

AD-A265 469



2

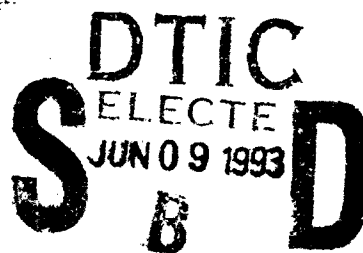
RL-TR-92-345, Vol III (of seven)  
Final Technical Report  
December 1992



# SYSTEM ENGINEERING CONCEPT DEMONSTRATION, Process Model

McDonnell Douglas Corporation

Alberto Ortiz



APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

93 6 075

*Copyright 1992 Software Productivity Solutions, Inc.  
This material may be reproduced by or for the U.S. Government pursuant to the copyright license  
under clause at DFARS 252.227-7013 (October 1988).*

Rome Laboratory  
Air Force Materiel Command  
Griffiss Air Force Base, New York

93-12877



This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RL-TR-92-345, Volume III (of seven) has been reviewed and is approved for publication.

APPROVED:



FRANK S. LAMONICA  
Project Engineer

FOR THE COMMANDER:



JOHN A. GRANIERO  
Chief Scientist  
Command, Control & Communications Directorate

If your address has changed or if you wish to be removed from the Rome Laboratory mailing list, or if the addressee is no longer employed by your organization, please notify RL ( C3CB ) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

# REPORT DOCUMENTATION PAGE

Form Approved  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing the burden, to Washington Headquarters Service, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave Blank)		2. REPORT DATE December 1992		3. REPORT TYPE AND DATES COVERED Final Feb 90 - Jul 92	
4. TITLE AND SUBTITLE SYSTEM ENGINEERING CONCEPT DEMONSTRATION, Process Model				5. FUNDING NUMBERS C - F30602-90-C-0021 DE - 527027 PR - 5581 TA - 18 DT - 54	
6. AUTHOR(S) Alberto Ortiz					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) McDonnell Douglas Corporation Douglas Aircraft Company 3855 Lakewood Blvd Long Beach CA 90846-0001				8. PERFORMING ORGANIZATION REPORT NUMBER N/A	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Rome Laboratory (CGCE) 525 Brooks Road Griffiss AFB NY 13441-4505				10. SPONSORING/MONITORING AGENCY REPORT NUMBER RL-TR-92-345, Vol III (of seven)	
11. SUPPLEMENTARY NOTES Rome Laboratory Project Engineer: Frank S. LaMonica/(315) 330-2054					
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.				12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) This final technical report documents the objectives, activities, and results of Air Force contract F30602-90-C-0021, entitled "System Engineering Concept Demonstration." The effort, which was conducted by Software Productivity Solutions, Inc., with McDonnell Douglas Corporation - Douglas Aircraft Company and MTM Engineering Inc. as subcontractors, demonstrated and documented the concept of an advanced computer-based environment which provides automation for Systems Engineering tasks and activities within the Air Force computer-based systems life cycle. The report consists of seven (7) volumes as follows: I) Effort Summary, II) Systems Engineering Needs, III) Process Model, IV) Interface Standards Studies, V) Technology Assessments, VI) Trade Studies, and VII) Security Study. This Volume (Volume III - Process Model), presents the details and results of the development of a system acquisition and development model that emphasizes System Engineering oriented tasks and activities over the entire system life cycle. The purpose of the model was to 1) better understand the tasks and activities of the system engineering process as imposed by various MIL/DoD/AF standards and regulations and 2) support the determination of completeness and adequacy of the envisioned systems engineering automation's system level requirements.					
14. SUBJECT TERMS System Engineering, System Life Cycle Tools, System Life Cycle Environment				15. NUMBER OF PAGES 128	
				16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED		18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED		19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	
				20. LIMITATION OF ABSTRACT UL	

## Table of Contents

List Of Figures .....	iv
List Of Tables .....	vi
1.0 Introduction.....	1
1.1 Report Organization.....	1
1.2 Task Definition .....	2
1.2.1 Goals.....	3
1.2.1.1 Road Map.....	3
1.2.1.2 Goals Supporting Steps.....	3
1.2.2 Objectives .....	6
1.2.3 Acceptability Criteria And Factors .....	7
1.3 Definitions & Terms .....	8
1.3.1 System Engineering .....	8
1.3.2 Process or Process Model .....	11
1.3.3 Modeling & Simulation .....	12
1.4 Process Modeling .....	13
1.4.1 Representations .....	15
1.4.2 Abstraction .....	17
1.5 Process Model Documents .....	18
2.0 Precedence List Of System Engineering Standards .....	21
2.1 DoDI 5000.2.....	21
2.2 MIL-STD-1521B .....	22
2.3 MIL-STD-973 .....	22
2.4 MIL-STD-499A .....	22
2.5 MIL-STD-499B .....	23
2.6 DoD-STD-2167A.....	23
2.7 AFR 800-14 .....	24
2.8 S&E Instruction 5451.2 .....	24

3.0 SECD Process Model .....	25
3.1 Description Of Results .....	25
3.1.1 Representation Methodology .....	25
3.1.1.1 ANSI Standard Symbology .....	26
3.1.1.2 Consistency Rules .....	28
3.1.1.3 Representation Steps .....	36
3.1.1.4 Organization .....	38
3.1.1.5 Assumptions .....	39
3.2 Description Of Major System Life Cycle Phases And Processes .....	40
3.2.1 Determination Of Mission Needs Phase .....	41
3.2.2 Concept Exploration Phase .....	44
3.2.3 Demonstration And Validation Phase .....	46
3.2.4 Engineering And Manufacturing Development Phase .....	48
3.2.4.1 Hardware, Software And The System Engineering Process .....	50
3.2.5 Production And Deployment Phase .....	51
3.2.6 Operations And Support Phase .....	53
3.2.6.1 Production & Deployment And Operation & Support Phases .....	55
3.3 Abstracted View .....	56
3.4 Discriminating Factors .....	60
4.0 Conclusions .....	63
4.1 System Engineering .....	63
4.2 System Engineering Process .....	64
4.3 Process Model .....	64
4.4 Lessons Learned .....	65
5.0 Future Applications And Directions .....	67
5.1 Reference Model .....	67
5.2 Process Management and Guide .....	67
5.3 Training Tool .....	68
5.4 Process Knowledgeable .....	68
5.5 Metrics And Instrumentation .....	68

References.....73

Appendix A - Literature Survey Of Existing System Engineering  
Processes And Models.....A-1

Appendix B Document Summaries.....B-1

Appendix C Representation Experiments.....C-1

OLD QUALITY INSPECTED 2

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

## List Of Figures

Figure 1.2.1.1-1 Process Model Development Road Map.....	4
Figure 1.4.1-1 System Engineering Process Variation.....	16
Figure 1.5-1 Process Model Documents.....	19
Figure 3.1.1.1-1 Symbology Used to Represent the Process Model .....	26
Figure 3.1.1.1-2 Input/Output Symbol Page View .....	27
Figure 3.1.1.2-1 Input/Output and Process Symbol Page View .....	29
Figure 3.1.1.2-2 Decision Symbol Page View .....	30
Figure 3.1.1.2-3 Review Symbol Page View .....	28
Figure 3.1.1.2-4 Milestone Symbol Page View .....	32
Figure 3.1.1.2-5 Input/Output Grouping Symbol Page View .....	33
Figure 3.1.1.2-6 Process Grouping Symbol Page View .....	34
Figure 3.1.1.2-7 Flow Symbol Page View.....	35
Figure 3.1.1.2-8 Feedback Symbol Page View .....	36
Figure 3.1.1.4-1 Process Model Organization.....	39
Figure 3.2-1 System Acquisition And Development Process Model.....	41
Figure 3.2.1-1 Determination Of Mission Needs Phase.....	43
Figure 3.2.2-1 Concept Exploration Phase.....	45
Figure 3.2.3-1 Demonstration And Validation Phase .....	47
Figure 3.2.4-1 Engineering And Manufacturing Development Phase .....	49
Figure 3.2.4.1-1 Hardware, Software, and System Engineering Process .....	50
Figure 3.2.5-1 Production And Deployment Phase.....	52
Figure 3.2.6-1 Operations And Support Phase .....	54

Figure 3.2.6.1-1 Process Model Production And Deployment and Operations ..	55
Figure 3.3a System Engineering And Development Upper Two Levels .....	57
Figure 3.3b System Engineering And Development Upper Two Levels .....	58
Figure 3.3c System Engineering And Deveioption Upper Two Levels .....	59
Figure 5.5-1 Density Of Metrics Versus System Life Cycle Functional .....	69
Figure 5.5-2 Road Map Components.....	71



## List Of Tables

Table 3.1.1.1-1 Process Model Symbology Description.....	28
--	----

## **1.0 Introduction**

This is the third volume of the Final Technical Report (FTR) for the System Engineering Concept Demonstration (SECD), contract F30602-90-C-0021, sponsored and managed by the Air Force Rome Laboratory (RL). SECD was an exploratory development effort which culminated in *Catalyst*, a fully researched and specified system concept which provides an automated environment of integrated, state-of-the-art software tools and methods. This document reports the results of SECD Process Model Task.

The SECD Process Model is a system acquisition and development model that emphasizes System Engineering activities over the entire system lifecycle. The Process Model is a graphical representation of the System Engineering lifecycle activities, agents, flows, feedbacks, and work products. This interactive Process Model provides a multi-dimensional view of government acquisition and contractor development activities. An integral part of the SECD program, the Process Model aided in developing the system's requirements which are documented in the System/Segment Specification (SSS). The model also demonstrated coverage and completeness of the System Engineering process.

By emphasizing System Engineering activities, the Process Model allowed us to represent and customize these activities within their natural domain. The Process Model includes a list of standards, in order of precedence, to provide validity and traceability to commonly used and approved sources. For the sake of completeness, the processes captured by the model are based on formal and informal activities not previously captured or formalized.

McDonnell Douglas Corporation-Douglas Aircraft Company's (MDC-DAC) primary role and task in the SECD program was to provide SPS and Rome Laboratory with insight and advice on System Engineering processes, policies, and practices. Another task was to develop the system engineering lifecycle Process Model. Our strong background in this area helped ensure a strong system perspective in the development of the *Catalyst* environment. As we move into the 21st century, MDC DAC is committed to improve the quality, cost and performance of our systems. Our interest in *Catalyst* resides in our belief that System Engineering and the automation of the System Engineering process is the key to a competitive, high quality, and better performance product line.

### **1.1 Report Organization**

This report complies with the requirements outlined in the Statement of Work (SOW) of Subcontract No. 1990-J5012-1 between Software Productivity Solutions,

Inc. (SPS) and McDonnell Douglas Corporation-Douglas Aircraft Company (MDC-DAC).

This report contains five sections as follows:

- Introduction
- Precedence list of standards
- SECD process Model
- Conclusions
- Future plans-applications and directions.

Each section contains detailed figures and descriptions which explain the development of the Process Model and the subsequent results. This report also contains three appendixes. Appendix A is a literature survey of existing processes and models. Appendix B provides document summaries. Appendix C displays the Process Model.

## **1.2 Task Definition**

A joint effort between SPS and MDC-DAC, the Process Model Task was defined by studying and researching RL's SECD Statement of Work (SOW), specifically paragraphs 4.1.4.1 and 4.1.3.1 which state, respectively,

"Examine Air Force and Department of Defense (DoD) Regulations, DoD and MIL-Standards, and pamphlets which are typically used during the development of computer-based systems (e.g., AFR 800-14 [and all regulations and standards referenced therein], DOD-STD-2167A, DOD-2168, MIL-STD-483A, MIL-STD-490A, AFSCP 800-14, AFSCP 800-43, AFSC / AFLCP 800-45, etc.; Editions in effect at RFP release). Within the context of these Regulations, standards, and pamphlets, identify the following: 1) engineering, management, and development oriented tasks and activities, 2) personnel roles that are typically responsible for the tasks (e.g., government acquisition manager, project manager, system analyst, programmer, IV&V agent, etc.), and 3) tasks that are conducive to automated assistance by a System Engineering environment and its associated toolset." and 4) "Support for engineering, management, and development activities of the various system lifecycle phases, including concept exploration, demonstration and validation, full-scale development, production, deployment, and post deployment support."

Interpreting these paragraphs provided the Process Model's basic core requirements. Since there were no other established precedents to guide us, our approach to reach the established goals, objectives, assumptions, and acceptability criteria evolved throughout the program. Some fundamental assumptions about the System Engineering process and discipline are embedded within the goals, objectives, and acceptability criteria. These topics are discussed in the following sections.

### **1.2.1 Goals**

The goals of the process modeling activity were varied and broad and were achieved by developing a road map and goal supporting steps. The road map illustrated the goal supporting steps as a function of time and work products. The goal supporting steps outlined the major modeling activities conducted during the SECD program. The following are a list of the Process Model goals:

- Validate *Catalyst* requirements to demonstrate coverage of the System Engineering activities by analysis.
- Prepare an example of a System Engineering process model for use as an environment training tool.
- Identify the default System Engineering processes for initialization of *Catalyst*.
- Provide SPS and Rome Laboratory general insight into System Engineering activities, interfaces, work products, techniques.
- Ensure a high utilitarian value and usability of *Catalyst* as an end product.

