104
TABLE 5-3 ACTIONS LOOKUP TABLE COLUMNS .............................................. 105
TABLE 5-4 ALTERNATIVES TABLE COLUMNS ................................................... 105
TABLE 5-5 COTS REQUEST (CSCR) TABLE COLUMNS ...................................... 106
TABLE 5-6 IMPACTS & RECOMMENDATION TABLE COLUMNS ......................... 107
TABLE 5-7 MEMO TABLE COLUMNS .................................................................. 107
TABLE 5-8 ORGANIZATIONS LOOKUP TABLE COLUMNS .................................. 108
TABLE 5-9 PLATFORMS LOOKUP TABLE COLUMNS .......................................... 108
TABLE 5-10 SITES LOOKUP TABLE COLUMNS .................................................. 108
TABLE 5-11 STATUS LOOKUP TABLE COLUMNS ............................................ 109
TABLE 5-12 SUB-SYSTEM LOOKUP TABLE COLUMNS ....................................... 109
TABLE 5-13 TEST REPORT COLUMNS ............................................................... 109
TABLE 5-14 POPULATED ACTIONS LOOKUP TABLE ......................................... 110
TABLE 5-15 POPULATED STATUS LOOKUP TABLE ......................................... 111
TABLE 6-1 REQUIREMENT VERIFICATION TRACEABILITY MATRIX ....................... 116
CHAPTER 1: INTRODUCTION

PROJECT OBJECTIVES

This project applies the management systems engineering methodology to the design and development of a management tool to be used in managing change of the Commercial Off The Shelf [COTS] software, firmware, and hardware utilized within the Mapping Branch of the National Imagery and Mapping Agency [NIMA] (*formerly known as the Defense Mapping Agency [DMA]*). The management tool is used to support the management process overlaying the work process for changing COTS products. By using the management systems engineering methodology to understand how COTS products are changed, the system requirements can be derived in a clear and concise manner. Next, the systems approach is applied to define the system requirements necessary in the development of this management tool. This management tool is hereafter referred to as the COTS Management System.

A key objective of this project is to properly integrate the various engineering disciplines throughout the life cycle in an effort to create a more robust overall system. For the purposes of this project, emphasis will be placed on the early phases of system design and development. By combining Systems Engineering, Management Systems Engineering, Human Factors Engineering, Software Engineering, Maintainability
Engineering, and other engineering areas, a holistic view of the COTS Management System can be obtained to ensure that the system truly supports the customer.

**BACKGROUND INFORMATION**

The COTS Management System is identified as required by NIMA due to the increasing number of COTS products that are being used within the Digital Production System [DPS]—NIMA’s six segmented computer systems combined to provide the tools and functionality necessary to produce the myriad of mapping, charting, and geodesic products in both hard-copy and digital format. The DPS is operated at each of NIMA’s three main mapping sites. NIMA anticipates that the use of COTS products in their environments will continue to increase as they migrate to more “open system architectures.” NIMA desires to define the COTS *change process*, which is all the steps that are required to perform a COTS version upgrade or insert a new COTS product into the DPS. They additionally want to implement a COTS Management System in an effort to achieve the Software Engineering Institute’s [SEI] Capability Maturity Model [CMM] Level 3.

Prior to the implementation of this process, there was no integrated COTS Management System in use at NIMA. Each segment of the DPS was under the control of a specific contractor responsible for the Research and Development [R&D] as well as the Operations and Maintenance [O&M] of their specific segment. In an effort to both reduce
costs and improve coordination among DPS segments, NIMA solicited a proposal for a Consolidated Maintenance Services [CMS] contract with the purpose of bringing the responsibility for O&M under a single organization. This contract was awarded to the Hughes team (consisting primarily of Hughes, Intergraph, and GDE) and went into effect on July 1, 1995. Currently, my company, Quality Systems Incorporated [QSI] is sub-contracted to GDE on the CMS contract. I have been involved with NIMA and CMS since August 29, 1995.

**PROBLEM STATEMENT**

Due to the complexities involved with NIMA’s computer systems, they have been developed in a segmented approach by various contractors as described above. There are numerous interfaces between segments, and the installation of uncoordinated COTS modifications to the system can have detrimental effects on the entire system. Therefore, it is necessary to ensure that the proper checks and balances are applied prior to any new COTS product insertion or COTS product upgrades.

Another part of the problem is in developing the methodology for selecting a new COTS product; in other words, which product will best meet the needs of both the requesting organization and NIMA. Often procurements are made without a complete understanding of the requirements, a unified long-term direction, or comprehensive market research to determine the best possible COTS selection.
**MAIN PROJECT GOAL**

The goal of this project is to design and develop a management tool to support the management process used in the change of COTS products within the DPS. Here, COTS change is defined as either a new version upgrade that is released by a vendor or a brand new product that is being inserted into the DPS.

**REASON FOR TOPIC SELECTION**

I was assigned to this task by CMS management in early July 1996, and at that time little progress had been made toward integrating the COTS change process. Some initial steps had been made toward flowcharting the proposed process and some rough forms had been created, but little was going on. The DPS segments were operating independently or not at all. There was a total lack of coordination and communication with respect to version upgrades, and no one knew how to make requests for new COTS products. My job was and still is to try to fix the problem.

I have been working closely with the CMS Integrated Logistics Services [ILS] group in an effort to get this process established and working. I selected this topic because there is much to be done toward establishing a working COTS Management System within NIMA. I am proposing that by instituting the processes and tools within this report, NIMA will be able to effectively manage their COTS products within the DPS.
Additionally, the methodologies and practices outlined in this paper can be inserted into any similar environment to achieve like results.

**PROJECT OVERVIEW**

The systems engineering approach involves designing systems within a life cycle context. For this project, management systems engineering methodology is applied to the early stages of the systems engineering life cycle in an attempt to fully understand the need and the “work processes” that the COTS Management System must be designed to satisfy. This relationship is depicted in Figure 1-1 below.
Figure 1-1 The Systems Engineering Life Cycle (Overview)

During Preliminary and Detail Design Phases, management systems methodology is again applied in an attempt to develop the tools that deliver the right kinds of information. Management systems engineering and the systems approach are concurrent, and they occur simultaneously throughout the entire life cycle that is depicted in Figure 1-
1. The emphasis here is to start using these concepts from the beginning to understand the need for data and to examine the interfaces between people and systems. Through proper use of management systems engineering, the COTS Management System will be developed as an effective tool that provides *rich information* to the proper decision makers in a timely fashion.
CHAPTER 2: CONCEPTUAL SYSTEM DESIGN

CONCEPTUAL SYSTEM DESIGN OVERVIEW

The conceptual system design phase is the initial level of design when using the systems engineering approach as described by Blanchard and Fabrycky in their book, Systems Engineering and Analysis. Figure 2-1 displays the major steps of the conceptual system design phase, which is where the initial system requirements are derived.
The conceptual system design process is used to define the system-level requirements, and it consists of three major activities. First, a definition of need is established based upon an identified deficiency (perceived or real). Feasibility studies are performed concurrently with requirements definition efforts to examine various technological approaches that can be used in system development. In the feasibility study, a broad scope of possible problem solutions are studied. The domain of feasible solutions is narrowed through requirements definition, which constrains the scope of feasible solutions.
The second major activity of conceptual design consists of several sub-activities that each build upon the previous step. Initially, a high level system operational concept is developed. From this system operational concept, a set of operational requirements is determined. The operational requirements then drive the high level maintenance concept for support of the system. All of these conceptual constraints can then be used to conduct a preliminary system analysis in which various feasible alternatives can be studied or modeled.

The third major activity of conceptual design involves developing a top-level system specification from the operational and maintenance requirements and the preliminary design work. Additionally, the technical performance measures that define the system are developed. These measures are weighted according to their importance to the final design, and aid engineers in making design decisions throughout the life-cycle.

**Definition of Need**

A need is defined by identifying the current system deficiencies that are the driving force behind the need. In this case, there is no current system to analyze, so the need is driven by the things that are required but are currently lacking. In support of achieving the SEI's CMM level 3 (and higher), NIMA needs a methodology that will be used to control and coordinate the *COTS change process*. Additionally, there is a need for the management tools necessary to implement this methodology.
The COTS change process is all the steps it takes to either: (1) evaluate a request for a new COTS software package, choose among potential alternatives, evaluate the impacts, and smoothly test and install the new software into the DPS; or (2) evaluate the impacts of a COTS version upgrade and smoothly test and install the new software into the DPS. The goal of both processes is the same: a transparent change to the system with no lost cartographic production time. Another goal of the evaluation process is to determine when making COTS changes is not a good idea. Often times, the associated risks or costs involved with making changes are not worth the benefits they could provide.

There are two main needs for the COTS Management System that are both driven by NIMA’s desire to conform to the CMM. The primary need is to have a management tool for the COTS change process that can be used by many people at different locations to coordinate COTS change activities. It is necessary to ensure that certain mandatory steps and procedures are followed prior to any changes to the COTS configuration. All changes should be requested by engineers, they should be properly evaluated and tested, and the appropriate installation procedures should be followed. There is a need to coordinate COTS change activities amongst many people at the three main sites, and information about the progress of a COTS change should be readily available. There is a need to track and status the COTS change process such that managers can take corrective action as needed. The tool serves as an intervention on the organization, and it also serves to measure performance.
There is also a need for historical recording of all major activities associated with a COTS change request. Therefore, historical accounting for all requests, whether installed to the DPS baseline or not, should be tracked and available for reference. There is a need to maintain COTS change information in a central repository where it can be readily accessed by those who need it, from many different locations. Queries against data in the system will provide users with valuable insight into the decision processes that led to the particular COTS change (or the failure to implement a change).

**SYSTEM OPERATIONAL CONCEPT**

The purpose of the COTS Management System is to facilitate the control, coordination, and tracking of the COTS change process. The system acts as a management tool to implement the COTS management process, which is the conglomeration of management interventions that are applied to the “work process” (the COTS change process).

Users will enter data with respect to the various COTS change activities that are applicable to the DPS. The system should facilitate entry of a COTS Specific Change Request [CSCR], which is the base record for all controlled COTS software changes (new COTS insertions or COTS version upgrades). The system will facilitate tracking the various activities associated with a CSCR throughout its life, and indicate if and when the
change was installed. Information pertaining to CSCRs shall be available to the users via specific queries and reports.

The new system shall itself be based on standard COTS software, and provide a user-friendly environment. The system shall employ current industry standards and utilize technology such that data is available to the user via local mini-computers. Data should be stored in a manner that could be easily migrated to new systems should the need arise.

**Feasibility Study**

Though there are many feasible alternatives to implement this solution, NIMA direction, along with current technological and economic constraints, limit the scope of possibility. The system at a high level must consist of one or more database servers. The system should support a multitude of clients that request database services. The system should implement a standard Database Management System (DBMS) to store data. To define the design space, the following five main components are listed along with some of the possible technologies for each component:

1. **Server Component (Hardware)** - IBM Mainframe, DEC Vax, Sun server, RISC 6000 server, Pentium PC.

2. **Server Component (Operating System)** - MVS, VMS, UNIX (or Solaris), HP-UX, Novell, Windows NT (Server), Banyan VINES.
3. **Client Component (Hardware)** - IBM terminal, DEC VAX terminal, DEC Alpha (or other RISC 6000), Sun client, Pentium PC, Macintosh PC.