#### **1.2.1.1 Road Map**

The road map provided a concise timeline of the goal supporting steps and work products. It graphically illustrated the path which was followed to develop the Process Model. Other work products represented in the road map were derived from the goal supporting step results. Figure 1.2.1.1-1 shows the Process Model Development Road Map.

#### **1.2.1.2 Goals Supporting Steps**

The goal supporting steps for the Process Model Task outlined the major activities followed to develop the process model. The following are the goal supporting steps:

- **Technical Library Searches** - This step obtained the necessary information and references for the work product. The task was carried out by performing global library searches, under various indexes, that would allow us to identify System Engineering processes and activities

from the McDonnell Douglas Corporation (MDC) programs. These activities and processes provided suitable candidates for the Process Model.

A research database program was developed to document the most relevant documents. This database was a collection of numerous System Engineering documents.

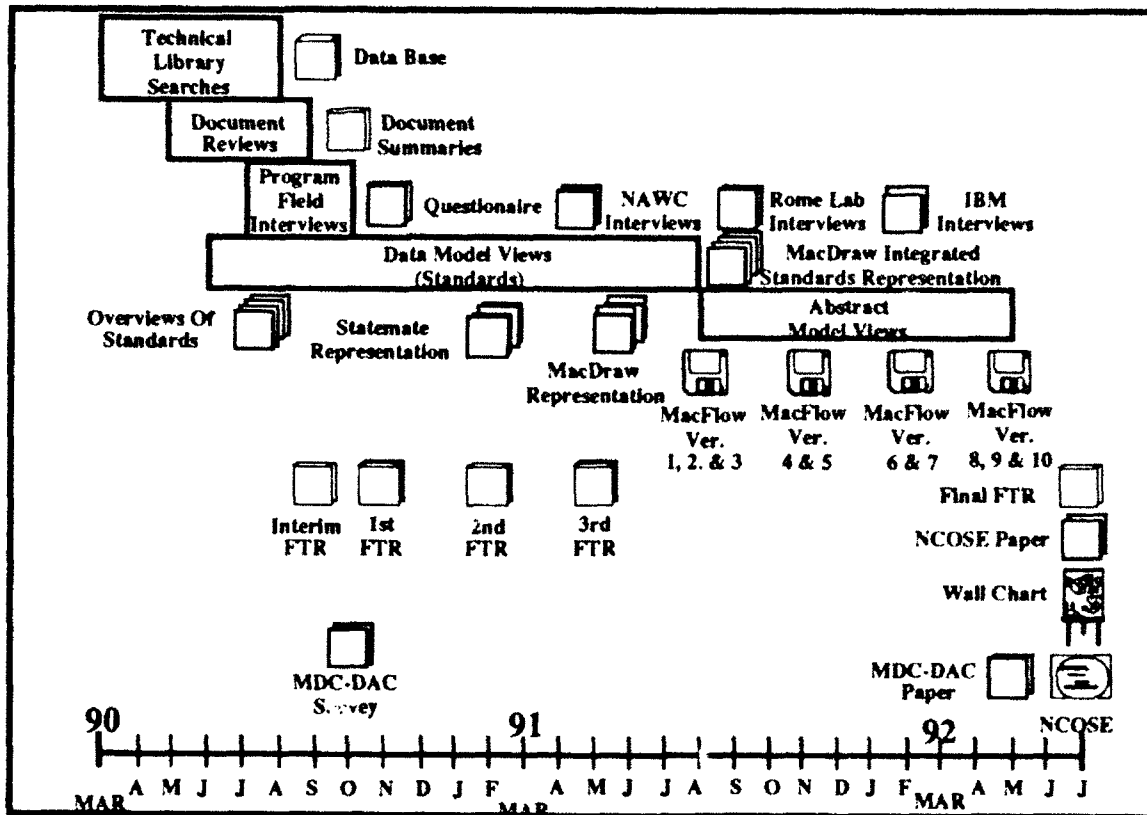


Figure 1.2.1.1-1 Process Model Development Road Map

- **Documentation Reviews** - This step reviewed the work products selected in the library searches and summarized the most relevant ones. The summaries provided supporting references for the Process Model activities and processes.

The documentation reviews provided three summaries:

Kapureh, Stephen J. *"A Systems Engineering Methodology For The Advanced Tactical Aircraft (ATA)."* Naval Post Graduate School, Monterey, California.

*"System Engineering Group Instruction 5451.2 from the Department Of The Naval Air System Command."* Naval Air System Command Headquarters, Washington, D. C. 20361.

Kellner, Mark. *"Software Process Modeling: Value and Experience."* Software Engineering Institute, Carnegie Mellon University Technical Report, pg. 23-54.

- **Field Interviews** - This step conducted field interviews with practicing systems engineers to ascertain System Engineering needs. This task was beneficial because it identified activities not found in the technical library searches or in the documentation review steps.

The field interviews were conducted with practicing systems engineers and employed the following objectives:

1. Understand the areas of high priority attention for systems engineering automation.
2. Understand the areas and degrees of variability in the systems engineering processes.

A total of 15 systems engineers were interviewed in 3 organizations. The organizations represented an industry cross-section:

- New system development and lifecycle support activities
- Government and industry
- Acquisition and in-house activities
- Small, medium and large systems

This step produced five work products:

1. A questionnaire, used during interviews
2. Naval Air Warfare Center (NAWC) Aircraft Division Warminster interviews
3. Rome Laboratory interviews
4. IBM Owego interviews
5. MDC-DAC survey

Volume 2 of the SECD Final Technical Report, Systems Engineering Needs, presents the field interviews.

- **Data Model Views** - This step integrated activities and processes discovered in the previous steps into a cohesive representation. The thrust of the data model views was to represent the interaction between the various agents and processes in the system lifecycle.

This step produced four work products:

1. Overview of standards
  2. I-Logix Statemate representation
  3. MacDraw representation
  4. Integrated MacDraw representation
- **Abstract Model Views** - This step developed a clear and concise representation of the system lifecycle and its associated System Engineering activities, flows, events, transitions, and interactions. It also produced ten versions of the Process Model. Appendix C displays the final version of the Process Model. (Section 3.0 details the structure and organization of the Process Model.)

### **1.2.2 Objectives**

The objectives of the Process Model task were to identify the System Engineering process during the entire system lifecycle, demonstrate coverage of the System Engineering process by the *Catalyst* environment requirements, and adapt the Process Model representation to the SECD Prototypes and Demonstrations Scenarios.

As part of the objectives, the conceptual requirements defined for the Process Model were as follows:

1. a good understanding of the System Engineering process,
2. completeness of *Catalyst* requirements,
3. establishment of the basis for the Operational Concept Document (OCD),
4. provision of an infrastructure for the prototype scenarios, and
5. incorporation of the results into the program prototypes and demonstrations. These conceptual requirements for the Process Model are based on customer needs and provided the basis for establishing acceptability criteria.

The Process Model was used to map *Catalyst* Computer Software Configuration Items (CSCI's) against activities in the upper three levels of the model. These mappings were documented in the Operational Concept Document (OCD), and

they demonstrated coverage and completeness of the System Engineering process by *Catalyst*. System requirements were documented in the System Segment/Specification (SSS).

The Process Model was also used to develop the prototype demonstration scenarios. As part of the system lifecycle, the prototype demonstrations included several scenarios such as tradeoff, timeline, and requirements flow down.

### **1.2.3 Acceptability Criteria And Factors**

"The acceptability criteria of a system or work product (Process Model) are relative to the utility in relation to a set of customer value standards" [CHE65]. Hence, the customer must assess the value of the system or work product. Typical customer criteria include performance, cost, reliability, time, and maintainability. In the case of the Process Model, the acceptability criteria augmented the defined requirements. The following is a list of the Process Model acceptability criteria's factors and their associated descriptions:

- **Readability** - The process model must be clear, understandable, and easy to follow by non-technical and untrained personnel.
- **Traceability** - The Process Model activities must be traceable and validated by formal/informal standards. Informal activities must be captured and identified.
- **Acceptability** - The Process Model must present activities that practicing system engineers identify and recognize as helpful to practitioners.
- **Uniformity** - The Process Model must present a uniform level of information at the upper three levels of detail.
- **Usability** - The Process Model should be easy to operate and apply.
- **Changeability** - The Process Model must be modifiable to specific organizations, programs, or projects. It must be tailorable to customer needs.
- **Consistency** - The Process Model must use a consistent and well-defined representation methodology and symbology.
- **Interoperability** - The Process Model must represent a clear and cohesive communication channel between the user and implementor (i.e., between user and developer, contractor and the government).

The acceptability criteria stated above were developed in an evolutionary manner through experimentation with various representations, methods, and tools. The acceptability criteria were the result of intense customer interaction and involvement with several representation ideas. These factors are not



traceable to any specific requirements, but they were the guidelines used in the development of the Process Model. Examination of the Process Model shows how each criteria is represented.

### **1.3 Definitions & Terms**

This section establishes a solid foundation for further discussion and analysis of the System Engineering process. Emphasis was placed on interactions among various engineering specialties during system development. Basic System Engineering principles and concepts are also introduced to provide a common understanding.

#### **1.3.1 System Engineering**

System Engineering is a problem solving technique that can be applied to numerous disciplines. It applies technical and management skills during the entire lifecycle of a system and transforms customer needs into a viable and operational system. The Defense Systems Management College states, "*In its simplest terms, system engineering is both a technical process and a management process*" [SEMG90]. One, however, should not conclude that System Engineering is a management skill. In reality, it is a multi-discipline skill with management aspects. There are tremendous technical challenges in System Engineering in addition to the management of cost, schedule, and resources. System Engineering is a team approach to problem solving, and it is consistent with the Total Quality Management System (TQMS) principles in existence today.

An on-going MDC System Engineering study defined System Engineering as follows [SEMC90]:

"A structured systematic process for integrating and applying financial, marketing, engineering, manufacturing, and human resource skills and efforts to:

- a) Transform customer needs into products and services which constitute a viable business
- b) **Organize, conceptualize, develop, produce, plan, deploy, measure and control the technical, operational, and business activities to achieve the best balance between customer and company interest."**

The words in bold accentuate the definition's engineering perspective. Without emphasizing these words, the definition is broad and includes a group of disciplines other than engineering. Clearly, this definition ties System Engineering to the profit aspect of the business.

Science has already divided these two disciplines into the areas of business and engineering. There is a difference between applying business information and developing it. The definition above assigns the system engineer with performance responsibility and with accountability of the system's performance during the entire lifecycle. Business and technical decisions inter-relate to each other and point out the special relationship between the program manager and the system engineer.

Chestnut's [CHE65] definition of 'System Engineering' provided more insight into the role of the system engineer. He presented ten different definitions for System Engineering but adhered to one definition set. A definition set of a term was the integration of many definitions into one. The following was his definition set:

"The Systems Engineering method recognizes each system is an integrated whole even though composed of diverse, specialized structures and subfunctions. It further recognizes that any system has a number of objectives and that the balance between them may differ widely from system to system. The methods seek to optimize the overall system functions according to the weighted objectives and to achieve maximum compatibility of its parts."

Chestnut's definition emphasized the technical aspects of System Engineering. He referred to a tightly integrated system as the compatibility of its parts. He also refers to the optimization of the overall system functions implying performance and effectiveness.

It is clear, however, upon further examination that Chestnut's definition has definite drawbacks. It is ambiguous in the distinction between the program manager and the system engineer. The difference between these two roles is too significant to ignore. Historically, program managers address business concerns and system engineers address technical concerns and provide the technical knowledge to support program decisions.

System Engineering is practiced at all levels in development of complex systems. Hence, large complex systems are composed of layers which employ the System Engineering practice. Although some aspects are not visible in some layers, the principles and concepts are readily visible.

The Defense System Management College (DSMC) provided another definition for the term 'System Engineering' [SEMG90]:

"System Engineering is the management which controls the total system development effort for the purpose of achieving an

optimum balance of all system elements. It is a process which transforms an operational need into a description of system parameters and integrates those parameters to optimize the overall system effectiveness".

DSMC's and Chestnut's definitions emphasized the overall performance and effectiveness of the system. The system performance and effectiveness was defined in terms of optimum balance. The added twist in DSMC's definition was its reference to controlling the total system development effort. Obviously, some management mechanism is needed in order to achieve control. DSMC's definition points out the technical and management aspects of System Engineering.

In the late 70's, the Air Force Systems Command (AFSC) published MIL-STD-499A "Engineering Management", a System Engineering standard that guides a program manager in managing engineering functions and the developing of a system. The existing standard was updated and a draft of MIL-STD-499B was released for review in May 15, 1991. This standard defines the term 'System Engineering' as follows:

"The application of scientific and engineering efforts to: (a) transform an operational need into a description of system performance parameters and a system configuration through the use of an iterative process of definition, synthesis, analysis, design, test, and evaluation; (b) integrate related technical parameters and ensure compatibility of all physical, functional, and program interfaces in a manner that optimizes the total system definition and design; (c) integrate reliability, maintainability, safety, survivability, human and other such factors into the total engineering effort to meet cost, schedule and technical performance objectives" [MIL-STD-499B].

MIL-STD-499A's definition was in concert with DSMC's and Chestnut's definition. However, it goes one step further by recognizing the total engineering effort. The total engineering effort was defined in terms of cost, schedule and technical performance.

At McDonnell Douglas Corporation Douglas Aircraft Company, our Standard Process System (SPS) DAC-PD-201, defined 'System Engineering' as follows:

"A disciplined design approach aimed at achieving producible and supportable designs that meet performance and safety requirements with minimum program risk."

This definition was consistent with previous definitions of 'System Engineering' but it emphasizes safety and program risk assessment. At MDC-DAC, the safety of our products is of great concern because our design responsibility spans the entire lifecycle of the produced system.

### **1.3.2 Process or Process Model**

The System Engineering process must be the enactment of the aforementioned definitions. Therefore, a process is an enactment of a function. Let's seek other perspectives of the term 'Process' or 'Process Model'.

Dr. Leon Osterweil provides the following definition of the term 'Process' [OST91].

"Device for producing a product or getting jobs done. The indirect nature of a process is an instance of a process description. Instances create a product or solve a problem. A process description is created to describe a wide class of instances. Humans create process descriptions to solve classes of problems."

From this definition, we can derive that processes are devices for creating and manipulating products, as well as devices for evolving products. It should be clear from this definition that processes are devices used to create, develop and control products. Osterweil treats processes as objects which encapsulate all their relevant information. Processes are instances of a whole, and the whole is a description of the sum of processes.

Using Webster's Dictionary, we combined a definitions of 'Process' and 'Model' to form a definition of a Process Model :

"... the specific embodiment of an architectural process framework within which planned or existing objects representing organizational, functional and behavioral activities are implemented."

This definition emphasizes the multi-dimensional nature of the process and its relation to a larger framework. It also implies that a process needs to be part of a whole. These definitions will be used for the term 'Process'.

This discussion would not be complete without mentioning Watts Humphrey's maturity levels of processes. He applied basic principles for statistical control to process improvement and established a definition for process. "*A process is said to be stable or under statistical control if its future performance is predictable within established statistical limits*" [HUM82]. Humphrey defined process maturity levels as follows [HUM89]:

1. Initial - Until the process is under statistical control, orderly progress in process improvement is not possible. While there are many degrees of statistical control, the first step was to achieve rudimentary predictability of schedule and costs.
2. Repeatable - The organization has achieved a stable process with a repeatable level of statistical control by initiating rigorous project management of commitments, costs, schedule, and changes.
3. Defined - The organization has defined the process as a basis for consistent implementation and better understanding. At this point advanced technology can be usefully introduced.
4. Managed - The organization has initiated comprehensive process measurements and analysis with significant quality improvements.
5. Optimizing - The organization now has a foundation for continuing improvement and optimization of the process

Humphrey's process maturity levels pointed out that processes are owned by organizations and they are implemented by teams. Therefore, a process can only be identified by teams. Humphrey appears to have missed a level zero. A level zero is when you have an Ad Hoc process which has not been identified. Before a statistical control can be achieved, one must have something to control. Most organizations have not reached level zero, and therefore, cannot start level one.

### **1.3.3 Modeling and Simulation**

The definition of terms 'Modeling' and 'Simulation' were essential for the discussion of the System Engineering process. Various quotes were extracted from Chestnut to define the terms 'modeling' and 'simulation' as they were in System Engineering. These extracts provided an insight to the utilization and application of models and simulation in System Engineering.

"A model is a qualitative or quantitative representation of a process or endeavor that shows the effects of those factors which are significant for the purposes being considered. A model may be pictorial, descriptive, qualitative, or generally approximate in nature; or it may be mathematical and quantitative in nature and reasonably precise. It is important that effective means for modeling be understood such as analog, stochastic, procedural, scheduling, flow chart, schematic, and block diagrams."

"Models are used essentially for evaluation and prediction purposes as well as for the analysis and study of the different parts

of the system so that the systems engineer or designer may arrive at sound engineering decisions regarding the system design."

*"As is used in connection with systems engineering, a model is a qualitative or quantitative representation of a process or endeavor that shows the effects of those factors which are significant for the purposes being considered. Modeling is the process of making a model. Although the model may not represent the actual phenomenon in all respects, it does describe the essential inputs, outputs, and internal characteristics, as well as provide an indication of environmental conditions similar to those of actual equipment."*

*"Simulation is the use of models and/or the actual conditions of either the thing being modeled or the environment in which it operates, with the models or conditions in physical, mathematical, or some other form. The purpose of simulation is to explore the various results which might be obtained from the real system by subjecting the model to representative environments which are equivalent to, or in some way representative of, the situations it is desired to understand or investigate. Simulation may involve system hardware and the actual physical environment, or it may involve mathematical models subjected to mathematical forcing or disturbance functions representative of the systems conditions to be studied [CHE65]."*

#### **1.4 Process Modeling**

Process modeling is an emerging technique [KEL90], yet the value of System Engineering Process Modeling is widely misunderstood. *Process Modeling can be defined as the abstraction of a set of instances to develop an entire description of a process, which, in turn, produces a model of that process.*

The need to produce larger, more complex, reliable, and maintainable systems in less time with higher quality standards has brought about the emergence of process models and process modeling techniques. The inefficiencies and shortcomings of existing commercial and military systems development methods has harvested the development and implementation of concepts such as standardization, reusability, modularity, concurrence, and automation. The goal to implement these concepts is to increase productivity and competitiveness. Process Models facilitate the implementation of these concepts into the development process and the system itself.

The techniques of modeling processes are applicable to developmental or conceptual models. A conceptual model is representative of a mission profile or its system elements while a process model is representative of the system

developmental process. A mission profile can be considered an operational process, but in this report a process referred to the developmental aspects of a system.

The modeling of the System Engineering process illustrates several of these developmental concepts previously mentioned. Kellner states "*that the quality of a software product is directly determined by the quality of the processes it represents and uses to develop and maintain itself.*" Therefore, we can conclude that the quality and applicability of process models is determined by the processes they contain. These processes in the System Engineering lifecycle provided the means for organizing other processes used to develop and maintain systems. Clearly, these processes play a substantial role determining the quality, responsiveness, cost, and schedule of a system. As a result, improvements to these processes should lead to significant improvement in the quality, cost, and schedule of a system.

Process models are analogous to the simulation of a system. Process Models consist of activities, transitions, and states experienced by the system during its lifecycle. The relationships among those activities conveys information with regards to the control flow and transitional flow. Classically, control design information has been derived from "*state space equations.*" Process models relate similar information about a system, previously considered conceptual at a high level, in a graphical form.

The application of developmental or conceptual process models is imperative. If properly implemented and maintained, the developmental and conceptual process models could possibly facilitate the evolution of a software environment such as *Catalyst*. The models would not only ascertain impacts of future changes, but also serve as a test bed for the design, prototype, development, test and integration of a software environment. Clearly, *Catalyst* would possess other elements such as User Interface, Data Base Management Systems, and Computing Platforms, but its behavior must be representative of the developmental and conceptual models. It must support related activities contained within these models. The knowledge provided by the Process Model is necessary to support the automation of the System Engineering process.

H. Chestnut, in his book "*System Engineering Tools*", states that "*understanding the process of engineering systems should lead to our being able to control it*" [CHE65]. Chestnut published a series of books addressing System Engineering issues ranging from formal definitions to actual detailed application solutions to System Engineering problems.

Corrigan, in "*Why System Engineering*", presented his ideas in the form of a question and answer discussion [COR66]. He presented a clear and concise System Engineering process in accordance with Air Force System Command

Manual (AFSCM) 375-5. Even though older publications, Chestnut's and Corrigan's works stand out for their depth and breadth of addressing System Engineering issues from an engineer's point of view and show that time has not changed the System Engineering problem.

Although Corrigan's and Chestnut's works are fascinating, it is Dr. Leon Osterweil from the University of Colorado Boulder, CO., who is considered the father of software process modeling. In his paper *"Software Processes Are Software Too"*, he develops the concept of software process modeling with multiple views. He introduces the notion that *"humans solve problems by creating process descriptions and then instantiating the processes to solve individual problems, rather than repetitively and directly solving individual instances of the problem"* [OST87].

Osterweil's representation or specification of a problem in instances is natural to our thinking. He attempts to represent the problem in a matter that is logical and understandable by humans. He states *"our specific approach is to suggest that contemporary "programming" techniques and formalisms be used to express software process descriptions."* This statement explicitly presents the opportunity to use today's programming techniques in the development of software process models.

In his conclusion about software processes, he states *"This strongly suggests the importance of devising a process programming language and a software environment capable of compiling and interpreting process programs written in that language. Such an environment would become a vehicle for the organization of tools for facilitating development and maintenance of both the specified process and the process program itself"* [OST87]. Osterweil clearly points out the value of process models and their relationship to environments such as *Catalyst*.

#### **1.4.1 Representations**

Our modeling strategy and approach was to meet the previously stated customer requirements, needs and acceptability criteria by experimentation. These representations, methodologies and tools included IDEF0, IDEF1, Delta Charts, MacDraw, Embedded Computer System Analysis Method (ECSAM), and others.

During our experiments, we encountered one of the most challenging issues in the SECD program, the issue of process variation. The System Engineering process not only varies throughout the system lifecycle, but from organization to organization, within an organization, and from person to person. The challenge of process variation is ascertaining how to develop a representation approach that supports various System Engineering methods, processes and practices. Our final approach was to represent activities and processes in a generic manner so they could be recognized and tailored by practitioners of System Engineering at



any level. This is one of the reasons for a generic Process Model, also known as the SECD Process Model.

Figure 1.4.1-1. illustrates the vertical and horizontal variations of the System Engineering process within the system lifecycle. These variations were the result of conflicting source documents and standards as well as differing organizational practices and roles, types of systems, and personal preferences.

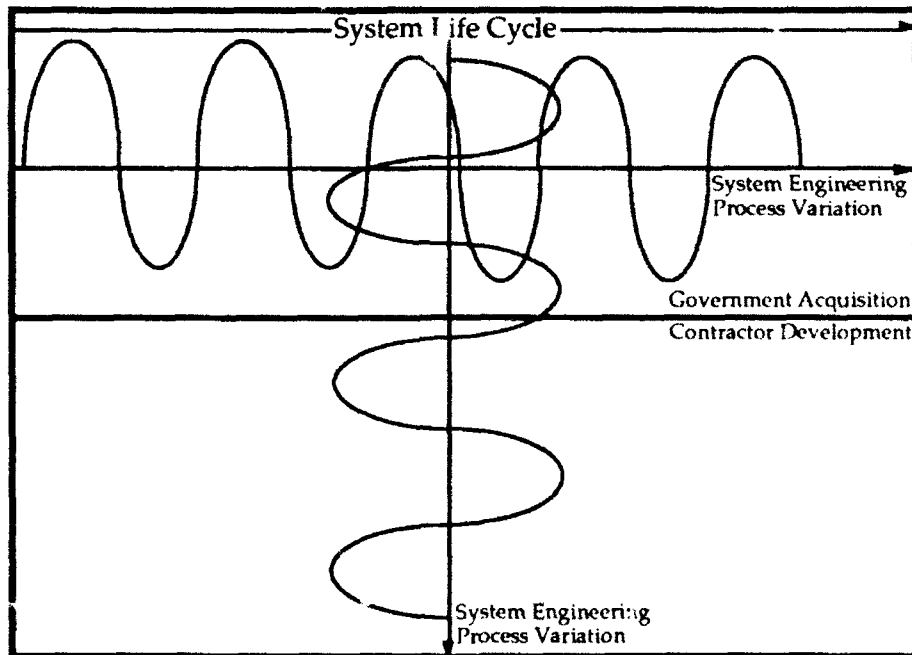


Figure 1. 4.1 -1 System Engineering Process Variation

During our examination of various source documents and standards, we noticed conflicting directives, even within the same standard. Therefore, there was no definitive process to follow other than our best judgement, practice, and experience. To resolve this conflict, we developed a list of standards, in order of precedence. This list of standards allows one to prioritize directives, standardize name labels, and validate activities at higher levels of abstraction. This work was the foundation of the Process Model traceability and usability characteristics. Since variations occurred across the board, acquisition and engineering standards were interlaced together to portray the overall lifecycle process. In our findings, this proved to be true in real life practices.

The System Engineering process variations appeared again during our program field interviews. Not only was the System Engineering process emphasized differently in other organizations, but it also varied within the same organization. Our Field Interviews were held in four locations- the Naval Air Warfare Center(NAWC)-Aircraft Division Warminster, Warminster PA; Rome

Laboratory-Rome, NY; IBM-Owego, NY; and MDC-DAC-Long Beach, CA. These field interviews revealed that the System Engineering discipline was practiced at the acquisition, development, and sub-System Engineering levels. System Engineers at Rome Laboratory emphasized studies, communication, and program management while practitioners at NAWC emphasized operational needs and specialty engineering practices. At IBM, the emphasis was on the System Engineering process management, and at MDC-DAC, the emphasis was on the program and supplier management and specialty engineering. The System Engineering process variations were diverse among all the aforementioned organizations.