4. **Client Component (Operating System)** - UNIX (or Solaris), HP-UX, Windows NT (Workstation), Windows 95 (DOS), Macintosh System 7.

5. **Database Component** - DB2, Ingres, Lotus Notes, MS Access, MS Foxpro, Filemaker Pro, Oracle, Sybase, Informix.

   Though this seems like a large realm of possible alternatives, the following parameters limit the scope of the design:

1. The Department of Defense [DOD] has currently endorsed Windows NT as its standard operating system; NIMA is planning to follow this direction.

2. This project is limited to $150,000 for a five (5) year life cycle.

3. A myriad of client PCs are in place or being purchased for NIMA use on the Unclassified Network. Existing Macintosh machines are being replaced by new Pentium PCs.

4. NIMA has MS Access and Filemaker Pro installed as standard software on most client PCs delivered within the past year. Oracle is the most predominant high-performance Relational Database Management System [RDBMS] used within the NIMA and DOD.
5. A RDBMS offers the greatest flexibility for working with multiple related objects and is the standard model used within NIMA and DOD.

There are also several ways to implement the client-server database. Using a network RDBMS application to run on the server is depicted in Figure 2-2. Here, the data and the application reside on the central server and clients connect and run the application on the server. The advantage here is a simpler design for operation and maintenance, since everything is in a central place. The disadvantage is in performance degradation since all the clients run the application on the server. Server performance is also a function of the server hardware, so this disadvantage can be overcome at a price.
Another possibility is to have only the data on the server, while the RDBMS application runs locally on each client. This is depicted by Figure 2-3, and the advantage to this configuration is increase server performance. The disadvantage is system maintenance becomes a distributed task.
Figure 2-3 Feasibility Study (Client Application).

A variation of the configuration shown in Figure 2-3 above is to use a back-end RDBMS to store the data on the server and another front-end RDBMS to store the interface objects (queries, forms, and reports) on the client. In this configuration, the front-end and back-end databases do not have to be the same application as long as they are compatible (Standard Query Language [SQL] and Open Database Connectivity [ODBC] compliant). The advantage here is that high performance RDBMSs like DB2, Oracle, Informix, and Sybase can be used on the back-end while user-friendly and familiar products like MS Access and MS Foxpro can be used at the front-end. This is the concept that will be assumed for the initial operational and maintenance concepts.
Given the realm of possibility and the constraints listed above, Table 2-1 lists the initial system configuration along with the rational for each choice. The configuration outlined in Table 2-1 provides a starting point as a technological approach to the design. It may be modified during the following stages of the life cycle as more information becomes known.
<table>
<thead>
<tr>
<th>Component (Hardware)</th>
<th>Choice</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pentium PC</td>
<td></td>
<td>sufficient performance at lowest cost; simpler maintenance</td>
</tr>
<tr>
<td>Windows NT</td>
<td></td>
<td>NIMA direction</td>
</tr>
<tr>
<td>Pentium PC</td>
<td></td>
<td>sufficient performance; readily available (no cost)</td>
</tr>
<tr>
<td>Windows NT {primary}</td>
<td></td>
<td>NIMA direction; NT Workstation not readily available</td>
</tr>
<tr>
<td>Window 95 {secondary}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Database (Application)</td>
<td>MS Access (Front-end)</td>
<td>Filemaker Pro not relational; MS Access exists; Oracle is the most prevalent high-performance RDBMS in use at NIMA and DOD; provides sufficient functionality at low cost</td>
</tr>
<tr>
<td>Oracle (Back-end)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Database (Location)</td>
<td>Data &amp; Interface - Server</td>
<td>Maintainability</td>
</tr>
</tbody>
</table>

With the Feasibility Study completed, the next focus will be on the system requirements for the COTS Management System.
SYSTEM OPERATIONAL REQUIREMENTS

MISSION DEFINITION

1. Prime Operating Mission - The COTS Management System has the primary mission of serving as the management tool for the COTS change process. The system supports the insertion of new COTS software and the upgrades of existing COTS software in a controlled, managed fashion.

2. Secondary Mission - The system provides a historical record of all major activities associated with a CSCR’s existence. Hence, historical accounting for all CSCRs, whether installed to the DPS baseline or not, shall be available via the COTS Management System.

3. Alternate Mission - This system shall serve as a prototype for the replacement of NIMA’s Automated Configuration Management System [ACMS], the primary system used for all changes to the DPS baseline. The current ACMS has been identified as in need of replacement, and this system shall serve as a small-scale model for the system that will replace ACMS.
PERFORMANCE AND PHYSICAL PARAMETERS

System Performance

1. The system shall be accessible to any PC client attached to the NIMA unclassified network.

2. All network connectivity equipment and operations shall be provided as Government Furnished Equipment [GFE] by NIMA. This includes existing client PCs and network printers.

3. The time to connect to the database shall not exceed 10 seconds from any client location.

4. The system shall incorporate a standard COTS relational database management system (RDBMS).

5. The system shall allow for at least twenty-five (25) concurrent users viewing or inserting new rows to the database.

6. The system shall allow for at least five (5) of the users to make updates to different database tables or rows simultaneously.

7. The system shall provide for record locking to ensure that two users do not make simultaneous edits to the same database row.
8. The system shall allow security of all database objects to include creation of user accounts and administrative groups.

9. The system shall permit the setting of specific permission for each database object.

10. The system shall support the creation and storage of CSCRs, Candidate Product (Alternatives) Reports, Memos, Impact & Recommendation Reports, and Test Summary Reports.

11. All database tables will have primary keys as numeric indexes (only integer or counter types).

12. Memos shall incorporate the ability to store linked objects via Object Linking and Embedding (OLE).

13. The system shall support the assignment of Task Responsibilities (Actions) to organizations and/or individuals for the work of various CSCR related activities. These tasks will be assigned due dates with controlled modification.

14. The system shall allow specific users to enter Task Responses to indicate status on particular Task Assignments (Actions).

15. The average time to input a new record into the database shall not exceed 3 minutes.

16. The system shall constrain data input selections where appropriate to ensure consistent information from queries.
17. The system shall support data structures for the purpose of generating metrics for the 

*COTS change process.*

18. The system will flag major CSCR activities with logical (Yes/No) data types as needed 
to facilitate queries.

19. The system shall have support queries by user name, organization, site, sub-system 
(segment), platform, product name, and CSCR number.

20. Responses to database queries shall not exceed 5 seconds.

21. The system shall utilize SQL and be ODBC compliant.

22. The time to create and print a CSCR Summary Report using the system shall not 
exceed an average of 20 seconds.

23. The time to create and print a COTS Action Assignment Report using the system shall 
not exceed an average of 30 seconds.

24. The system shall employ user-friendly object-oriented design employing help screens 
and menu windows. The system shall adhere to human factors engineering standards 
for 5th - 95th percent (anthropomorphic).

25. Data input shall be via mouse and keyboard.

26. The system shall employ automatic back-up procedures to protect data.
27. The system shall be designed to allow future system upgrades and modifications.

28. During normal system operations, the main database tables shall be replicated to the two remote sites at least every two hours.

Physical Parameters

1. The front-end system shall be capable of being operated from standard PC client workstation in a space equivalent to a average size business desktop (3’ by 4.5’).

2. The back-end system shall be capable of being operated from standard PC server in a space equivalent to a large size business desktop or standard storage rack (5’ by 5’).

3. The database shall utilize less than 1 GB of storage for both operational and maintenance versions.

Use Requirements

1. The system shall be operational from 0600 to 1900 EST.

2. The anticipated system use profile is shown in Figure 2–4.

3. The system shall be capable of operation at any time with the exception of maintenance periods.

4. Maintenance shall be scheduled after normal business hours whenever feasible.
OPERATIONAL DEPLOYMENT AND DISTRIBUTION

1. The system shall be operated at the three main NIMA mapping production facilities located in Reston, Virginia; Bethesda, Maryland; and St. Louis, Missouri. A diagram of the system distribution in shown in Figure 2-5.

2. The main system server shall be located at the Reston site and it shall be accessible via the Sensitive Unclassified Information Network [SUI].

3. The database shall be replicated from the main server to servers at the St. Louis and Bethesda sites every two hours of normal system operation, as defined above.
OPERATIONAL LIFE CYCLE

The system is anticipated to be utilized for the next five years; therefore, it should be designed such that it can be maintained until the year 2002 and problem associated with “Century Rollover” shall be avoided. In the year 2002, it is expected that all pertinent data will be migrated to a new system; hence, the system should conform to common standards to allow for relatively simple data migration.
EFFECTIVENESS FACTORS

1. Operational Availability - System availability during defined operational hours shall be at least 0.975.

2. Dependability - During operational periods (not including scheduled maintenance periods), reliability shall be at least a rating of 0.990.

3. Mean Time Between Maintenance [MTBM] - 500 hours of system use.

4. Mean Time Between Failures [MTBF] - 2,500 hours of system use.

5. Mean Time to Repair [MTTR] - Not to exceed ½ hour.

6. Maintenance Downtime [MDT] - Not to exceed 1 hour in any one maintenance period.

7. Full Database Backup Time - Not to exceed 20 minutes.

8. User Skill Level - Users shall have basic computer operation skills consisting of keyboard data entry, mouse operation, and use of a software applications that are Graphical User Interface [GUI] based.

9. System Maintainer Skill Level - System maintainers shall be skilled in the use of the selected RDBMS(s) and other supporting software applications.
ENVIRONMENT

The system shall be accessible to NIMA’s unclassified network. A standard business office environment shall exist for the system workstations. No abnormal environmental conditions are applicable.

SYSTEM MAINTENANCE CONCEPT

ANTICIPATED LEVELS OF MAINTENANCE SUPPORT

The COTS Management System is anticipated to require limited organizational, intermediate, and depot level of maintenance. The relationship amongst these hierarchical levels of maintenance is depicted by Figure 2-6 below.
Organizational Maintenance

Organizational maintenance is the lowest level of support and is performed locally at each system site by system operators (NIMA and/or contractor with no special training on the COTS Management System). The system operator is responsible for all activities
at this level to include limited maintenance and trouble-shooting. The organization will perform the following organizational maintenance tasks:

- Equipment cleaning (keyboards, mouse, and display screen).
- Visual inspection of equipment (cable inspection, display performance, network performance, and disk drive performance).
- External Adjustments (display brightness and other display parameters).
- Basic workstation trouble-shooting limited to the operating system or other standard office software.

The system operator maintenance requirements involve mostly basic standard preventative maintenance actions. They also perform some basic trouble-shooting on the workstations. The system operator notifies the intermediate maintenance organization of identified problems requiring attention beyond their scope of responsibility.

**Intermediate Maintenance**

The contractor is responsible for the intermediate maintenance support. A majority of system maintenance activities are anticipated to occur at this level. Intermediate maintenance is comprised of a team that is normally at the Reston site (if off-site, they will be available to travel to the system site if required). It is expected that much of the support can be performed remotely from the Reston facility (via the network and telephone/E-mail).
The contractor will perform the following intermediate maintenance tasks:

- Detailed Inspections (network connections, network performance parameters).
- Removal and Replacements of Components (installing a new disk drive device, keyboard, mouse, or display screen).
- Major Service (replacing a complete server or workstation, replacing a network hub connection, database changes).
- Complicated Adjustments (Internet protocol addressing changes, network physical changes).
- Scheduled backups of database files (to a removable media).
- Scheduled runs of anti-virus scanning software (including updating virus definitions monthly).
- Diagnostic Tools (use of utility programs to increase performance).