Even in the program management area from government to contractor, we found variations in the System Engineering process. To satisfy this broad base of differences, the SECD Process Model represented government acquisition and contractor development with emphasis in System Engineering activities. Therefore, we concluded that the SECD Process Model supported the three basic groups of the System Engineering roles which are engineering, management, and communication.

Our representation approach was to use the acceptability criteria, previously introduced, to develop a representation methodology and then to select a tool consistent with this criteria. We choose an Macintosh flow charting tool called "*iMacFlow*" because it supports the representation of ANSI standard symbols. These symbols allowed us to better understand and read the SECD Process Model. In addition, "*MacFlow*" supports the hierarchical representation of a process in various interactive modes. This functionality allowed us to abstract complex processes and interactions into simplified representations. Readability was also a discriminating factor of the representations and tools researched.

#### **1.4.2 Abstraction**

Abstraction refers to the hierarchical representation of a process. We followed a set of abstraction guidelines in the development of the Process Model. These guidelines included specific levels of uniformity throughout the model, visibility to major formal reviews, reasonable process functional flow and work products. We integrated "*As-Is*" activities, processes, and work products in a consistent functional flow.

The specific level of uniformity throughout the model was implemented by simply counting the number of symbols represented at each level. In areas where readability or understanding could be compromised, the additional symbols were added or deleted.

Another abstraction guideline was the visibility of major formal reviews. We found that the majority of people can relate and understand the system lifecycle process by following major milestones and formal reviews. This particular abstraction was very difficult because the Process Model represents interaction between Government and Contractor processes. These processes were implemented at a higher level than formal reviews. A judgement was made as to whether or not these processes should be included.

A reasonable functional flow and work products was an abstraction guideline. A series of functional flow revisions were made in order to implement this abstraction guideline. Practitioner process recognition and acceptability were targets. Completeness of work products was inspected against standards and flows. Many reviews were conducted before the Process Model was baselined.

One should refer to the representation methodology, Section 3.1.1, for a detailed explanation of the SECD Process Model symbols, colors and shapes. This section will aid in understanding the figures.

### **1.5 Process Model Documents**

Five documents resulted from the Process Model task. In chronological order, these documents are the following:

1. MDC-DAC Paper
2. National Council On System Engineering (NCOSE) Paper
3. SECD Final Technical Report (FTR)
4. MDC-DAC Final Technical Report (FTR) Supporting Document
5. NCOSE Presentation.

Each document provides a different level of information about the development of the SECD Process Model. The NCOSE presentation and paper was less detailed than the MDC-DAC paper and the MDC-DAC FTR Supporting Document.

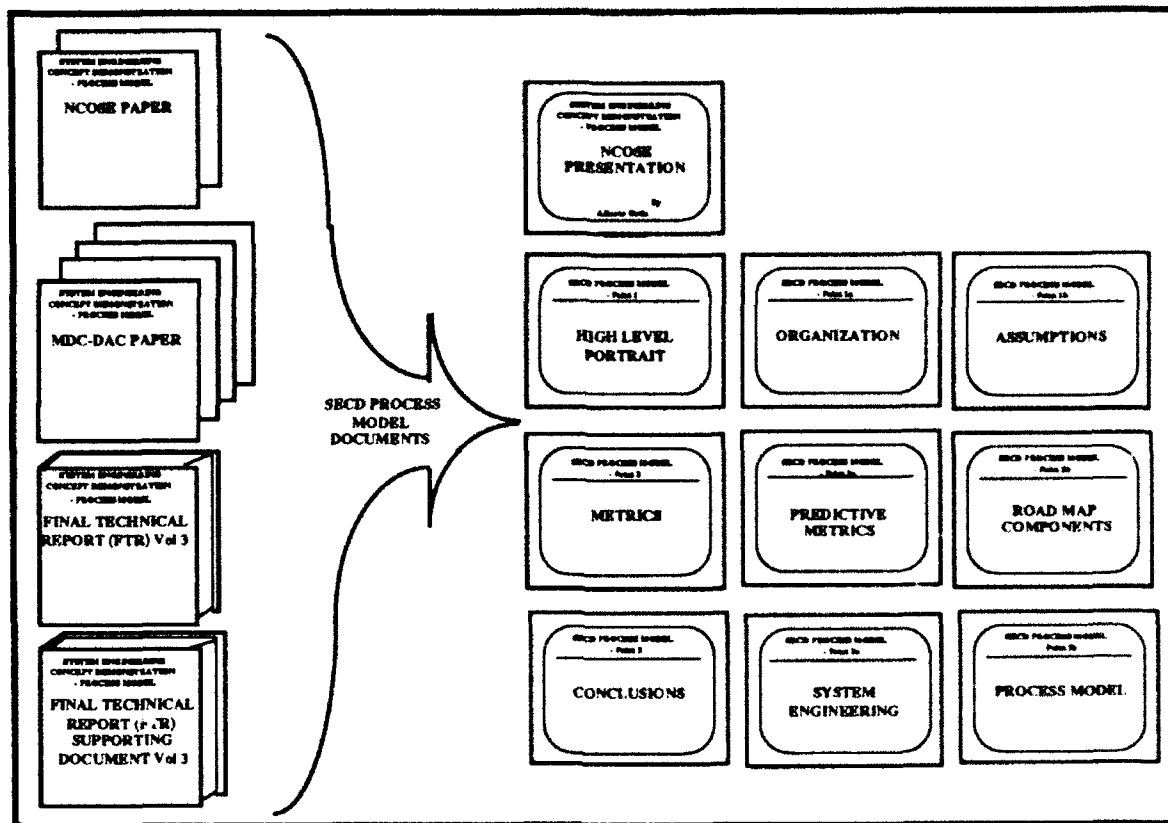


Figure 1.5-1 Process Model Documents

Figure 1.5-1 illustrates the Process Model Documents and the NCOSE Presentation. Other Process Model work products, such as the wall charts and computer software versions, are not represented in this figure.

This page intentionally left blank.

## 2.0 Precedence List Of System Engineering Standards

The precedence list of standards was developed to organize and manage the documents and standards found by the library searches. The precedence list allows us to prioritize, validate and trace activities and processes into commonly used and approved sources. In turn, these sources identify the activities and processes within the system life cycle.

The main advantage of the precedence list of standards is that it resolved numerous conflicts between standards, directives, and documents. The precedence of these standards was determined by the issuing agency (i.e., DoD, Air Force, Navy, Army). The issuing agency also defined the document's relevance of information and the granularity of the information contained within the document. The following paragraphs discuss the precedence list of standards for the SECD Process Model.

### 2.1 DoDI 5000.2

The Department of Defense Instruction (DoDI) 5000.2, *Defense Acquisition Management Policies and Procedures*, dated June 4, 1991, is an acquisition standard. This standard is divided into 15 parts. Each part describes the acquisition process requirements for the system life cycle. According to DoDI 5000.2, each of the following applies to the entire system life cycle:

- acquisition planning and risk management,
- logistics, and
- configuration management.

The later parts of this standard provide applicable procedures for the conduct of program reviews, assessments and acquisition boards. DoDI 5000.2 is a complete standard that contains a clear and detailed acquisition process. It focuses on program management, and it provides insight to government acquisition activities. Although fairly organized, the standard does not provide a thorough reference section and this inadequacy makes it difficult to find information quickly and easily.

Each part of the standard provides a milestone perspective for each phase of the system life cycle. However, the standard does not provide detailed information for the Production and Deployment phase or the Operations and Support phase.

DoDI 5000.2 supports and references other documents in the 5000 series. We used this standard to represent the acquisition process.

## **2.2 MIL-STD-1521B**

The Military Standard (MIL-STD) 1521B, *Technical Reviews and Audits For Systems, Equipments, and Computer Software*, dated June 4 1985, is an engineering and development standard. This standard is organized into three major paragraphs, each of which outlines the standard's topics. Appendices A to K detail these topics and provide an application guide for tailoring the standard.

MIL-STD-1521B outlines the procedures for performing formal reviews. However, the procedures do not provide the necessary details and the check list and certification sheets are not useful in an actual FCA/PCA situation as numerous readiness reviews are held in real life prior to a formal review. MIL-STD-1521B does not capture any readiness reviews and other so called "informal" activities. In the Process Model task, MIL-STD-1521B was used to ensure the presence of all major reviews and to ascertain the engineering work flow.

## **2.3 MIL-STD-973**

The Military Standard (MIL-STD) 973 Draft, *Configuration Management*, dated April 22 1991, is a new configuration management standard. It provides guidance for establishing configuration management requirements for a program. MIL-STD-973 applies to the entire system life cycle and it supersedes the MIL-STD-480's series.

The standard's organization is consistent with other Military Standards and is oriented towards Engineering Change Proposal (ECP) generation and management. It does not contain a very detailed discussion about the processes.

MIL-STD-973 was used in the Process Model task to validate flows of work products in the system life cycle. The Data Item Description (DID) for a Configuration Management Plan was included. Upon approval, this standard should become a commonly used document by all system engineers.

## **2.4 MIL-STD-499A**

The Military Standard (MIL-STD) 499A, *Engineering Management*, dated May 1 1974, is a System Engineering standard. MIL-STD-499A provides the basis for the System Engineering Management Plan. A System Engineering Management Plan (SEMP) captures the planned approach and strategy. MIL-STD-499A provides guidance about technical planning; its content is very generic.

This standard maps well with MIL-STD-1521B and discusses the system engineering process. The information presented is not complete and at times ambiguous. It does not support concurrence or reusability. It is a guide to major system engineering process, activities, and work products, but it lacks instruction about implementation.

## **2.5 MIL-STD-499B**

The Military Standard (MIL-STD) 499B, *System Engineering*, dated May 15 1991, is a System Engineering standard. MIL-STD-499B is still in draft form.

MIL-STD-499B is a process oriented standard. It provides good information concerning technical management and engineering orientation, but does not provide detailed information about the implementation method. This standard is consist with MIL-STD-1521B.

During the Process Model task, standards 499B and 1521B were used to validate and trace system engineering processes, activities, and work products. MIL-STD-499B was helpful in identifying process work products while the overall system engineering process presented in MIL-STD-499A was useful to define the requirement's sub-process for the system engineering process.

## **2.6 DoD-STD-2167A**

The Department of Defense Standard 2167A (DoD-STD-2167A), *Defense System Software Development*, dated February 29 1991, is a software development standard. It provides a detailed and comprehensive description of the software development process and work products. The software development process described within DoD-STD-2167A is tailorable and measurable. DoD-STD-2167A is a military standard, but many commercial software developers use it as a guide to the software development process. MDC-DAC Commercial Aircraft Company uses a commercial version of DoD-STD-2167A called DO-178A.

In the Process Model task, DoD-STD-2167A was used to identify the software development process within the system development life cycle. In the Process Model, software development is considered a separate process and software engineering is considered a specialty. A clear separation was made between the hardware, software, integration, and system engineering development processes. One of the major conclusions in the Process Model task is that different development methodologies can be applied concurrently to the hardware, software, and system engineering development processes.

Although DoD-STD-2167A is a software development standard, it can be tailored and applied to hardware development. This process is evident while allocating



Computer Software Configuration Items (CSCIs). Hardware and software must be synergistic if the system is going to perform effectively.

## **2.7 AFR 800-14**

The Air Force Regulation (AFR) 800-14, *Life Cycle Management Of Computer Resources In Systems*, dated September 29 1986, is an acquisition standard. This standard is part of the AFR 800 series of standards for computer systems and resources. It establishes acquisition policies and procedures for a computer-based system throughout its life cycle.

This standard emphasizes the various organizations within the Air Force. It portrays an overall Air Force structure and the agent's roles and responsibilities in the design, development, production, and support of a computer system. This standard, however, is not completely consistent with others.

In the Process Model task, AFR 800-14 was used to reference and validate activities, processes and flows. Various representation experiments were developed using AFR 800-14. In this regulation, each system life cycle phase is described separately. System activities, processes, and work products are identified within each phase. The processes, activities, and work products within the phases of AFR 800-14 do not integrate well together because there is not enough information provided about the domain boundaries between phases. The acquisition process conveyed is realistic.

## **2.8 S&E Instruction 5451.2**

The Navy Air Warfare Center (NAWC), *Systems and Engineering Group Instruction (S&E INST) 5451.2* is a system engineering instruction. This instruction details the system engineering process currently documented at NAWC. A unique method to capture the system engineering process is used. This method consists of capturing not only the process, but inputs, outputs, work products, and actions by the Headquarters and the system engineer. The entire system life cycle is documented in the instruction.

The S&E INST is a good compendium of various standards and practices and is entirely unique to NAWC. This instruction was a key document in areas where no data about the process was available. Its structure and organization is easy to follow. In the Process Model task, the S&E INST 5451.2 was used as a guide and reference for the overall activities, processes, and work products.

### **3.0 SECD Process Model**

This section presents the SECD Process Model. The Process Model was hierarchically decomposed into various levels of detail, and it represented an integrated portrait of major system engineering standards. As an acquisition and development model, it emphasized the system engineering activities. This emphasis allowed us to make visible the interactions between contractor development and government acquisition. When compared with other models, the SECD Process Model has a number of discriminating factors. This section will show that its most important discriminating factor is its emphasis on communicating information about process interaction.

As discussed in Section One, the model has been captured in electronic form. The remainder of this report provides a paper representation of the upper two levels of the model. This section also discusses the major developmental methods followed to develop the Process Model. It explains the major system life cycle phases and process, abstracted views, discriminating factors, and metrics and instrumentation.

#### **3.1 Description Of Results**

The SECD Process Model can be thought of as an acquisition and development model with strong emphasis placed on system engineering activities. This model is representative of the Government Acquisition and Contractor Development activities and interactions over the entire system life cycle.

The model was developed through a set of representation experiments which were conducted to ascertain a methodology that is consistent with the goals, objectives and acceptability criteria introduced in Section One. These experiments showed that none of the existing representations met the requirements set for the Process Model. Hence, a new representation methodology had to be derived. The Process Model was organized in a hierarchical structure, providing additional levels of detailed for each level of depth. This interactive model presents information in various modes of operation.

##### **3.1.1 Representation Methodology**

A new representation methodology was developed for the Process Model. This methodology was not a tool dependent methodology, and it could be implemented with pencil and paper. The new representation methodology meets customer requirements and the needs and acceptability criteria defined in Section One.

The representation methodology consist of the ANSI standard symbology, consistency rules, and representation steps. The methodology was applied using the Macintosh tool "MacFlow".

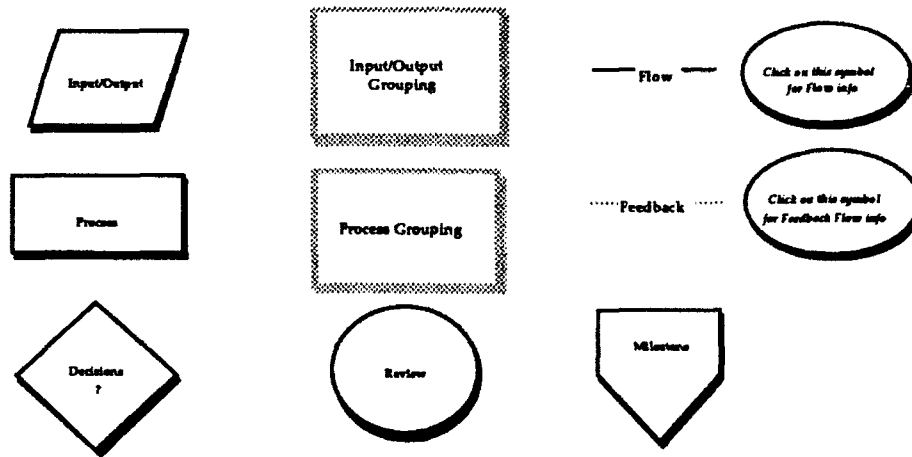
### 3.1.1.1 ANSI Standard Symbology

The ANSI standard symbology was used in the Process Model to aid in the understanding and readability of the representation. Figure 3.1.1.1-1. illustrates the symbology used to represent the Process Model. A total of nine symbols were used to develop the Process Model. These symbols were color coded for better recognition and readability.

The SECD Process Model is an Acquisition, Development, and System Engineering Process Model for the System Engineering Concept Demonstration (SECD)

#### SECD Process Model Symbology

(click on the symbol for additional information on its usage)



Browsing the model, one can find that it operates in three basic operating modes. Those operating modes are described as follows.

- The 1st MODE is "Shadow Zoom"- this mode provides a graphical expansion of the specific process by double clicking the shadowed edge
- The 2nd MODE is "Shadow Comment"- this mode provides a textual comment of the desired
- The 3th MODE "Shadow Launch"- this mode allows you to launch into other applications in the computer system or in any other networked system to execute, retrieve

The shadowed edge around the symbol indicates the current mode of operation of the model. In order to change modes, choose the label "Symbols" from the color Apple Menu Bar at the top and select the desired mode of operation by dragging into the appropriate mode.

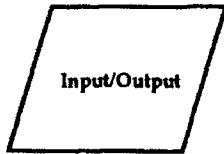
- About Help - this is a symbol that provides information about the process model symbology and color codes. This label is located only at the top level charts of each phase to aid



Double Click in the Shadowed portion of the appropriate symbol to obtain the abstracted information at the next level of detail. The help button provides assistance in the utilization of symbols and flows. Specific information about the desired symbol can be obtain by addressing the appropriate mode of the model

Figure 3.1.1.1.-1 Symbology Used to Represent the Process Model and the Help Window

The SECD Prototype Tradeoff Scenario demonstrates six levels of abstraction in the Process Model. The Process Model symbology is explained within the "Help" icon for the upper three levels of abstraction. The Macintosh tool "MacFlow" was used to develop these views and each view has a Help Window. Figure 3.1.1.1-1. also shows the Help Window. Figure 3.1.1.1-2 shows the Input/Output Symbol Page View.

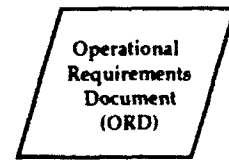


An Input/Output represents tangible data that may be input to, or output from a process or review. The same symbol is used for both inputs and outputs since an output of one process will likely be an input to another.

Input/Output labels are nouns

An Input/Output may be:

A Formal Document  
or  
Deliverable



A Report  
or White Paper

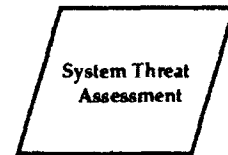


Figure 3.1.1.1-2 Input/Output Symbol Page View

Table 3.1.1.1-1. Process Model Symbology Description presents the symbols and their associated descriptions and color codes.

Table 3.1.1.1-1 Process Model Symbology Description

Symbol	Description	Color Code
Parallelogram	Input/Output	Turquoise
Rectangle	Process	Light Purple
Diamond	Decision	Yellow
Circle	Review	Green
Tombstone	Milestone	Hot Pink
Hashed Rectangle	Process Grouping	Light Purple (Perimeter Only)
Cross Hashed Rectangle	Input/Output Grouping	Turquoise Perimeter Only)
Line	Flow	Blue
Hashed Line	Feedback	Dark Red

### 3.1.1.2 Consistency Rules

Consistency was one of the acceptability criteria factors presented in Section One. This criterion improved the degree of readability in the Process Model. In order to implement consistency rules throughout the Process Model, a set of page views was developed to convey the "do's" and "don't's" of our representation methodology. These views are presented beneath each symbol in the Help Page View of the Process Model (Figure 3.1.1.1-2). In our discussion, we present a page view for each symbol identified in Table 3.1.1.1-1 and a brief description of their applicable consistency rules. The consistency rules are the key to understanding the Process Model representation. The representation also meets the acceptability criteria factor of readability.

## Input/Output –

The Input/Output Symbol Page View is shown in Figure 3.1.1.2-1.

Input/Output labels are nouns. This symbol represents formal and informal work products (reports, letters, specifications, etc.) throughout the system life cycle. This symbol can be found as an input or an output to a process, review or decision symbols.

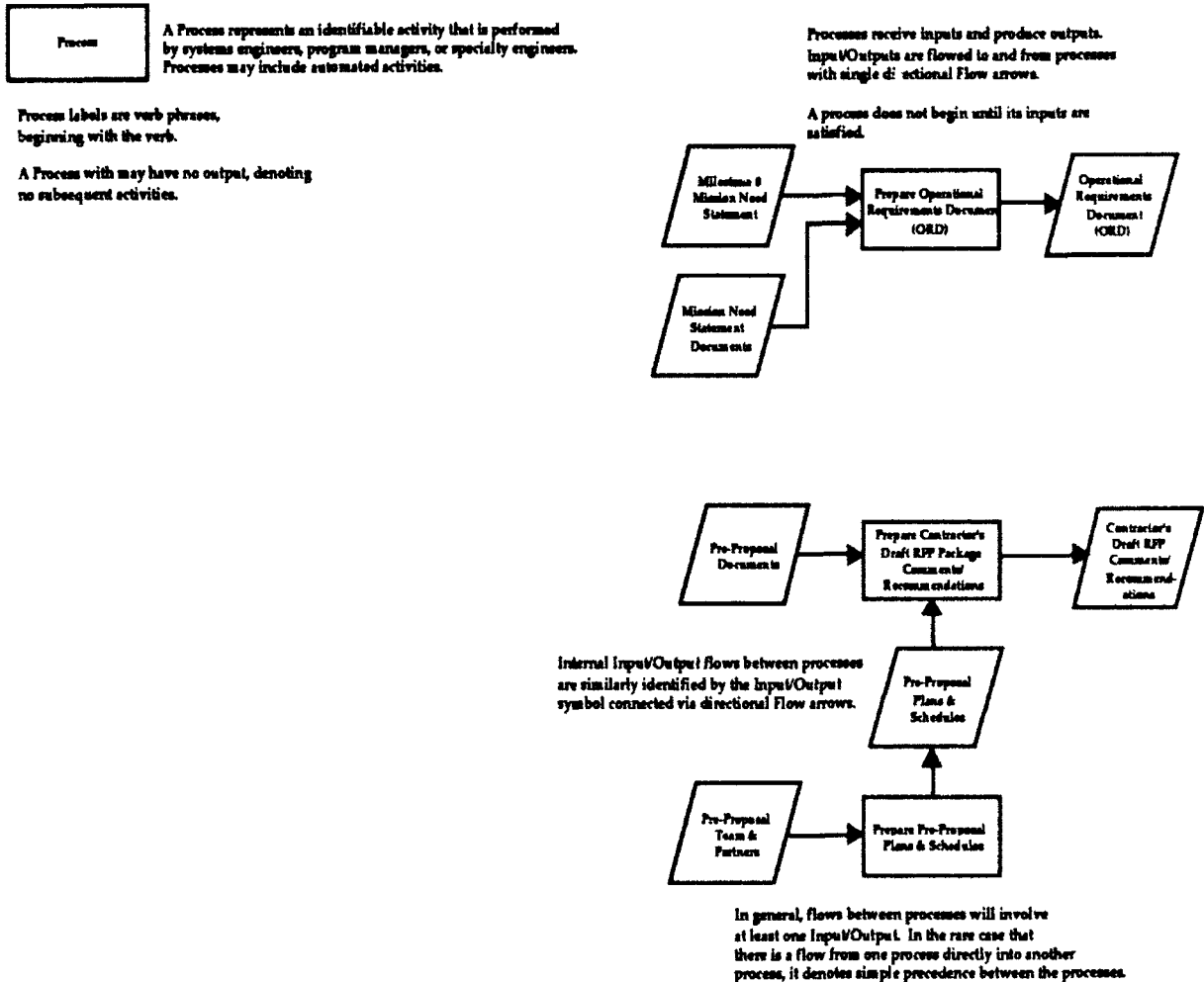


Figure 3.1.1.2-1. Input/Output and Process Symbol Page View

## Process -

The Process Symbol Page View is also shown in Figure 3.1.1.2-1. This page view illustrates that it is possible to have multiple Input/Output symbols to and from a process. Notice there are no process to process flows in the page view. This rule identifies all flows between processes by using the input/output symbol; it

also helps eliminate flow ambiguities. Process labels are verb phrases beginning with a verb, and they are hierarchically decomposed in the Process Model.