This team is responsible for system setup, initialization, diagnostics, calibration, trouble shooting, anti-virus scans, and end item replacement. They must also perform routine database backups to secure the system against data loss during failure. Additionally, the team coordinates all higher level maintenance requirements to include logistics requirements, commercial product vendor support, and software engineering support.
Depot Level Maintenance

This level actually consists of two different types of support to include, commercial product vendor support and software engineering support. Since the majority of the system software and hardware components are acquired commercially, these vendors must provide the system support for their respective components. The intermediate maintenance level coordinates all of these activities through the ILS group.

The other main support is for the software which was designed by the contractor. All contractor software problems must be addressed at this level by contractor software engineers. Additionally, contractor configuration engineers handle integration problems with commercial products and company software.

Responsibilities for Support

The ILS group is responsible for the logistics and supply support of the COTS Management System. The ILS group will receive requirements to maintain adequate supplies on hand to meet mean time to repair requirements and basic logistic supply readiness to maintain the system operational.
REPAIR POLICY

The basic repair policy is to completely replace any failed network server or workstation and then off-line repair the failure at an on-site maintenance area. This policy attempts to maintain the system operational through the availability of complete spares at each of the three facilities. All maintenance and repair should occur on-site whenever possible. Each system workstation must be accessible at all times by maintenance or repair personnel.

EFFECTIVENESS RESPONSIVENESS

Database Server Responsiveness - Any cause of failure to the main database server (network problems, RDBMS problems, etc.) should be corrected within 30 minutes. If necessary, one of the replication sites should be temporarily “switched over” to become the active central RDBMS site. Once a “switch over” has occurred, the new central site will retain responsibility for data integrity for the remainder of the day. At that time, a coordinated transfer will occur such that no data items are lost.

Supply Responsiveness - Spare workstations are stored on site for immediate and readily available use. Office supplies are also available and maintained at adequate levels on site for immediate use. Replacement of failed components should occur within 5 working days from the component vendor. All replacements are coordinated via ILS.
**MAINTENANCE ENVIRONMENT**

The system servers shall be housed within the Data Services areas at each site. The Data Services area environment is a standard office environment with adequate lighting, ventilation, and cooling/heating. Therefore, no unusual limitations are placed on temperature, lighting, humidity or vibration conditions. Transportation limitations are restricted to constraints normally adhered to when transporting personal computers.

**SYSTEM SPECIFICATION**

From the operational and maintenance system-level requirements, a system specification “Type A” is created to provide the basis for the preliminary design. For this project, a partial system specification is comprised of a combination of the operational and maintenance requirements previously generated. They are outlined in Table 2-2 below.
<table>
<thead>
<tr>
<th>Requirement</th>
<th>Design Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prime Operating Mission</td>
<td>Serves as the management tool for the <em>COTS change process</em></td>
</tr>
<tr>
<td>Secondary Mission</td>
<td>Provides a historical record of all major activities associated with a CSCR’s existence</td>
</tr>
</tbody>
</table>
| Physical Parameters           | • The front-end system shall be capable of being operated from standard PC client workstation in a space equivalent to a average size business desktop (3’ by 4.5’).  
                                 | • The back-end system shall be capable of being operated from standard PC server in a space equivalent to a large size business desktop or standard storage rack (5’ by 5’).  
                                 | • The database shall utilize less than 1 GB of storage for both operational and maintenance versions. |
| Normal Operation Hours        | 0600 to 1900 EST                                                                 |
| Operational Deployment        | Operated at the three main NIMA mapping production facilities located in Reston, Virginia; Bethesda, Maryland; and St. Louis, Missouri |
| Operational Life-Cycle        | • Through the year 2001  
                                 | • All “Century Rollover” problems avoided  
                                 | • Data easily migrated to new system |
| System Performance            | (as detailed above)                                                              |
| Operational Availability      | ≥ 0.975                                                                          |
| Dependability                 | Reliability ≥ 0.990                                                              |
| MTBM                          | ≥ 500 hours                                                                      |
The system specification shown above is only a small part of a truly complete system specification. According to Blanchard and Fabrycky (pp. 48-51), the system specification should include:

1. General description of the system and its function.

2. Operational requirements.

3. Maintenance concept definition.

5. Performance characteristics.

6. Physical characteristics.

7. Effectiveness characteristics.

8. Design characteristics.


10. Logistic support.

11. Design documentation.


The majority of this information exists in previous sections of this chapter; however, it will not be compiled into a complete formal system specification for the sake of brevity on this project.

**Technical Performance Measures**

In an attempt to balance life-cycle cost with system effectiveness, Technical Performance Measures [TPMs] are developed based on technical design parameters that can be modified to accommodate the customer’s system requirements. The TPMs are
quantitative measures which can be assigned a weighed value based on their relative importance to the overall system. Hence, TPMs establish a prioritization for all subsequent system development efforts. Resources are aligned where needed most, and various engineering and support groups are assigned responsibility for each TPM. Based on the system specification outlined above, the following TPMs are established as shown in Table 2-3 below.
<table>
<thead>
<tr>
<th>TECHNICAL PERFORMANCE MEASURE (TPM)</th>
<th>QUANTITATIVE REQUIREMENT</th>
<th>RELATIVE IMPORTANCE</th>
<th>RESPONSIBLE ENGINEERING DISCIPLINE(S)</th>
<th>SUPPORTING ENGINEERING DISCIPLINE(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIFE-CYCLE COST</td>
<td>$39,000/YEAR (5 YEARS)</td>
<td>25%</td>
<td>SYSTEMS ENGINEERING</td>
<td>LOGISTICS FINANCE</td>
</tr>
<tr>
<td>DATABASE CONNECTION TIME</td>
<td>10 SECONDS (MAX)</td>
<td>5%</td>
<td>SYSTEMS ENGINEERING</td>
<td>MAINTAINABILITY</td>
</tr>
<tr>
<td>CONCURRENT USERS (VIEW &amp; INSERT ROWS)</td>
<td>25 (MIN)</td>
<td>5%</td>
<td>SOFTWARE ENGINEERING</td>
<td>SYSTEMS HUMAN FACTORS</td>
</tr>
<tr>
<td>CONCURRENT USERS (EDIT EXISTING ROWS)</td>
<td>5 (MIN)</td>
<td>2%</td>
<td>SOFTWARE ENGINEERING</td>
<td>SYSTEMS HUMAN FACTORS</td>
</tr>
<tr>
<td>NEW RECORD INPUT</td>
<td>3 MINUTES (MAX)</td>
<td>3%</td>
<td>SOFTWARE ENGINEERING</td>
<td>SYSTEMS HUMAN FACTORS</td>
</tr>
<tr>
<td>DATABASE QUERY RESPONSE</td>
<td>5 SECONDS (MAX)</td>
<td>6%</td>
<td>SOFTWARE ENGINEERING</td>
<td>SYSTEMS HUMAN FACTORS</td>
</tr>
<tr>
<td>CSCR SUMMARY REPORT (CREATE &amp; PRINT)</td>
<td>20 SECONDS (MAX)</td>
<td>3%</td>
<td>SOFTWARE ENGINEERING</td>
<td>SYSTEMS HUMAN FACTORS</td>
</tr>
<tr>
<td>ACTION ASSIGNMENT REPORT (CREATE &amp; PRINT)</td>
<td>30 SECONDS (MAX)</td>
<td>3%</td>
<td>SOFTWARE ENGINEERING</td>
<td>SYSTEMS HUMAN FACTORS</td>
</tr>
<tr>
<td>AVAILABILITY (OPERATIONAL 6AM - 7PM)</td>
<td>97.5% (MIN)</td>
<td>8%</td>
<td>SYSTEMS ENGINEERING</td>
<td>MAINTAINABILITY</td>
</tr>
<tr>
<td>DEPENDABILITY</td>
<td>99.0% (MIN)</td>
<td>5%</td>
<td>SYSTEMS ENGINEERING</td>
<td>MAINTAINABILITY</td>
</tr>
<tr>
<td>SIZE OF CLIENT COMPONENT (PC)</td>
<td>3 x 4.5 FEET (MAX)</td>
<td>1%</td>
<td>SYSTEMS ENGINEERING</td>
<td>HUMAN FACTORS</td>
</tr>
<tr>
<td>SIZE OF SERVER COMPONENT (PC)</td>
<td>5 x 5 FEET (MAX)</td>
<td>1%</td>
<td>SYSTEMS ENGINEERING</td>
<td>HUMAN FACTORS</td>
</tr>
<tr>
<td>MAINTAINABILITY (TIME FOR FULL SYSTEM BACKUP)</td>
<td>20 MINUTES (MAX)</td>
<td>3%</td>
<td>MAINTAINABILITY ENGINEERING</td>
<td>SYSTEMS HUMAN FACTORS</td>
</tr>
<tr>
<td>MAINTAINABILITY (MTTR)</td>
<td>30 MINUTES (MAX)</td>
<td>10%</td>
<td>MAINTAINABILITY ENGINEERING</td>
<td>SYSTEMS HUMAN FACTORS</td>
</tr>
<tr>
<td>MAINTAINABILITY (MTBF)</td>
<td>2,500 HOURS (MIN)</td>
<td>6%</td>
<td>MAINTAINABILITY ENGINEERING</td>
<td>SYSTEMS SOFTWARE</td>
</tr>
<tr>
<td>MAINTAINABILITY (MTBM)</td>
<td>500 HOURS (MIN)</td>
<td>4%</td>
<td>MAINTAINABILITY ENGINEERING</td>
<td>SYSTEMS SOFTWARE</td>
</tr>
<tr>
<td>HUMAN FACTORS (ERROR RATE/YEAR)</td>
<td>2% (MAX)</td>
<td>10%</td>
<td>HUMAN FACTORS ENGINEERS</td>
<td>SYSTEMS MAINTAINABILITY</td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td>100%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

These TPMs will be taken to the next level of design. Though they provide a beginning frame of reference, they may need to be adjusted to meet customer requirements as more data becomes available.
CHAPTER 3: APPLICATION OF MANAGEMENT SYSTEMS ENGINEERING

Use of the management systems engineering methodology, as detailed by Kurstedt in Management Systems Theory, Applications, and Design, to analyze and understand the types of data and information that are needed will help to ensure that the right kinds of tools are developed to support the COTS change process. “A good management tool in the wrong situation will fail. You need to make sure the tools you have fit you and your operation.” (Kurstedt, p.580).

By studying the COTS change process using the principles of management systems engineering, it becomes possible to develop a COTS management process that can effectively provide the necessary control and coordination. By using management systems engineering methodology to identify and understand the COTS change process, it becomes possible to develop the tools that are required to facilitate the COTS management process. In the end, the COTS Management System will be the automated implementation of those management tools that are identified via management systems engineering methodology.
BUILDING MANAGEMENT TOOLS

Designing tools to manage and perform tasks, gather information, and make decisions is an extremely important step in designing an automated COTS Management System. A tool can be as simple as a meeting or a status report to something as complicated as a detailed step by step procedure. Each tool within an organization has a unique purpose and goal. The commonality between all tools used within the organization is that every tool has the purpose of helping personnel perform tasks more efficiently and with absolute reliability.

There are various methods used to develop and design tools, but for this project the management systems engineering methodology is adopted. The application of the Management System Model [MSM] (Kurstedt, p. 201) is used in the development of management tools for the COTS change process. By understanding the various components of the MSM, the proper tool set and implementation can developed. A diagram of the MSM (in expanded form) is shown in Figure 3-1 below.
Figure 3-1 Expanded Management System Model. (Taken from Kustesdt, p.201).

By working with the three components of the MSM: *Who Manages, What is Managed, and What is Used to Manage;* the COTS change process can be defined in
terms of manageable pieces. Of particular interest for this project is the lower portion of the diagram in Figure 3-1, *Information-Oriented Performance*. This is where data are transformed into information; which is then portrayed to decision makers. Subsequently, focus on the interfaces to either side of the *What is Used to Manage* portion of the MSM (Measurement - Data; Information Portrayal - Information Perception) is of key interest to developing adequate tools to implement a *COTS management process*. The overall goal is to have a positive impact on *Organizational Performance*, which encompasses all of the MSM (as shown in the figure above).

In developing an understanding of the *COTS change process*, a management tool known as the Work Breakdown Structure will be employed. This tool will serve to define the generic tasks that are performed for any COTS change that is applied to the DPS. According to Kurstedt (p.1050), “The work breakdown structure serves as the basis for all following project management tools by organizing tasks into related groups.” Hence, application of this tool helps set the stage for all other tools which will be subsequently developed. The use of this tool also marks the beginning of the application of Management System Analysis, which will be discussed more later in this chapter.

**DEFINING THE COTS CHANGE PROCESS**

The first step in being able to support any work process is to define it in a concise and understandable manner. As stated above, the *COTS change process* (or the methods
for implementing changes to COTS products) is all the steps it takes to either: (1) evaluate a request for a new COTS software package, choose among potential alternatives, evaluate the impacts, and smoothly test and install the new software into the DPS; or (2) evaluate the impacts of a COTS version upgrade and smoothly test and install the new software into the DPS. By defining the COTS change process utilizing the work breakdown structure, it can be understood to a degree from which tools can be developed to implement the COTS management process. To provide further detail, the work breakdown structure will be embellished with information from the MSM.

**Work Breakdown Structure for the COTS Change Process**

The following is the rough work breakdown structure for the COTS change process. It is detailed with information specific to the MSM such that it can be used to develop the necessary tools to implement a COTS management process.

[0] High Level COTS Change Process

The high level COTS change process is depicted by Figure 3-2 below.
Figure 3-2 The COTS Change Process.
Purpose: To determine if and when COTS changes should be made; to decide what the change should be; and if necessary, to smoothly install it to the operational environment.

Description: When possible, DPS functionality is upgraded by the insertion of COTS products to replace existing proprietary software. This supports the NIMA effort to migrate to open systems. Upgrades of existing COTS products are also included in this process.


What is Managed: The COTS change process.

What is Used to Manage: Computer data, meetings, telephone conversations, etc.

Sub-Activities:

[1.0] New Product Requests.


[4.0] Product Installation.

[1.0] New Product Requests - Searching for Alternatives*

This task is depicted pictorially in Figure 3-3 below.
Figure 3-3 Function 1: New Product Requests.

Purpose: The purpose of this task is to narrow the field of candidate products to the single one that both meets the user’s requirements and will most effectively support the
DPS. The product chosen should also conform to NIMA policy for procurement of COTS products. [*For COTS version upgrades, Task 1.0 is bypassed.*]

**Description:** This is the initial study and selection phase of the process. Potential products are carefully reviewed against system requirements before one is chosen and recommended to continue in the *COTS change process.*

**Who Manages:** Systems Engineers from the Maintenance Engineering and Test [ME&T] Organization.

**What is Managed:** The definition of system requirements for the COTS product; the search for feasible alternatives; and the selection of the “best fit” COTS product.

**What is Used to Manage:** Change request, ILS database, requirements matrix, computer (Internet), periodicals, telephone (contacts), product literature, etc.

**Sub-Activities:**


[1.2] Select “Best Fit” COTS.

**[1.1] Research Alternative COTS Products**

**Purpose:** The purpose of this step is to find COTS products that will meet the requirements stated in the request. This step includes searching the ILS Database to see if the requested product (or one that meets the requirements) is already on hand.
Description: Extensive research is done to create a list of possible products. A subset list of products is then chosen for a more detailed investigation. This investigation may require the use of techniques such as the Internet or company demonstrations.

Who Manages: Systems Engineers from ME&T.

What is Managed: The definition of system requirements for the COTS product and the search for feasible alternatives.

What is Used to Manage: Change request, ILS database, requirements matrix, computer (Internet), periodicals, telephone (contacts), product literature, product demonstrations.

[1.2] Select “Best Fit” COTS

Purpose: The purpose of this step is to select the “best fit” COTS product by making an unbiased decision based upon measurable criteria and the available information for each product that is being considered.

Description: A set of measurable criteria that the decision should be based upon must be defined. Relative weight factors are then assigned to each of the criteria based upon the importance of each. Once the information for each product and each criteria is obtained, a score is calculated for each product. This score determines which product fits the defined requirements best.

Who Manages: Systems Engineers from ME&T.
**What is Managed:** The selection of the “best fit” COTS product.

**What is Used to Manage:** Change request, requirements matrix, product literature, decision theory, product ranking matrix.

**[2.0] PRODUCT EVALUATION**

This task is depicted pictorially in Figure 3-4 below.
Figure 3-4: Function 2: Product Evaluation.

**Purpose:** The purpose of the product evaluation is to ensure that all possible impacts of upgrading or procuring the software are considered.
**Description:** All pertinent information is reviewed to determine the possible impacts of the new product/upgrade to the baseline system. Various groups in CMS and NIMA may be tasked with performing analysis of their domain to determine the impacts of the requested COTS change. Impacts to application software, hardware, firmware, interfaces, operating systems, operations, timelines, and so forth must all be considered. The goal is to make a well informed, highly coordinated decision about whether or not to install the COTS change and when and how to best do it.

**Who Manages:** CMS Chief Engineer.

**What is Managed:** The decision for whether or not to perform the change and how and when to do it.

**What is Used to Manage:** Change request, CMS functional managers (coordinating impact analysis and recommendations), product documentation, system documentation, electronic mail, telephone, technical exchange meetings [TEMs], status reports, Memos.

**Sub-Activities:**

1. **[2.1] Assign Tasks to Stakeholders.**
2. **[2.2] Submit Impact & Recommendation Statements.**
3. **[2.3] Consolidate into CMS Recommendation.**
4. **[2.4] Review CMS Recommendation Memo.**
[2.1] Assign Tasks to Stakeholders

**Purpose:** This step is where the functional managers assign the appropriate team members to study the possible impacts of the COTS upgrade or procurement.

**Description:** The functional managers of all areas potentially affected by the change choose the team members most suited to study the COTS product being considered for upgrade or procurement.

**Who Manages:** Functional managers.

**What is Managed:** The assignment for the evaluation of the effects of the requested COTS change.

**What is Used to Manage:** Change request, product documentation, system documentation, electronic mail, telephone conversations, TEMs.

[2.2] Submit Impact & Recommendation Statements

**Purpose:** The purpose of this step is to gather and document information regarding the possible impacts of the upgrade/procurement to the baseline system.

**Description:** Each assigned team member reviews the change request and product information and then makes a statement regarding the possible impacts of the upgrade/procurement. They also make recommendations on if the change should be made and, if so, when and how to best do it.
Who Manages: Functional managers.

What is Managed: The evaluation of the effects of the requested COTS change.

What is Used to Manage: Change request, product documentation, system documentation, electronic mail, telephone conversations, TEMs.

[2.3] Consolidate into CMS Recommendation

Purpose: The purpose of this step is to create a document that summarizes all known impacts and related recommendations for the Chief Engineer to review.

Description: Systems engineers analyze all impact & recommendation statements and form a system level recommendation. This is then summarized in the form of a concise document that can be easily read and understood by CMS management.

Who Manages: Systems Engineers from ME&T.

What is Managed: The system level analysis and recommendation pertaining to the requested COTS change.

What is Used to Manage: TEMs, product documentation, system documentation, electronic mail, telephone conversations, impact and recommendation statements.

[2.4] Review CMS Recommendation Memo

Purpose: To give the Chief Engineer final control of all COTS changes.
**Description:** This step is where the Chief Engineer reviews the system level recommendation and either concurs, denies, or asks for further analysis. Continuing beyond this point implies software will be procured and installed for test.

**Who Manages:** Chief Engineers.

**What is Managed:** The overall recommendation pertaining to the requested COTS change.

**What is Used to Manage:** TEMs, CMS Recommendation Memo, electronic mail, telephone conversations.

[3.0] PRODUCT TESTING

This task is depicted pictorially in Figure 3-5 below.
[3.1] Produce Test Plan

[3.2] Load Product in Test Environment

[3.3] Perform Test Activities

Results acceptable?

[3.4] Produce Test Reports & Recommendation

continue?

Designated Test Group

Supported by Configuration Management & ILS Groups

Figure 3-5 Function 3: Product Testing.
**Purpose:** The purpose of product testing is to install and test the COTS change in the Integrated Development and Maintenance Environment [IDME] to determine if it performs as expected. The goal is to ensure that there is a smooth transition of the COTS change to the Designated System Environment [DSE].

**Description:** Product testing is performed within the IDME to evaluate what impacts the COTS change may potentially have if the COTS product is installed to the DSE. A test plan is produced, the COTS is installed and configured in the IDME, test procedures are performed, and results are appropriately documented. If the testing produces negative results, a recommendation to not install the COTS product may ensue.

**Who Manages:** Designated test group lead (made up of individuals from one or more key functional areas depending upon the complexity of the change).

**What is Managed:** The installation of the COTS in a test environment and testing to determine how the COTS will integrate into the DPS.

**What is Used to Manage:** TEMs, installation schedules and procedures, test procedures, electronic mail, telephone conversations.

**Sub-Activities:**

[3.1] Produce test plan.

[3.2] Load product in test environment.

\[1\] The DSE is also known as the “Operational Environment” or “Production.”
[3.3] Perform test activities.

[3.4] Produce test reports and recommendation.

[3.1] PRODUCE TEST PLAN

**Purpose:** The purpose of this step is to develop a plan for how to execute the COTS testing. This step includes allocating IDME resources via a IDME Resource Scheduling Request [IRSR].

**Description:** The test plan for COTS products will range from informal install and verification to detailed test procedures. In any event, at a minimum it should include how and where to install, how to verify new functionality, how to determine negative impacts, and how to backout changes.

**Who Manages:** Designated test group lead.

**What is Managed:** The creation of the COTS test plan and allocation of test resources.

**What is Used to Manage:** TEMs, installation schedules and procedures, old test plans.

[3.2] LOAD PRODUCT IN TEST ENVIRONMENT

**Purpose:** The purpose of this step is to install and configure the COTS change in the IDME. The goal is for the COTS product to operate as it would within the DSE.
Description: The COTS product is loaded and configured on designated resources (as specified in the IRSR) in the IDME.