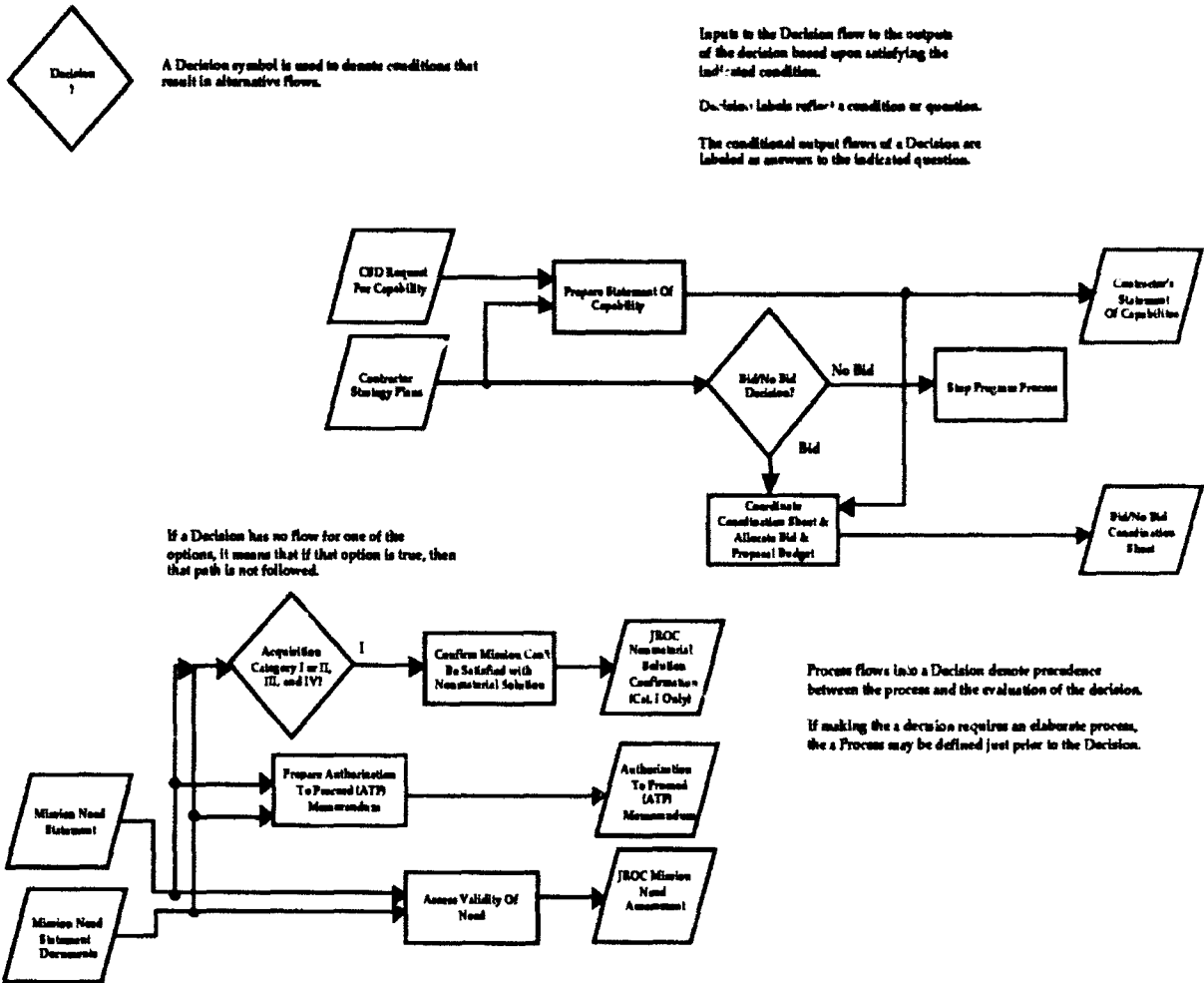


Figure 3.1.1.2-2. Decision Symbol Page View

### Decision -

The Decision Symbol Page View is shown in Figure 3.1.1.2-2. The decision symbol is used to denote conditions that result in alternative flows. Decision labels reflect a condition or a question. Notice that a process can flow into a decision or vice versa. This flow pattern represents a precedence between the two symbols. A condition must be satisfied before a process can be completed. Other rules for the decision symbol are detailed in Figure 3.1.1.2-3.

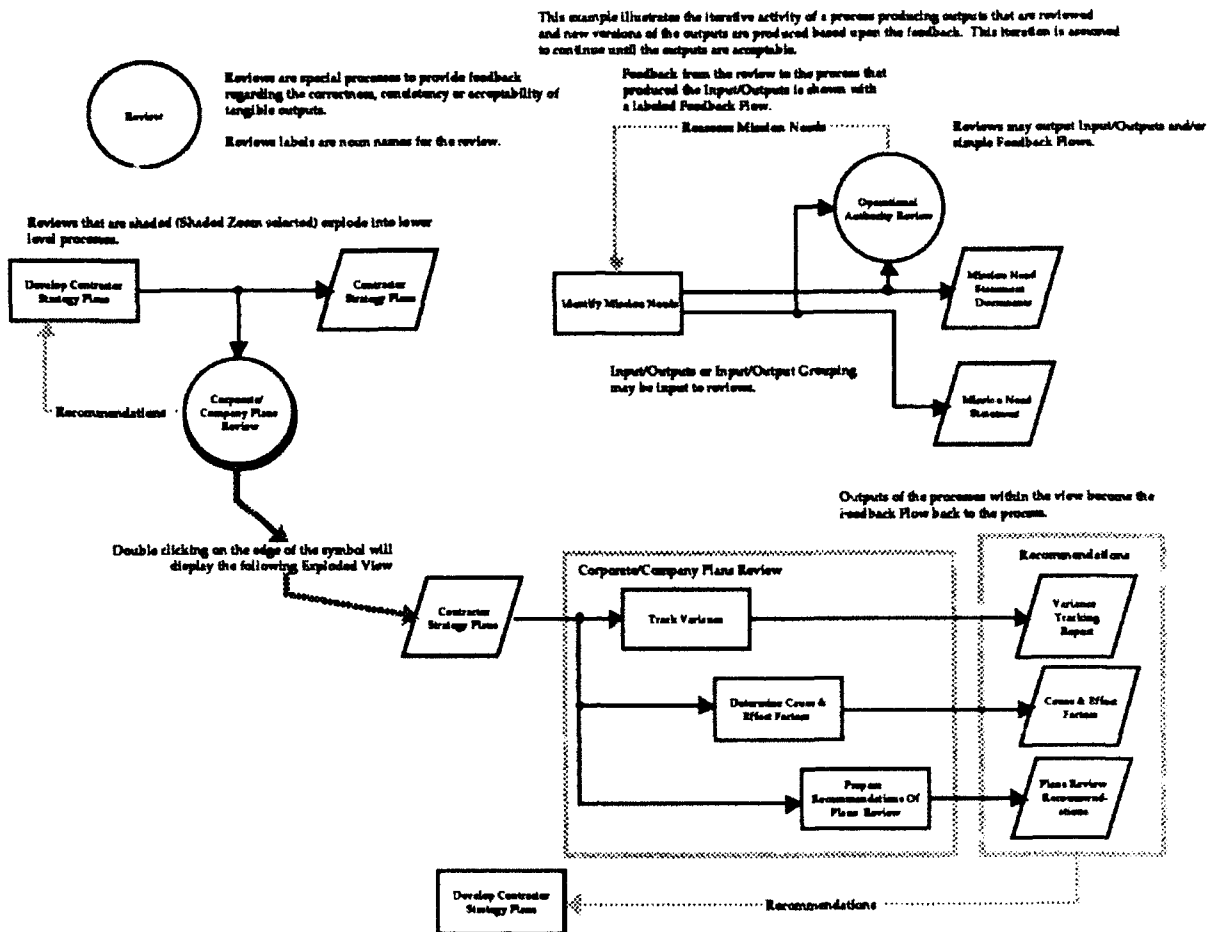


Figure 3.1.1.2-3. Review Symbol Page View

## Review -

The Review Symbol Page View is shown in Figure 3.1.1.2-3. Reviews are special processes that provide feedback with regard to outputs. Review labels are noun names for the review. The review symbol is used in the Process Model to provide iterations of outputs without flowing new versions of the work products. This iteration continues until the work products are acceptable. This assumption allows us to simplify the representation.



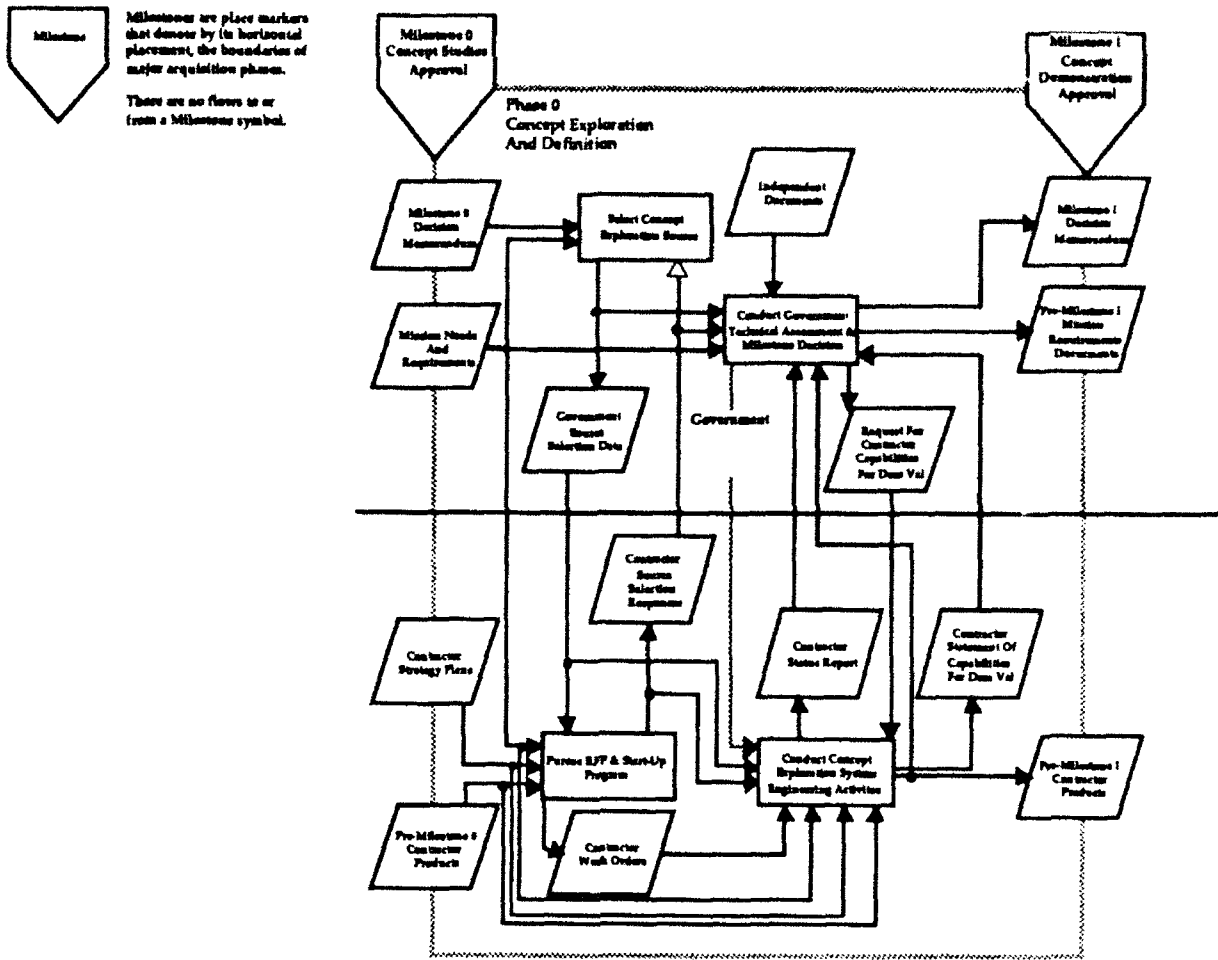


Figure 3.1.1.2-4. Milestone Symbol Page View

**Milestone -**

The Milestone Symbol Page View is shown in Figure 3.1.1.2-4. Milestones are place markers that denote, by their horizontal placement, the boundaries of major acquisition phases. There are no flows to, or from, a milestone symbol. In our representation, milestones are boundaries of major acquisition phases.