Who Manages: Configuration Management [CM²].

What is Managed: The installation of the COTS in a controlled test environment.

What is Used to Manage: TEMs, IRSR, installation schedules and procedures.

[3.3] PRODUCT TEST ACTIVITIES

Purpose: The purpose of software testing is to verify the new COTS functionality and to evaluate the impacts that the COTS change will have on the baseline system.

Description: Testing may range from informal verification to detailed testing of individual threads, depending on the situation. The test procedures should be well documented such that test activities can be reproduced.

Who Manages: Designated test group lead.

What is Managed: The testing to determine how the COTS change will affect the DPS.

What is Used to Manage: Test procedures, test results summary.

² CM manages and controls all facets of the DPS baseline system.
[3.4] PRODUCE TEST REPORTS AND RECOMMENDATION

**Purpose:** This step is necessary to document the test and its results. Additionally, a recommendation to install or not install is made based on the test results.

**Description:** A Test Summary Report is created which should include a brief summary of test activities, a summary of the results, and a recommendation.

**Who Manages:** Designated test group lead.

**What is Managed:** The summarization of test activities and recommendation based on the test results.

**What is Used to Manage:** Test results, test procedures, TEMs.

[4.0] PRODUCT INSTALLATION

This task is depicted pictorially in Figure 3-6 below.
Figure 3-6 Function 4: Product Installation.
**Purpose:** This step is required to transition the COTS change to the DSE in a smooth manner.

**Description:** The COTS change will be submitted to the Engineering Review Board [ERB\(^3\)] for approval; and, once approved, it will be installed to the DSE by CM.

**Who Manages:** Systems Engineers from ME&T.

**What is Managed:** The installation and verification of the COTS change to the DSE.

**What is Used to Manage:** The Automated Configuration Management System [ACMS], Resource Scheduling Request [RSR], schedules, TEMs.

**Sub-Activities:**

- [4.1] ERB Assessment.
- [4.2] Create Installation and Checkout Plan.
- [4.3] Install to DSE.

**[4.1] ERB Assessment**

**Purpose:** This step involves creating a Discrepancy Report [DR] in ACMS and getting the ERB’s approval for the COTS change.

---

\(^3\) The ERB is also referred to as the Engineering Change Control Board [ECCB].
**Description:** The information about the COTS change is used to create a corresponding DR in ACMS. The initial installation and transition information is provided.

**Who Manages:** The ERB.

**What is Managed:** The approval to perform a change to the DPS operational system.

**What is Used to Manage:** ACMS, TEMs.

**[4.2] Create Installation and Checkout Plan**

**Purpose:** To create the COTS installation and checkout plan for CM.

**Description:** The information about the COTS change, installation procedures, and verification procedures are developed. A Resource Scheduling Request (RSR) is also submitted in this step.

**Who Manages:** Systems Engineers from ME&T.

**What is Managed:** The plan for installation and verification of the COTS change.

**What is Used to Manage:** Impact & recommendation reports, test summary reports, product information, Email, TEMs, RSR.

**[4.3] Install To DSE**

**Purpose:** To install the COTS product to the DSE.
Description: The COTS change is transitioned to the DSE in accordance with the plan.

Who Manages: CM

What is Managed: The baseline DSE.

What is Used to Manage: ACMS, TEMs, RSR, CM procedures, telephone conversations.

[4.4] Verify New COTS Functionality

Purpose: To verify the new COTS functionality and to ensure that no negative impacts are occurring.

Description: The user will verify that the new COTS functionality is present and that there are no problems being experienced that can be associated with the COTS change.

Who Manages: Systems Engineers from ME&T.

What is Managed: The verification of the COTS change.

What is Used to Manage: Affected system, procedures, TEMs, telephone conversations.

APPLICATION OF THE MANAGEMENT SYSTEM ANALYSIS

The MSM represents a structured approach for understanding the management process because the MSM provides a simple framework for describing a management
system, its components, and their relationships. Shown in Figure 3-7 below is the model for building management tools which includes the MSM encircled by the functions of Management System Analysis (Kurstedt, p.246).

Figure 3-7 Management System Analysis (Functions for Building Management Tools). [Taken from Kurstedt, p.246].
As shown above, the five management system analysis functions are imposed upon the basic MSM. The five management system analysis functions flow counterclockwise around the basic MSM. To build management tools, the following steps are taken about the MSM:

1. Delimit the domain of responsibility and understand the operation [CCW1].
2. Determine the interventions (decision-action pair) that are needed to improve the work process [CCW2].
3. Define the information that best supports decisions, and the management tool features that are necessary to get the information [CWW3].
4. Deduce the data that comprise the necessary information, and how to measure for those data [CCW4].
5. Determine the key indicators that imply a working operation, the outputs can be used to measure the indicators, and the relationships among the key indicators [CCW5].

In essence, by developing the work breakdown structure and defining the elements of the MSM, much of the initial system analysis has been performed. Next, the elements of *What is Used to Manage* is applied to the *COTS change process* (depicted by Figure 3-2 above) to define the information that is required for the management process. The various "elements of data" for the *COTS management process* are shown as trapezoids in Figure 3-8 below.
Figure 3-8 Management System Analysis of the COTS Change Process.
By defining the necessary interventions on the organization, the basis of functional analysis for the COTS Management System can be established. The COTS Management System is the automated approach to imposing a COTS management process to the changing of COTS products. It should be made clear that not all elements of the COTS management process will be implemented via the COTS Management System; some elements will not be automated because in some cases it is not desirable or feasible. These non-automated tools will be mentioned; however, the focus of this report is on the automated COTS Management System.

**A Holistic View of Management Tools**

The *What is Used to Manage* portion of the MSM (refer to Figure 3-1) was studied to impose a COTS management process on the COTS change process. During subsequent design, management systems engineering is applied to define and develop the implementation of the interventions previously identified. To aid in the Preliminary Design and the Detail Design, the “Conceptual Model for Management Tools and Guides” (Kurstedt, p.380) is utilized.

**The Conceptual Model for Management Tools and Guides**

The “Conceptual Model for Management Tools and Guides” is shown in Figure 3-9 below.
Figure 3-9 Conceptual Model for Management Tools and Guides. (Adapted from Kurstedt, p.380).

The “Conceptual Model for Management Tools” provides a framework for understanding the *What Is Used To Manage* block of the MSM; and it’s focus is on information-oriented performance. The COTS Management System acts as an intervention on the organization, it facilitates the application of the MSM on the organization, and it serves to measure performance; therefore, it should be designed consistent with the “Conceptual Model for Management Tools.” The COTS Management System also provides the feedback loop necessary to facilitate the *COTS management process.*
The diagram in Figure 3-10 is a preliminary attempt to establish relationships amongst the various elements of data (or objects) that were gathered in Chapter 2. The dotted trapezoids represent data objects that are prime candidates as database tables in the COTS Management System. The CSCR is identified as the primary record for the COTS Management System, and all other database objects relate to it. Action Assignments impose the intervention of the COTS Management System on the organization. Note that Status Report and Information Queries are dashed, to represent that they are actually outputs derived from the other data elements. These outputs will actually be used to measure performance and apply feedback in the form of further management interventions. Elements in shaded trapezoids are interventions that are presently identified to be implemented with means other than the COTS Management System (references to their associated objects are shown with dashed lines with small-headed arrows). For instance, the ACMS DR number would be a column in the CSCR table, providing a cross-reference. The Decision Matrix would be referred to (or even embedded) in a Memo object.
Figure 3-10 Establishing Relationships Among Data Objects.
The "Conceptual Model for Management Tools" is important because as design progresses, the need for information and performance measurements becomes a key concept in the database structure. For example, the need to capture activity dates is necessary to measure elapsed times for activities. An important measure of effectiveness for any activity is the length of time it takes to be completed. By understanding the desired outputs from the tool, the proper data elements can be identified and incorporated into the database design.

With the relationships in Figure 3-10 established and the "Conceptual Model for Management Tools" in mind, attention is next turned to preliminary system design and the functional analysis of the COTS Management System.
CHAPTER 4: PRELIMINARY SYSTEM DESIGN

PRELIMINARY SYSTEM DESIGN OVERVIEW

The preliminary system design phase is the next step after the conceptual system design work of Chapter 2 and the application of management systems engineering of Chapter 3. The preliminary system design phase uses the technical baseline established as the “System Specification” and begins the design process. The preliminary system design activities are shown in Figure 4-1.
Figure 4-1 Preliminary System Design Phase (Overview). (“Adapted from Blanchard and Fabrycky, p.56.”)
The preliminary system design process primarily performs a system functional analysis and allocation of system level requirements to subsystem units (or modules). First, a functional analysis is performed on the operational and maintenance functional areas of the system to identify the major areas of concern to the development of the COTS Management System. Second, top-level system requirements are allocated to the modules that comprise the COTS Management System. Finally, a trade-off and optimization analysis is performed to evaluate specific design alternatives available to meet the allocated requirements.

**FUNCTIONAL ANALYSIS**

The functional analysis for the system development cycle is a useful mechanism to begin understanding the detailed functionality of the COTS Management System. Of particular interest is the operational and maintenance functional flow diagrams because they help to define the working system. By understanding what the system is to do from a functional perspective, design becomes a matter of taking the necessary actions to create the specified functionality within the previously defined constraints. All requirements and TPMs must be met and validated through the actual design, testing, and distribution phases. A partial functional flow analysis is included here as part of this project.
TOP-LEVEL FUNCTIONAL FLOW ANALYSIS

The diagram in Figure 4-2 depicts the top-level functional flow for the COTS Management System life cycle through system O&M. Block 1 in the system functional flow defines the system operational requirements as determined in Chapter 2. Block 2 pertains to the feasibility study that was performed as part of the conceptual system design in Chapter 2. Blocks 3 and 4 show the design of the system and support elements, which is part of this Chapter and Chapter 5. Block 5 shows the system integration and test steps necessary to validate the system against the defined requirements. The actual procurement and production of the complete system is represented by Blocks 6, 7, and 8. The distribution and setup of the system in the customer’s environment is shown in Block 9. Finally, Blocks 10 and 11 show the operation and maintenance of the final system.
Figure 4-2 COTS Management System Top-Level Flow.
By further decomposing the activities in Block 10 and 11, the functional Operation and Maintenance activities of the system will be defined from the functional perspective.

For the sake of brevity in this report, only a subset of the operational and maintenance functionality will be decomposed to lower levels.

**Operational Functional Flow Analysis**

Shown below in Figure 4-3 is the first-level operational flow for the COTS Management System. This level is defined using RDBMS concepts, since all operational functionality is related to the database. The major operational functions are: (1) inserting new data, (2) editing existing data, (3) querying data, (4) creating reports, and (5) performing security functions. These are the main operational functions necessary to implement the previously defined conceptual design.
Figure 4-3 First-Level Operational Flow Diagram.
In Figure 4-4, the inserting new data function [10.1] is further decomposed into sub-functions. Here, the data objects previously identified using management systems engineering methods are created using the COTS Management System. These data objects are (1) COTS System Change Request [CSCR], (2) Impact & Recommendation Statements, (3) Test Summary Reports, (4) Memos, and (5) Action Assignments. This level will be useful in the initial definition of many of the main database objects.
Figure 4-4. Second-Level Operational Flow Diagram.
At the third-level, the operational flow of creating a CSCR is broken out as shown in Figure 4-5. Basically, changes are either: (1) the result of a new COTS product version being available, or (2) a request for a new product. Updates in the ILS database “trigger” the creation of a new CSCR to evaluate the COTS version upgrade (the trigger would ideally be automated, but could also be implemented manually). When a request for a new COTS product is received, a ILS database query is initiated to see if the COTS product already does exist on the system (this is a check to ensure that the request is not really for an updated version). If the query returns a result (the COTS product exists), the entry of CSCR data ensues. If the request is truly for a new product (COTS doesn’t exist), then the COTS Requirements must be defined. This is a process of balancing the users requirements, DPS system requirements, and NIMA’s long-term objectives to produce a consolidated requirements list. For each alternative that is researched and/or demonstrated, a Product Information Report is generated. Next is the evaluation and selection steps, where the product that best satisfies the defined COTS products is chosen. The selected product becomes the basis for the new CSCR, which is finally created for the new COTS request.
Figure 4-5 Third-Level Operational Flow Diagram.
MAINTENANCE FUNCTIONAL FLOW ANALYSIS

Shown below in Figure 4-6 is the first-level system maintenance flow. The maintenance functions are broken down into the following main activities: (1) database maintenance, (2) client system trouble-shooting, (3) server system trouble-shooting, and (4) hardware maintenance. Here, hardware maintenance is defined as coordinating the replacement and repair of hardware end-items and network components.
Figure 4.6 First-Level Maintenance Flow Diagram.
Figure 4-7 depicts the second-level maintenance flow for the conduct database maintenance function \[11.1\]. The activities included here are performance of: (1) table structure modifications, (2) interface modifications, (3) data maintenance, (4) database backup and replication, and (5) database recovery. These functions help define the system structure from a maintenance perspective. An example is that by grouping the data structure and the user interface in separate modules, the maintenance function becomes more manageable.
Figure 4-7 Second-Level Maintenance Flow Diagram.
Next the table structure modification block [11.1.1] is decomposed into the following activities: (1) select the appropriate database, (2) select the table, (3) enter design mode, (4) select the appropriate columns, (5) modify the necessary data attributes, (6) modify relationships if necessary, (7) check referential integrity and data, and (8) re-link the modified table back to the interface. These steps are depicted in Figure 4-8 below.
Figure 4-8 Third-Level Maintenance Flow Diagram.
FUNCTIONAL REQUIREMENTS ALLOCATION

After performing functional analysis on the operational and maintenance flows for the COTS Management System, the TPMs for the system level requirements are allocated amongst the various system components. Figure 4-9 below shows how the previously defined TPMs are allocated to the hardware, software, and system support components of the COTS Management System.
Figure 4-9 Allocation of Functional Requirements.
Next, the software component is broken into modules based on functionality. This is a straightforward process because the system is based on a COTS database. The data and interface objects are separated to allow ease of maintenance as suggested by Jennings in *Using Access 2 for Windows*. The security module and the operating system are also logically separate modules. Shown below in Table 4-1 is the functional modules for the COTS Management System.

**Table 4-1 Software Functional Modules.**

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Tables</td>
<td>Contains the database table structure and data elements.</td>
</tr>
<tr>
<td>User Interface</td>
<td>Contains queries, forms, reports, macros, and menus used by all users of the COTS Management system.</td>
</tr>
<tr>
<td>Security</td>
<td>Contains permission for system users and groups for all database objects.</td>
</tr>
<tr>
<td>Operating System</td>
<td>Windows 95 or Windows NT.</td>
</tr>
</tbody>
</table>

**Trade Off and Optimization Analysis**

Further analysis of the system requirement shows that the system requirements may be met by utilizing MS Access for both the back-end and front-end of the COTS Management System. This has a positive cost impact by reducing the need to purchase Oracle and a network version of MS Access while yielding acceptable results.
As shown in Figure 4-10, the initial design will use MS Access for centralized Tables, Queries, Forms, and Reports. Only the actual MS Access software application software will run on the client, to increase performance for the application. Tables (Data) will be separated from the Queries, Forms, Reports, and Macros (interface objects) to increase maintainability. Tables can be linked to the interface using MS Access. Also, MS Access will be used to provide database security via a workgroup security file. File and Directory security will be implemented on the server via a New Technology File System [NTFS] partition that is created using Windows NT.
Assuming the concept above is utilized for Detail Design and Development, the system performance will be verified during System Test and Evaluation. If this approach fails to yield adequate performance, then the original concept of using Oracle on the server and connecting (ODBC) via MS Access will be investigated further. In any event, the
approach outlined here will result in either a working system or an excellent prototype that can be easily modified to increase performance.

The three sites will have a majority of users browsing data only, as spelled out in the system requirements. By using database replication, data can be routinely distributed to the sites for local use. This eliminates the need for these view only users to enter the Wide Area Network [WAN]. The result is increased database performance since they access via their Local Area Network [LAN] only. All updates to the database will be performed at the central site (currently Reston). Hence, database locking will only need to be implemented on the database at this site. This approach is depicted in Figure 4-11 below. With trade off and optimization complete, the next life cycle phase, Detail Design and Development, is turned to.
Figure 4-11 Client-Server Relationship.
CHAPTER 5: DETAIL DESIGN AND DEVELOPMENT

DETAIL DESIGN AND DEVELOPMENT OVERVIEW

The detail design phase is the next step after the preliminary system design work of Chapter 4. The detail design phase in this project has the primary goal of outlining the specifics necessary to implement a design that satisfies the requirements previously identified for the COTS Management System. This is where the use of current technology is utilized to actually come up with a design that can be used to build the system. The major process used in the detail design phase is shown in Figure 5-1. The process of detail design is iterative, since each step toward the final design yields more information to feedback into the design.
INITIAL SYSTEM DESIGN

The system will be initially implemented as outlined in the Trade Off and Optimization Analysis section of the Preliminary System Design chapter. A prototype will be developed as a starting point to validate the system and requirements amongst a set of potential system users. As stated by Sommerville in Software Engineering (p.104), "One way to counter the difficulty which users have in formulating and understanding static specifications is to develop a system prototype and to allow users to experiment with it. The system requirements are validated because users discover requirements errors or omissions early in the software process."

Figure 5-1. Detail Design and Development Phase (Overview). (Adapted from Blanchard and Fabrycky, p.83).
MS Access will be utilized as the client application software, and it will be used to develop files for the database tables (data), queries, reports, forms, and macros (interface), and security. All shared database objects will be located on the central database server, which will be a Pentium PC running MS Windows NT at the Reston site. Additional file level security will be provided by Windows NT. This overall concept is depicted by Figure 5-2 below.
Figure 5-2 Client-Server Configuration.

The concept of data replication to servers at the St. Louis and Bethesda sites is also depicted, and clients will access sites locally for browsing while all updates will go to the central server at Reston. There will be an Operational Environment as well as a
Development and Test Environment on the Reston server. Only development and
maintenance personnel will have access to the Development and Test environment, which
is where all upgrades and fixes to the COTS Management system will be made and tested.
Figure 5-2 also depicts the backup of all COTS Management System objects. This backup
will be made to a floppy disk or tape device depending on the size of objects.

The information in Appendix A defines the minimum system requirements for any
client or server running the COTS Management System based on MS Access. The
specific Operating System used will further define requirements, especially for servers or
clients using Windows NT. These clients should be Pentium PCs with at least 32 MB of
RAM (although NT will run on a lesser platform) to ensure sufficient performance.

The database will be designed utilizing the concepts for the objects as previously
defined using management systems engineering methodology (refer to Figure 3-10). The
primary system table is the COTS Request (CSCR), which captures all pertinent
information regarding the particular request for COTS change. When requests for new
COTS products are made, the Alternatives table captures data about evaluated COTS
products. The tables Action Assignment and Action Response are used to assign tasks to
organizations and maintain status. The tables Impacts & Recommendation, Test Summary
Report, and Memo are used to record specific information and events for the CSCR. The
Memo table adds flexibility by incorporating an OLE column to link pertinent documents
to the database.
Figure 5-3 shows a high-level entity relationship diagram for the COTS Management System. Six “lookup tables” are incorporated to ensure data entry consistency for queries and maintenance. Lookup tables are useful for maintenance because changes to defined items, like system platform names or organizations, can be made in one place. Since all tables that incorporate these values do so by reference, the change automatically propagates through the system. Lookup tables also restrict the user to selecting inputs from a finite set of possible values, which will yield consistent query results. Allowing text fields for these items would mean that querying on them would be extremely difficult because of the many permutations of possible values that might be input by different users of the system.
Figure 5-3 COTS Management System - Database Design (Overview)
The following tables give the basic column information for the system Data Module prototype and were derived using its database documentor tool. They are listed in alphabetical order as Table 5-1 through Table 5-13.

Table 5-1 Action Assignments Table Columns

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action_Assignment_ID</td>
<td>Number (Long)</td>
<td>4</td>
</tr>
<tr>
<td>CSCR_Ref</td>
<td>Number (Long)</td>
<td>4</td>
</tr>
<tr>
<td>Action_Assignment</td>
<td>Number (Integer)</td>
<td>2</td>
</tr>
<tr>
<td>Action_Due_Date</td>
<td>Date/Time</td>
<td>8</td>
</tr>
<tr>
<td>Responsible_Org_ID</td>
<td>Number (Integer)</td>
<td>2</td>
</tr>
<tr>
<td>Assign_Last_Name</td>
<td>Text</td>
<td>50</td>
</tr>
<tr>
<td>Assign_First_Name</td>
<td>Text</td>
<td>50</td>
</tr>
<tr>
<td>Assign_Phone</td>
<td>Text</td>
<td>50</td>
</tr>
<tr>
<td>Action_Comments</td>
<td>Memo</td>
<td>-</td>
</tr>
<tr>
<td>Action_Closed</td>
<td>Yes/No</td>
<td>1</td>
</tr>
<tr>
<td>Action_Close_Date</td>
<td>Date/Time</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 5-2 Action Response Table Columns

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action_Response_ID</td>
<td>Number (Long)</td>
<td>4</td>
</tr>
<tr>
<td>Action_Assignment_Ref</td>
<td>Number (Long)</td>
<td>4</td>
</tr>
<tr>
<td>Action_Response</td>
<td>Memo</td>
<td>-</td>
</tr>
<tr>
<td>Response_Date</td>
<td>Date/Time</td>
<td>8</td>
</tr>
<tr>
<td>Response_Last_Name</td>
<td>Text</td>
<td>50</td>
</tr>
<tr>
<td>Response_First_Name</td>
<td>Text</td>
<td>50</td>
</tr>
<tr>
<td>Response_Phone</td>
<td>Text</td>
<td>50</td>
</tr>
</tbody>
</table>
### Table 5.3 Actions Lookup Table Columns

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action_ID</td>
<td>Number (Integer)</td>
<td>2</td>
</tr>
<tr>
<td>Action</td>
<td>Text</td>
<td>50</td>
</tr>
<tr>
<td>Description</td>
<td>Text</td>
<td>255</td>
</tr>
</tbody>
</table>