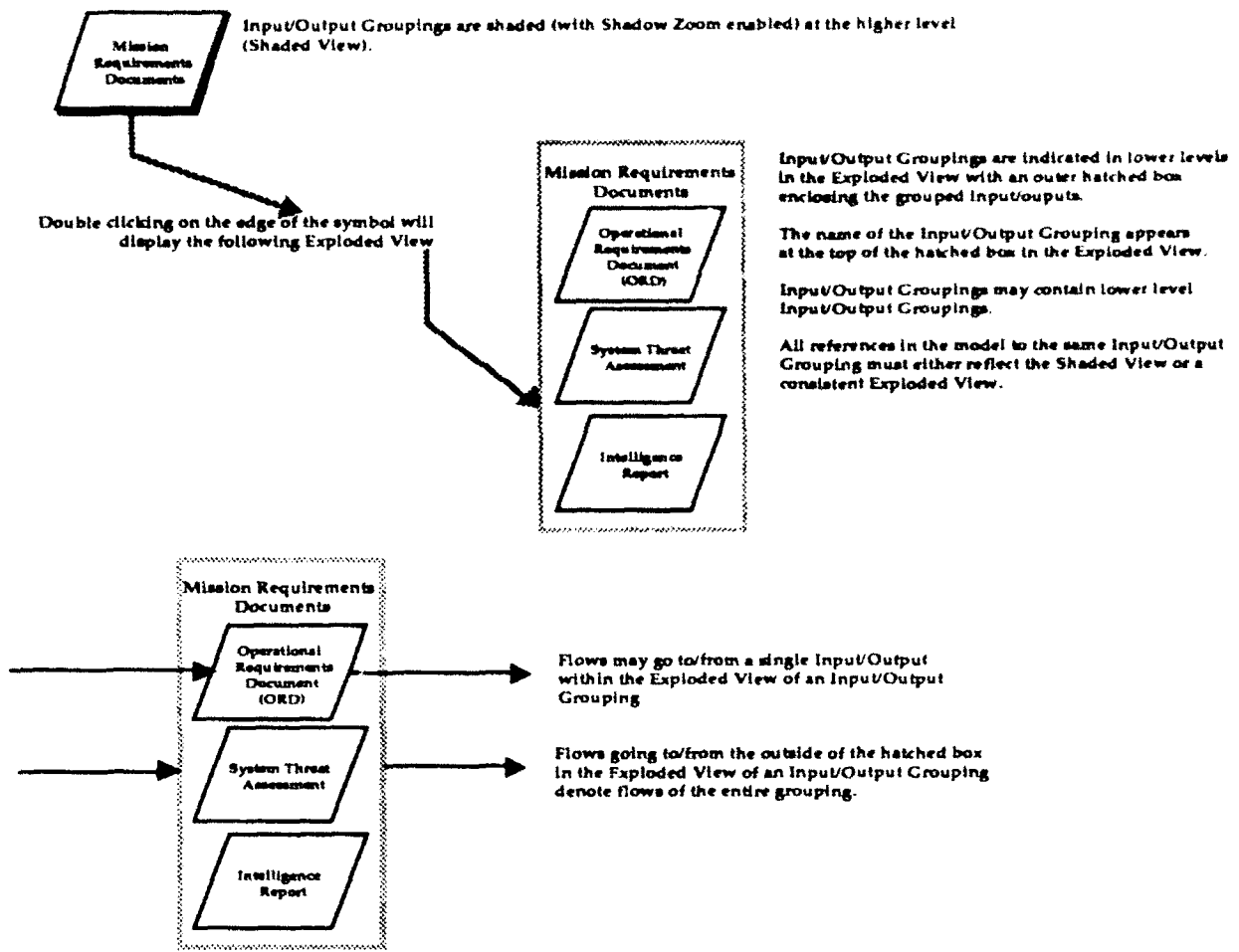


Figure 3.1.1.2-5. Input/Output Grouping Symbol Page View

### Input/Output Grouping -

The Input/Output Page View is shown in Figure 3.1.1.2-5. The input/output grouping symbol is shaded at a higher level. At the lower level, the input/output grouping symbol is used to show the hierarchical decomposition of the input/output symbol. The input/output grouping symbol contains only input/output symbols. In the Process Model, flows are allowed to the input/output grouping and input/output symbols. Flows to the input/output grouping are inputs or output from, or to, each input/output symbol in the grouping.

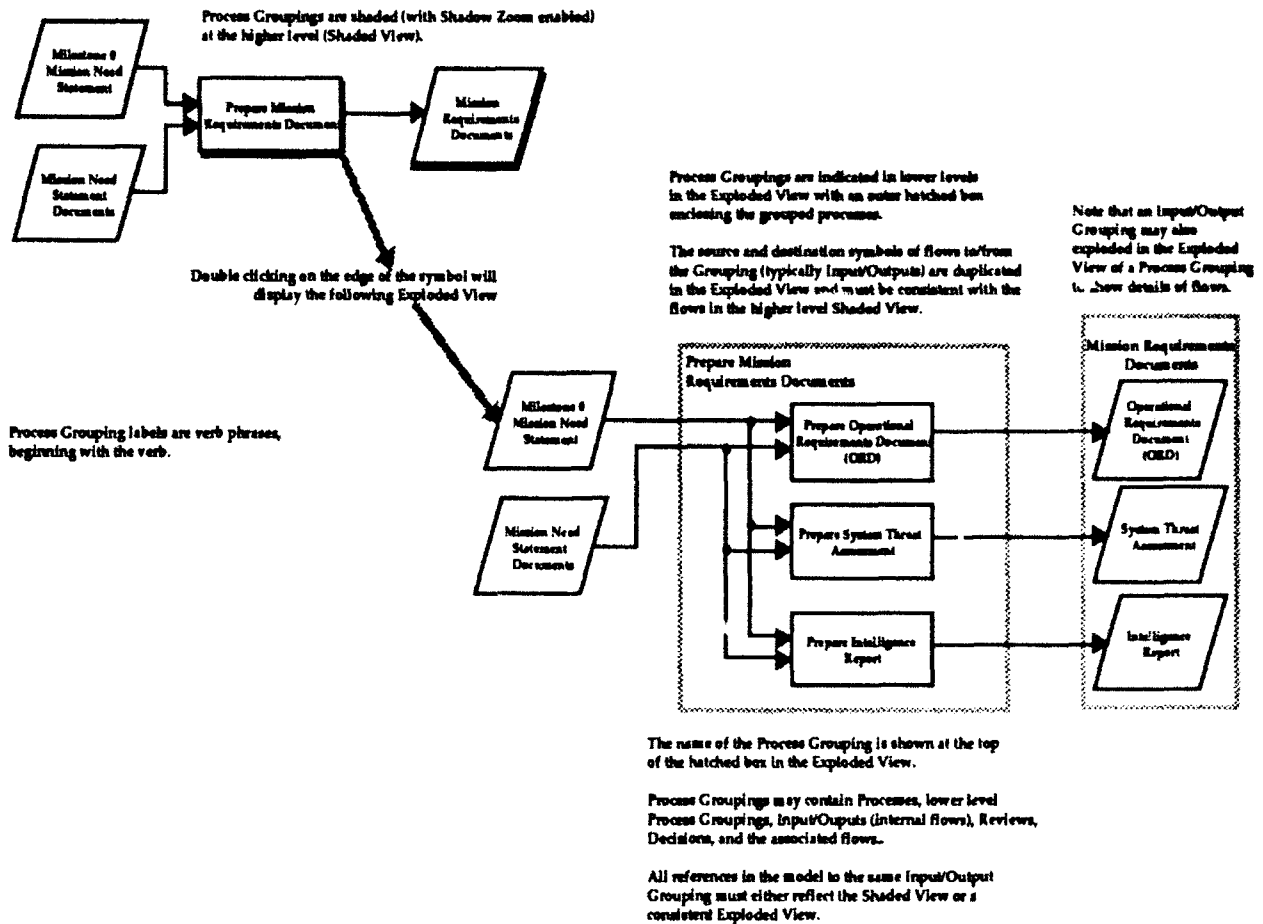


Figure 3.1.1.2-6. Process Grouping Symbol Page View

### Process Grouping -

The Process Grouping Page View is shown in Figure 3.1.1.2-6. The process grouping symbol is shaded at a higher level. At a lower level, the process grouping symbol is used to show the hierarchical decomposition of the process symbol. In the Process Model, flows are allowed to processes, reviews, and decision symbols within the process grouping symbol. Flows to any of the Process Model symbols are inputs or outputs from or to each symbol in the grouping. Notice that at the highest level, the entire system life cycle is grouped using the process grouping symbol.

- Flow → Flows generally connect Input/Outputs with Processes, Process Groupings, Reviews, and Decisions.
  - ▬ This symbol is used to collect Flows for diagrammatic readability. It is called "Collector" and it flows all inputs to output.
  - Multiple destination (or source) flows are indicated as branching lines.
- Alternatively, Flows denoting precedence may connect Processes or Process Groupings with Reviews or Decisions, or between Reviews and Decisions.

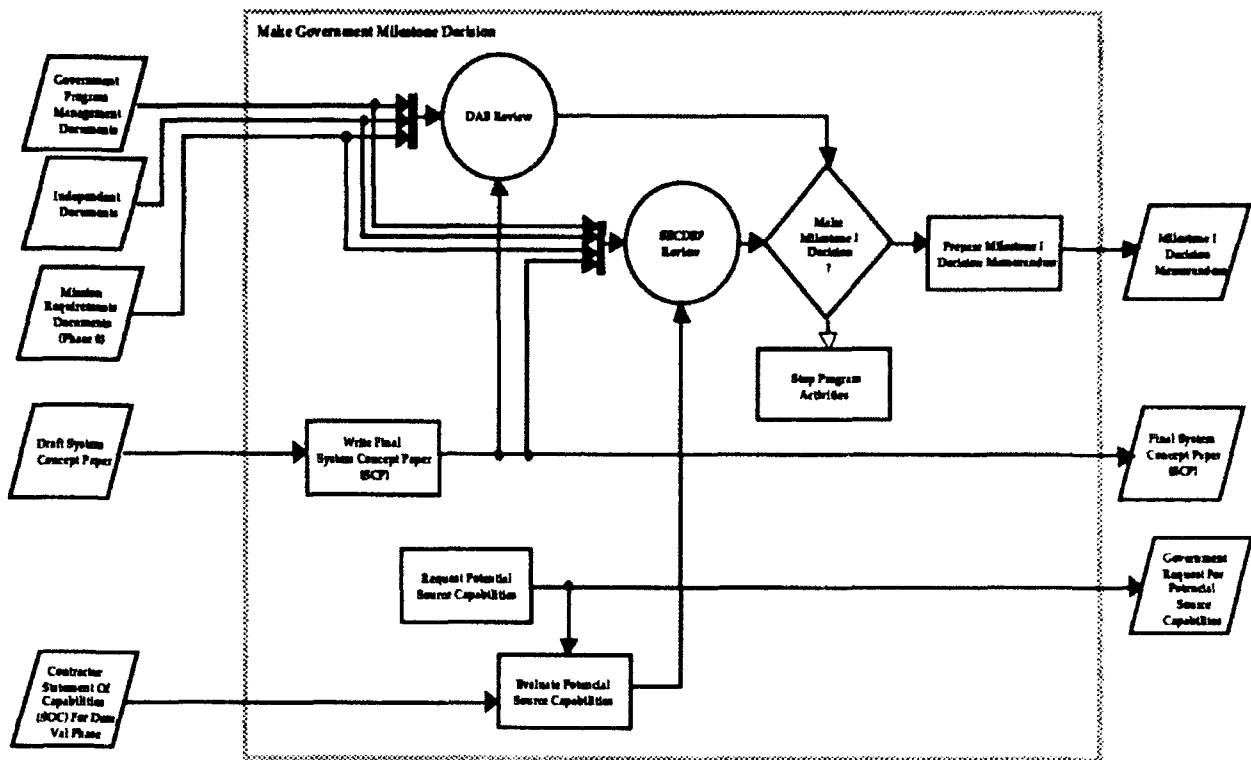


Figure 3.1.1.2-7. Flow Symbol Page View

**Flow -**

The Flow Page View is shown in Figure 3.1.1.2-7. The flow symbol is represented by a line with an arrow to indicate the direction of the flow. Flows generally connect symbols between each other. In the Process Model, it is possible to have multiple destination flows in different directions. An additional symbol, called "Collector" is used to collect flows for readability and simplification purposes. Flow can only go in one direction within each arrow. This rule solves various ambiguity problems in the representation.

Feedback

Feedback flows demonstrate the iterative feedback to processes based upon review of their resultant Outputs.

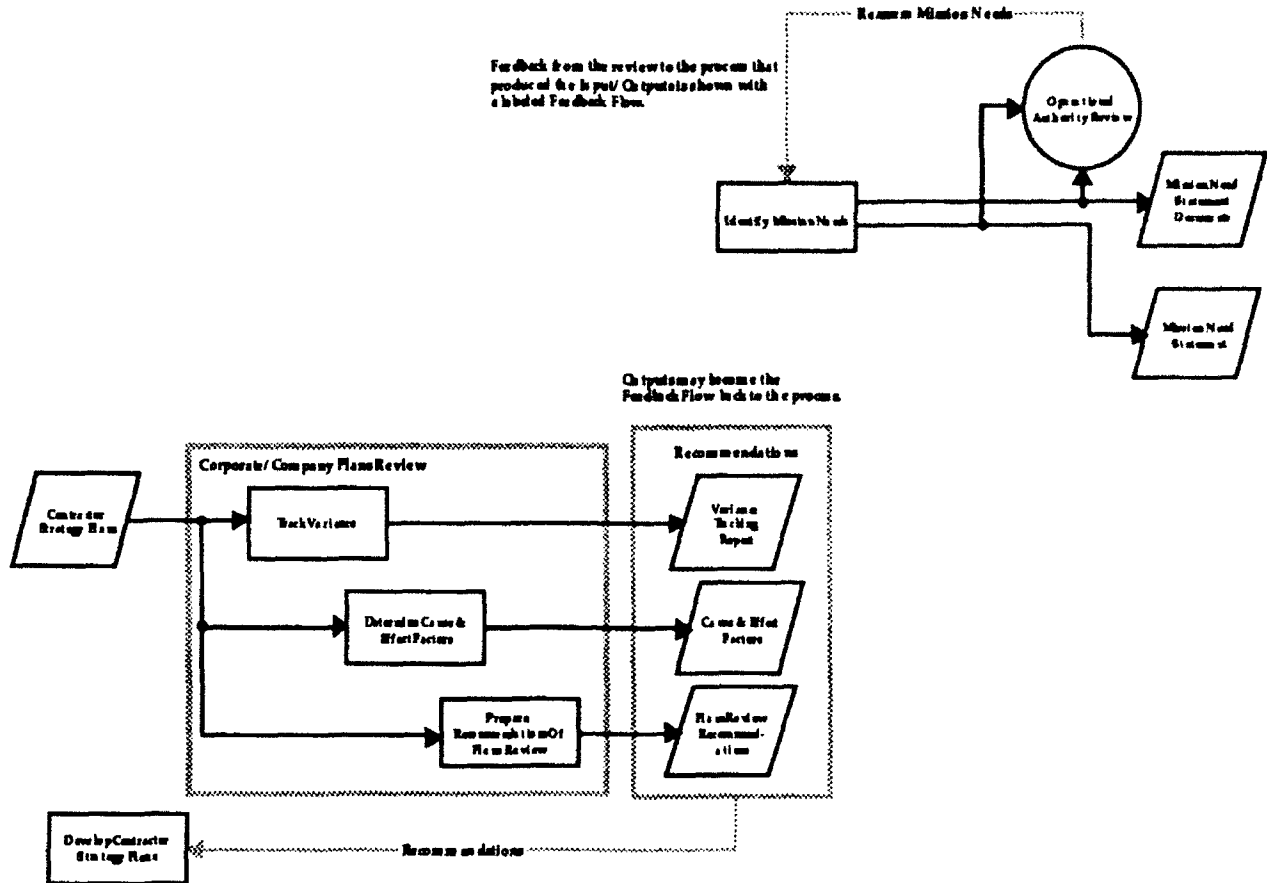


Figure 3.1.1.2-8. Feedback Symbol Page View

### Feedback -

The Feedback Page View is shown in Figure 3.1.1.2-8. The Feedback symbol is represented by a dashed line and an arrow to indicate the direction of the feedback. In the Process Model, it is possible to have multiple destination feeds with different directions. An additional symbol, not shown in Figure 3.1.1.2-9, is a "Feedback Collector". This symbol is used to collect feedbacks in order to simplify and improve readability. It is possible to have "forward" feedbacks, but the symbol mainly feeds information backwards about a formal review to a given process.