### Table 5.4 Alternatives Table Columns

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alternatives_ID</td>
<td>Number (Long)</td>
<td>4</td>
</tr>
<tr>
<td>CSCR_Ref</td>
<td>Number (Long)</td>
<td>4</td>
</tr>
<tr>
<td>Product_Name</td>
<td>Text</td>
<td>50</td>
</tr>
<tr>
<td>Product_Version</td>
<td>Text</td>
<td>50</td>
</tr>
<tr>
<td>Viable_Candidate</td>
<td>Yes/No</td>
<td>1</td>
</tr>
<tr>
<td>Demo_Date</td>
<td>Date/Time</td>
<td>8</td>
</tr>
<tr>
<td>Comments</td>
<td>Memo</td>
<td>-</td>
</tr>
<tr>
<td>Vendor</td>
<td>Text</td>
<td>100</td>
</tr>
<tr>
<td>POC_Last_Name</td>
<td>Text</td>
<td>50</td>
</tr>
<tr>
<td>POC_First_Name</td>
<td>Text</td>
<td>50</td>
</tr>
<tr>
<td>POC_Phone</td>
<td>Text</td>
<td>15</td>
</tr>
<tr>
<td>POC_FAX</td>
<td>Text</td>
<td>15</td>
</tr>
<tr>
<td>POC_Email_Address</td>
<td>Text</td>
<td>50</td>
</tr>
<tr>
<td>Vendor_Street</td>
<td>Text</td>
<td>50</td>
</tr>
<tr>
<td>Vendor_Street2</td>
<td>Text</td>
<td>50</td>
</tr>
<tr>
<td>Vendor_City</td>
<td>Text</td>
<td>50</td>
</tr>
<tr>
<td>Vendor_State</td>
<td>Text</td>
<td>2</td>
</tr>
<tr>
<td>Vendor_Zip</td>
<td>Text</td>
<td>50</td>
</tr>
<tr>
<td>Other_Vendor_Info</td>
<td>Memo</td>
<td>-</td>
</tr>
</tbody>
</table>
### Table 5.5 COTS Request (CSKR) Table Columns

#### Table: COTS Request

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control_Number</td>
<td>Number (Long)</td>
<td>4</td>
</tr>
<tr>
<td>Date_Created</td>
<td>Date/Time</td>
<td>8</td>
</tr>
<tr>
<td>New_COTS</td>
<td>Yes/No</td>
<td>1</td>
</tr>
<tr>
<td>Product_Requested</td>
<td>Text</td>
<td>50</td>
</tr>
<tr>
<td>Version_Requested</td>
<td>Text</td>
<td>25</td>
</tr>
<tr>
<td>Product_Current</td>
<td>Text</td>
<td>50</td>
</tr>
<tr>
<td>Version_Current</td>
<td>Text</td>
<td>25</td>
</tr>
<tr>
<td>Vendor_Name</td>
<td>Text</td>
<td>50</td>
</tr>
<tr>
<td>Comments</td>
<td>Memo</td>
<td>-</td>
</tr>
<tr>
<td>Date_Needed</td>
<td>Date/Time</td>
<td>8</td>
</tr>
<tr>
<td>Priority</td>
<td>Text</td>
<td>1</td>
</tr>
<tr>
<td>Req_Plat_ID</td>
<td>Number (Integer)</td>
<td>2</td>
</tr>
<tr>
<td>SubSystem_ID</td>
<td>Number (Integer)</td>
<td>2</td>
</tr>
<tr>
<td>Site_ID</td>
<td>Number (Integer)</td>
<td>2</td>
</tr>
<tr>
<td>Req_Last_Name</td>
<td>Text</td>
<td>25</td>
</tr>
<tr>
<td>Req_First_Name</td>
<td>Text</td>
<td>25</td>
</tr>
<tr>
<td>Req_Org_ID</td>
<td>Number (Integer)</td>
<td>2</td>
</tr>
<tr>
<td>Phone</td>
<td>Text</td>
<td>20</td>
</tr>
<tr>
<td>Closed</td>
<td>Yes/No</td>
<td>1</td>
</tr>
<tr>
<td>Date_Closed</td>
<td>Date/Time</td>
<td>8</td>
</tr>
<tr>
<td>Status_ID</td>
<td>Number (Integer)</td>
<td>2</td>
</tr>
<tr>
<td>DR_Number</td>
<td>Number (Long)</td>
<td>4</td>
</tr>
<tr>
<td>Installed</td>
<td>Yes/No</td>
<td>1</td>
</tr>
<tr>
<td>Date_Installed</td>
<td>Date/Time</td>
<td>8</td>
</tr>
<tr>
<td>RFC_Required</td>
<td>Yes/No</td>
<td>1</td>
</tr>
<tr>
<td>RFC_Number</td>
<td>Text</td>
<td>20</td>
</tr>
<tr>
<td>Vendor_ID</td>
<td>Number (Long)</td>
<td>4</td>
</tr>
<tr>
<td>Product_Serial_Number</td>
<td>Text</td>
<td>20</td>
</tr>
</tbody>
</table>
### Table 5-6 Impacts & Recommendation Table Columns

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Impact&amp;Reco_ID</td>
<td>Number (Long)</td>
<td>4</td>
</tr>
<tr>
<td>Imp_Control_Number</td>
<td>Number (Long)</td>
<td>4</td>
</tr>
<tr>
<td>Impact</td>
<td>Memo</td>
<td>-</td>
</tr>
<tr>
<td>Risk</td>
<td>Memo</td>
<td>-</td>
</tr>
<tr>
<td>Recommendation</td>
<td>Memo</td>
<td>-</td>
</tr>
<tr>
<td>Comments</td>
<td>Memo</td>
<td>-</td>
</tr>
<tr>
<td>Rev_Org_ID</td>
<td>Number (Integer)</td>
<td>2</td>
</tr>
<tr>
<td>Rev_Last_Name</td>
<td>Text</td>
<td>25</td>
</tr>
<tr>
<td>Rev_First_Name</td>
<td>Text</td>
<td>25</td>
</tr>
<tr>
<td>Rev_Phone</td>
<td>Text</td>
<td>20</td>
</tr>
<tr>
<td>Rev_Date</td>
<td>Date/Time</td>
<td>8</td>
</tr>
</tbody>
</table>

### Table 5-7 Memo Table Columns

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memo_ID</td>
<td>Number (Long)</td>
<td>4</td>
</tr>
<tr>
<td>Memo_Control_Number</td>
<td>Number (Long)</td>
<td>4</td>
</tr>
<tr>
<td>Memo</td>
<td>Memo</td>
<td>-</td>
</tr>
<tr>
<td>CMS_Approval_Memo</td>
<td>Yes/No</td>
<td>1</td>
</tr>
<tr>
<td>Date_Created</td>
<td>Date/Time</td>
<td>8</td>
</tr>
<tr>
<td>Object</td>
<td>OLE Object</td>
<td>-</td>
</tr>
<tr>
<td>Object_Name</td>
<td>Text</td>
<td>50</td>
</tr>
</tbody>
</table>
### Table 5.8 Organizations Lookup Table Columns

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Org_ID</td>
<td>Number (Integer)</td>
<td>2</td>
</tr>
<tr>
<td>Org_Name</td>
<td>Text</td>
<td>50</td>
</tr>
<tr>
<td>Description</td>
<td>Memo</td>
<td>-</td>
</tr>
</tbody>
</table>

### Table 5.9 Platforms Lookup Table Columns

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Platform_ID</td>
<td>Number (Integer)</td>
<td>2</td>
</tr>
<tr>
<td>Platform_Name</td>
<td>Text</td>
<td>25</td>
</tr>
<tr>
<td>Description</td>
<td>Text</td>
<td>50</td>
</tr>
</tbody>
</table>

### Table 5.10 Sites Lookup Table Columns

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Site_ID</td>
<td>Number (Integer)</td>
<td>2</td>
</tr>
<tr>
<td>Site_Name</td>
<td>Text</td>
<td>20</td>
</tr>
<tr>
<td>Description</td>
<td>Text</td>
<td>50</td>
</tr>
</tbody>
</table>
### Table 5-11. Status Lookup Table Columns.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status_ID</td>
<td>Number (Integer)</td>
<td>2</td>
</tr>
<tr>
<td>Status_Name</td>
<td>Text</td>
<td>50</td>
</tr>
<tr>
<td>Description</td>
<td>Memo</td>
<td>-</td>
</tr>
</tbody>
</table>

### Table 5-12. Sub-System Lookup Table Columns.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>SubSystem_ID</td>
<td>Number (Integer)</td>
<td>2</td>
</tr>
<tr>
<td>SubSystem_Name</td>
<td>Text</td>
<td>25</td>
</tr>
<tr>
<td>Description</td>
<td>Text</td>
<td>50</td>
</tr>
</tbody>
</table>

### Table 5-13. Test Report Columns.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test_Rpt_ID</td>
<td>Number (Long)</td>
<td>4</td>
</tr>
<tr>
<td>Test_Control_Number</td>
<td>Number (Long)</td>
<td>4</td>
</tr>
<tr>
<td>Test_Description</td>
<td>Memo</td>
<td>-</td>
</tr>
<tr>
<td>Results</td>
<td>Memo</td>
<td>-</td>
</tr>
<tr>
<td>Test_Recommendation</td>
<td>Memo</td>
<td>-</td>
</tr>
<tr>
<td>Tester_Last_Name</td>
<td>Text</td>
<td>25</td>
</tr>
<tr>
<td>Tester_First_Name</td>
<td>Text</td>
<td>25</td>
</tr>
<tr>
<td>Phone</td>
<td>Text</td>
<td>20</td>
</tr>
<tr>
<td>Test_Date</td>
<td>Date/Time</td>
<td>8</td>
</tr>
</tbody>
</table>

It should be noted that all Date formats are “mm/dd/yyyy” such that century rollover problems will not be an issue.
Next, the data for the lookup tables must be developed consistent with the concept defined by management systems engineering. Of particular interest are the “Actions” and “Status” lookup tables, because they are used to assign the tasks and measure how a particular activity is progressing. An initially populated “Actions” table is shown in Table 5-14 below.

**Table 5-14 Populated Actions Lookup Table.**

<table>
<thead>
<tr>
<th>Action_ID</th>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Define COTS Requirements</td>
<td>Begin Research Actions</td>
</tr>
<tr>
<td>2</td>
<td>Research candidate solutions</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Evaluate &amp; document alternatives</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Select best COTS product solution</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Coordinate product sales demo</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Obtain demo copy of product</td>
<td>Begin Logistics Actions</td>
</tr>
<tr>
<td>21</td>
<td>Obtain cost information</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Obtain COTS product upgrade</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Procure new COTS product</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>Complete Impacts and Recommendations for CSCR</td>
<td>Begin Evaluation Actions</td>
</tr>
<tr>
<td>41</td>
<td>Demo product in DPS environment</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>Produce CMS COTS Recommendation</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>Install &amp; configure COTS in IDME</td>
<td>Begin Test Actions</td>
</tr>
<tr>
<td>61</td>
<td>Perform COTS product testing</td>
<td></td>
</tr>
<tr>
<td>62</td>
<td>Complete Test Report for CSCR</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>Create Dr for CSCR [ACMS]</td>
<td>Begin ACMS Actions</td>
</tr>
<tr>
<td>81</td>
<td>Enter TI for COTS DR [ACMS]</td>
<td></td>
</tr>
<tr>
<td>82</td>
<td>Evaluate ERB Assessment of COTS DR [ACMS]</td>
<td></td>
</tr>
<tr>
<td>83</td>
<td>Create RFC for CSCR [ACMS]</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>Install &amp; configure COTS in DSE</td>
<td>Begin CM Actions</td>
</tr>
<tr>
<td>101</td>
<td>Submit CTO package</td>
<td></td>
</tr>
<tr>
<td>102</td>
<td>Verify new COTS functionality in DSE</td>
<td></td>
</tr>
</tbody>
</table>

These values were derived from the management systems engineering analysis performed in Chapter 3. These values are used to assign consistent tasking to groups involved in the COTS change process.
Next, an initially populated “Status” table is shown in Table 5-15. This table provided the inputs necessary to measure how many CSCRs are in each status. This can help indicate where problem areas are occurring within the process of changing COTS products. It should be noted that all Status_IDS > 100 indicate a CSCR for a product that ultimately did not get installed.

<table>
<thead>
<tr>
<th>Status_ID</th>
<th>Status_Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>On Hold</td>
<td>Suspended animation state</td>
</tr>
<tr>
<td>2</td>
<td>Pending RFC/ECP</td>
<td>Outside scope of contract; requires RFC to implement</td>
</tr>
<tr>
<td>10</td>
<td>Evaluation Pending</td>
<td>Evaluation not started</td>
</tr>
<tr>
<td>11</td>
<td>Evaluation in Progress</td>
<td>Evaluation on-going</td>
</tr>
<tr>
<td>20</td>
<td>Awaiting Testing</td>
<td>After CMS Mgt approval; prior to test</td>
</tr>
<tr>
<td>21</td>
<td>Being Tested</td>
<td>System, application, and integration level testing</td>
</tr>
<tr>
<td>40</td>
<td>Pending ERB Assessment</td>
<td>Post-test, Pre-install</td>
</tr>
<tr>
<td>50</td>
<td>Awaiting DSE Installation</td>
<td>Install coordinated with CMO</td>
</tr>
<tr>
<td>100</td>
<td>Closed-Installed to DSE</td>
<td></td>
</tr>
<tr>
<td>101</td>
<td>Closed-Failed ERB Approval</td>
<td></td>
</tr>
<tr>
<td>102</td>
<td>Closed-Failed Testing</td>
<td></td>
</tr>
<tr>
<td>103</td>
<td>Closed-Failed CMS Approval</td>
<td></td>
</tr>
<tr>
<td>104</td>
<td>Closed-Other</td>
<td>Closed for other reasons [specified in I&amp;R]</td>
</tr>
</tbody>
</table>

The information above outlines only some of the details for the COTS Management System prototype. The Interface Module and Security Module are built on top of the Data Module to produce a working prototype that can be installed on the network database server platform running the Windows NT operating system. The Interface Module contains a myriad of queries, forms, reports, and macros utilized to assist the user in navigating through the database system. These objects are built using MS Access and they are based on the defined table structure. The Security Module
provides the access permissions for the users and groups of the system. This module is developed entirely with MS Access’ security tools.

With a working prototype of the COTS Management System developed, attention is next turned to System Test and Evaluation of the system.
CHAPTER 6: SYSTEM TEST AND EVALUATION

TEST AND EVALUATION OVERVIEW

Throughout the systems engineering process, test and evaluation of the COTS Management System is required to ensure all pertinent requirements are being adequately addressed. The system must be tested against all previously identified requirements and evaluated to determine any possible design short falls.

The test and evaluation process begins with planning based on the previously determined system requirements, followed by accomplishing (developing the system), which leads into measuring via test and evaluation. If the results are satisfactory, the system is produced. If results are unsatisfactory, a search is made for feasible alternatives to correct deficiencies. Assuming alternatives are available, changes to the system are implemented and a better system is the result. This is an iterative process, and until test and evaluation yields acceptable results, the system is not produced. The worst case scenario would be if the system fails to achieve acceptable results and there are no feasible alternatives available. In this case the system can not be produced unless requirements can be relaxed in the area where the deficiencies are present. This process is shown in Figure 6-1 below.
Figure 6-1 Test and Evaluation Process (Overview). (Taken from Blanchard and Fabrycky, p.101).
Testing and evaluation begins during the conceptual design and continues through the system operational phase. The goal of test and evaluation is to identify design deficiencies in the COTS Management System as early as possible to limit the cost incurred with system changes. Tests of the operational system are the most accurate; however, implementing system changes later in the system engineering life cycle are the most expensive to make.

To measure test results, the TPMs which were defined above in the Conceptual System Design phase are used. If the quantitative requirement was modified during subsequent development, then the adjusted values would be used. To test the COTS Management System, a combination of inspection, test, and demo techniques are utilized. Inspection entails verifying a requirement by physical observations; Test implies that engineers perform predefined procedures to verify the results; and Demo involves running a prolonged period of system usage with actual users of the system. Shown in Table 6-1 is a Requirements Verification Traceability Matrix based on the defined TPMs for the COTS Management System. This matrix provides the framework and planning for test and evaluation of the system.

As shown in Figure 6-1 above, system evaluation (#3) is done by verification methods, which may yield unfavorable results. This leads to a search for feasible alternatives (#4) and implementation of changes (#5) to the system. An example of this would be if testing showed that the TPMs for database performance could not be met with
MS Access. An alternative might be to implement the Oracle RDBMS on the server (back-end).

<table>
<thead>
<tr>
<th>Technical Performance Measure (TPM)</th>
<th>Quantitative Requirement</th>
<th>Inspection</th>
<th>Test</th>
<th>Demo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Life-Cycle Cost</td>
<td>$30,000/year (5 years)</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Database Connection Time</td>
<td>10 seconds (max)</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Concurrent Users (View &amp; Insert Rows)</td>
<td>25 (min)</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Concurrent Users (Edit Existing Rows)</td>
<td>5 (min)</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>New Record Input</td>
<td>3 minutes (max)</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Database Query Response</td>
<td>5 seconds (max)</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSR Summary Report (Create &amp; Print)</td>
<td>20 seconds (max)</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Action Assignment Report (Create &amp; Print)</td>
<td>30 seconds (max)</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Availability (Operational 6am - 7pm)</td>
<td>97.5% (min)</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Dependability</td>
<td>99.9% (min)</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Size of Client Component (PC)</td>
<td>3 x 4.5 feet (max)</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Size of Server Component (PC)</td>
<td>5 x 5 feet (max)</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maintainability (time for full system backup)</td>
<td>20 minutes (max)</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Maintainability (full recovery time)</td>
<td>30 minutes (max)</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Maintainability (MTBF)</td>
<td>2,500 hours (min)</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Maintainability (MTBM)</td>
<td>500 hours (min)</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Human Factors (error rate/year)</td>
<td>2% (max)</td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
CHAPTER 7: CONCLUSIONS AND RECOMMENDATIONS

CONCLUSION

The systems engineering approach was successfully used in this project to design and develop an initial COTS Management System. Management systems engineering was integrated into the early design phases to yield a greater understanding of the requirements for the management tool. The concurrent use of the systems engineering process from Blanchard and Fabrycky and the management system engineering methodology from Kurstedt provided a logical and systematic approach to the design of the COTS Management System. The goals of this project are considered complete; however, system engineering is a continuous process, so the system design can not be considered final until production and implementation are performed.

FURTHER STUDY RECOMMENDATIONS

When the system is produced and operated as intended, further information about the design will become apparent. At that time, in all likelihood, there will be enhancements identified for the COTS Management System which should be studied and implemented as appropriate. It should be emphasized that the systems approach is
iterative; and although an initial design has been defined within this project, the development process is never over until the system is disposed of.

This project focused on the development of the data portion of the COTS Management System. The user interface for this system represents an area where further study is warranted using the practices of human factors engineering. Due to the limited timeframe for this project, this area was not explored in detail.

This project focused on the design of a specific management tool to be used in the process of changing COTS software within a controlled environment (NIMA). It is recommended that further study be performed to determine the practicality of the use of such tools in other environments. Additionally, management systems engineering could be applied to many other aspects of the process for changing COTS products within NIMA. This would undoubtedly yield more useful management tools and deserves further study.
Appendix A

The following information on MS Access system requirements was obtained from the Microsoft Internet World Wide Web Page (www.microsoft.com).

To use Microsoft Access, you need:

- Personal computer with a 386DX or higher processor (486 recommended)

- Microsoft Windows 95 operating system or Microsoft Windows NT Workstation operating system version 3.51 or later (will not run on earlier versions of Windows)

- 12 MB of memory for use on Windows 95; 16 MB of memory for use on Windows NT Workstation

- Hard disk space required: 14 MB compact; 32 MB typical; 42 MB custom (maximum)

- One 3.5" high-density disk drive

- VGA or higher-resolution video adapter (SVGA 256-color recommended)

- Microsoft Mouse or compatible pointing device

Options:

- Microsoft Windows 95, Windows NT Workstation 3.51, Windows NT Server 3.51, or Novell® NetWare® network

- Windows-compatible printer
• 2400 or higher baud modem (9600 baud modem recommended)

• Audio board with headphones or speakers

• Data formats supported:

  Directly imports, exports, and links to: dBASE III Plus®, dBASE IV®, dBASE® 5.0, Microsoft Excel 3.0 to 7.0, Microsoft FoxPro®, 2.x, Microsoft SQL Server(TM), * Paradox® 3.0 to 5.0, ASCII text, and any ODBC-compliant databases

  Import and export: Lotus® 1-2-3® and Visual FoxPro(TM) 3.0
References


Vita

Daniel Lee Basil was born in Baltimore, Maryland on July 1, 1965, the second of two sons, to Charles Thomas and Anna Mae Basil. Dan’s father was an accountant for Control Data Corporation and his mother became the manager of a small shoe store. Dan attended public school in Dundalk, Maryland until he graduated first of a class of over four-hundred from Patapsco Senior High School in June of 1983.

Dan entered the United States Naval Academy on July 6, 1983 and began collegiate studies in the Fall. After four years, Dan completed his Bachelor of Science in Systems Engineering and graduated with merit on May 20, 1987. Dan’s senior design project, a collision avoidance system, was one of a select few systems engineering projects to be presented to the Superintendent of the Naval Academy during Commissioning Week.

Dan was commissioned an officer in the United States Marine Corps and trained as a communications officer at Quantico, Virginia. While in the Marine Corps, Dan served as a platoon commander at the 7th Communication Battalion in Okinawa, Japan from April 1988 to April 1989. Additionally, Dan served as a series officer for Third Recruit Training Battalion and as a staff officer for the Western Recruiting Region during his three years at the Marine Corps Recruit Depot in San Diego, California. Dan earned his MBA from the United States International University in San Diego, California in June 1992.
After his military service, Dan went to work for General Electric Aerospace on December 7, 1992 as a System Engineer in Springfield, Virginia. As a government contractor, he was involved with both O&M and R&D activities for complex database-oriented computer systems. After two corporate mergers, he left Lockheed Martin in August 1995 to accept a position with Quality Systems Incorporated. Dan is currently a Lead Systems Engineer working on a government contract for the National Imagery and Mapping Agency [NIMA] in Reston, Virginia.