### 3.1.1.3 Representation Steps

The Process Model Representation Steps were devised to provide a systematic and consistent approach to the modeling of the process. These steps were

essential in communicating representation ideas to the reviewers in the SECD program.

To develop the representation steps, we followed the same development steps used for the Process Model. The representation steps are included in this report to communicate the complexity and difficulty involved in modeling a process task. The steps are as follows:

**Step 1** - Place milestone symbols at the boundaries of the system life cycle phase being modeled. Use DoDI 5000.2 and 5000.1 to identify terminology and name labels.

**Step 2** - Place major review symbols within the boundaries defined in Step 1. Use MIL-STD-1521B and MIL-STD-973 to identify terminology and name labels. Refer to AFR 800-14 and DoD-STD-2167A for validation of terminology.

**Step 3** - Place major input/output symbols within the system life cycle phase under development. Use the precedence list of standards and source documents to identify the necessary transition work products from one phase of the system life cycle to another.

- a) Assign outputs
- b) Assign inputs
- c) Flow down outputs
- d) Flow down inputs

**Step 4** - Place major decision symbols in the system life cycle phase. Identify major decision points in the system life cycle. Represent formal and informal decisions as part of the process.

**Step 5** - Place major process symbols in the system life cycle phase. Represent major processes and add informal ones to augment the understanding of the overall process. Group activities into formal and informal processes. Use the precedence list of standards and other source documents to validate the terminology.

**Step 6** - Place the flow and feedback symbols in the system life cycle phase. Use the narrative provided in the precedence list of standards and source documents to obtain an understanding of the interaction and the roles in the process. Ensure that the level of abstraction is consistent with established acceptability criteria and uniformity at the top three levels is consistent.

**Step 7** - Conduct a final horizontal and vertical functional flow inspection to determine fidelity, completeness, and flow of the processes and phase.

Present results, rationale, assumptions, and conclusions to program reviewers.

**Step 8 - Add comments and baseline processes within the Process Model.**

### **3.1.1.4 Organization**

This section describes the organization of the SECD Process Model within the context of the Macintosh tool "MacFlow". (The reader is advised to review the "MacFlow" user's manual to better understand the terminology used in the following discussion.)

The Process Model is organized in hierarchical levels of abstraction. The first level expands graphically into a more detailed representation of the system life cycle. The Process Model is integrated to allow this type of interactive navigation. Figure 3.1.1.4.-1, Process Model Organization, illustrates the upper three levels of the Process Model. These levels can be accessed by double clicking in the shaded border of the Process Grouping symbol.

**Level 1.** The first level of the Process Model is called "*System Acquisition And Development Process Model With Emphasis On System Engineering Activities.*" It represents the system life cycle activities for each phase of system development. The milestone symbols are used to illustrate boundaries between phases while process symbols are used to illustrate system life cycle phases. These symbols are enclosed by a process grouping symbol. This enclosed organization is one of the major assumptions of the Process Model task. Although the real life organization could not be enclosed, this assumption was necessary to scope the task and establish the domain boundaries. The four symbols represented at the lower left hand side of the first level in Figure 3.1.1.4-1. are the Help Page View, the "*Shadow Zoom*" (Graphical Expansion) mode, the "*Shadow Comment*" (Textual Expansion) mode, and the "*Shadow Launch*" (Application Jump) mode. Each symbol on the screen can be enacted in all three modes of the Process Model.

**Level 2.** The second level of the Process Model is called "*System Acquisition and Development Top Level Process Model with Emphasis in System Engineering Activities.*" It expands the first level and sub-divides the system development phase between government acquisition and contractor development. The four symbols on the lower left hand side of the second level represent exactly the same functions identified in the first level. At this level, the milestone symbols can be seen inside each phase, denoting boundaries between system development phases.

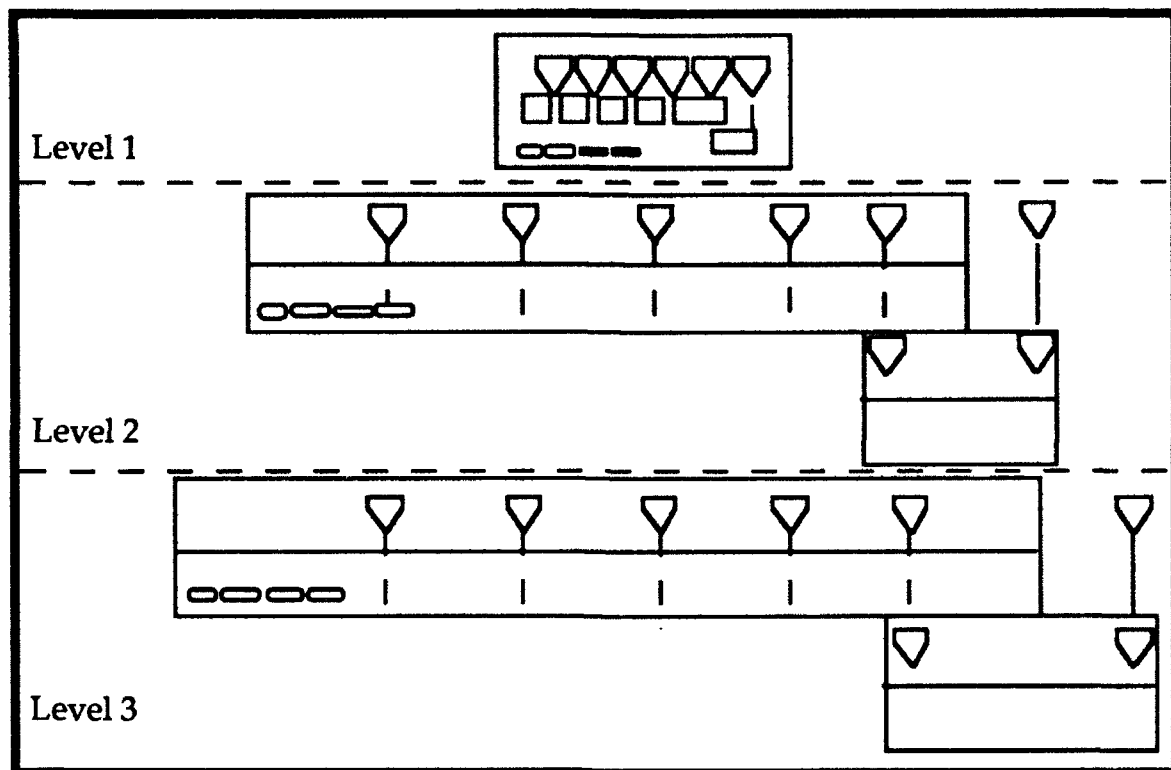


Figure 3.1.1.4-1. Process Model Organization

**Level 3.** The third level of the Process Model is named in accordance with each individual phase (i.e., Determination Of Mission Needs, Concept Exploration and Definition, Demonstration and Validation, Engineering & Manufacturing Development, Production & Deployment, and Operations and Support). At this level of abstraction, lower level processes and input/output symbols can be seen. The separation between Government Acquisition and Contractor Development is maintained. The three system engineering process groups are now visible at this level. These groups are requirements, analysis and planning. A total of eight levels of varied detail are available through most individual process symbols.

### 3.1.1.5 Assumptions

A number of assumptions were made during the development of the SECD Process Model. This section documents the most relevant assumptions to provide an understanding of the perspectives that were used during the development of the model. Some of these assumptions were necessary to establish domain boundaries in our representation.

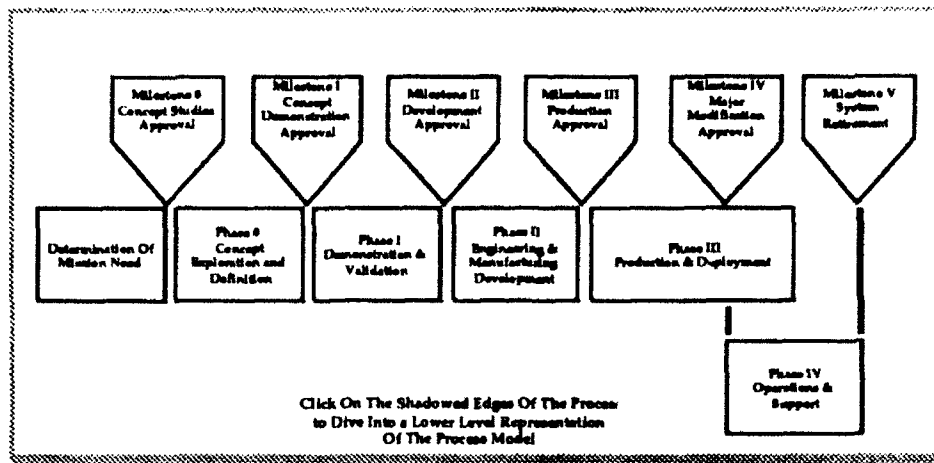


- a. The system life cycle is a closed process.
- b. The hardware and software development processes are a specialty engineering discipline.
- c. The system integration process is a specialty engineering discipline.
- d. The system engineering process cannot be separated from the system development or acquisition process.
- e. It is necessary to represent both the government acquisition and contractor development decisions in order to completely represent the system engineering process.
- f. A precedence list of standards is necessary to resolve conflicts and ambiguities between standards, and to trace and validate processes
- g. The order in which symbols are represented in the model will impact the final perspective illustrated by the model.
- h. The Process Model must be generic in order to be applicable to different types of systems, system development approaches, and organizational variations.

### **3.2 Description Of Major System Life Cycle Phases And Processes**

Representations of processes were conceived in order to develop the Process Model. Figure 3.2-1, System Acquisition And Development Process Model With Emphasis on System Engineering Activities, illustrates the Department of Defense (DoD) acquisition process as defined in DoDI 5000.2. This process was used to define the system life cycle which is the first level of the Process Model. The figure contains a list of standards used to define the system engineering life cycle processes, reviews, milestones, feedbacks, flows, and work products.

The three ovals at the bottom of the page represent the modes of operation in the Process Model as well as the informational views available to the user. The Help oval provides a tutorial about the Process Model representation rules.



**Process Model Precedence List Of Standards:**

- DoDI 5000.2, " Defense Acquisition Management Policies and Procedures ", February 23, 1991
- MIL-STD-1521B, " Technical Reviews and Audits for Systems, Equipments, and Computer Software ", June 4, 1991
- MIL-STD-973, " Configuration Management ", Undated Draft
- MIL-STD-499A, " Engineering Management ", May 1, 1974
- MIL-STD-499B, " System Engineering ", Undated Draft
- DOD-STD-2167A, " Defense System Software Development ", February 29, 1968
- AFR 800-14, " Lifecycle Management Of Computer Resources In Systems ", September 29, 1986
- S&E INST 5451.2, " System Engineering Technical Support Implementation ", September 21, 1987

Double Click in the Shadowed portion of the appropriate symbol to obtain the abstracted information at the next level of detail. The help button provides assistance in the utilization of symbols and flow. Specific information about the desired symbol can be obtain by addressing the appropriate mode of the model.

Figure 3.2-1 System Acquisition And Development Process Model With Emphasis On System Engineering Activities

Additionally, we describe each of the system life cycle phases and a relevant process within the Engineering & Manufacturing Development Phase and the interaction between Production & Deployment and Operations & Support Phases. Figure 3.2.4.1-1 and 3.2.5.1-1. illustrate these process descriptions.

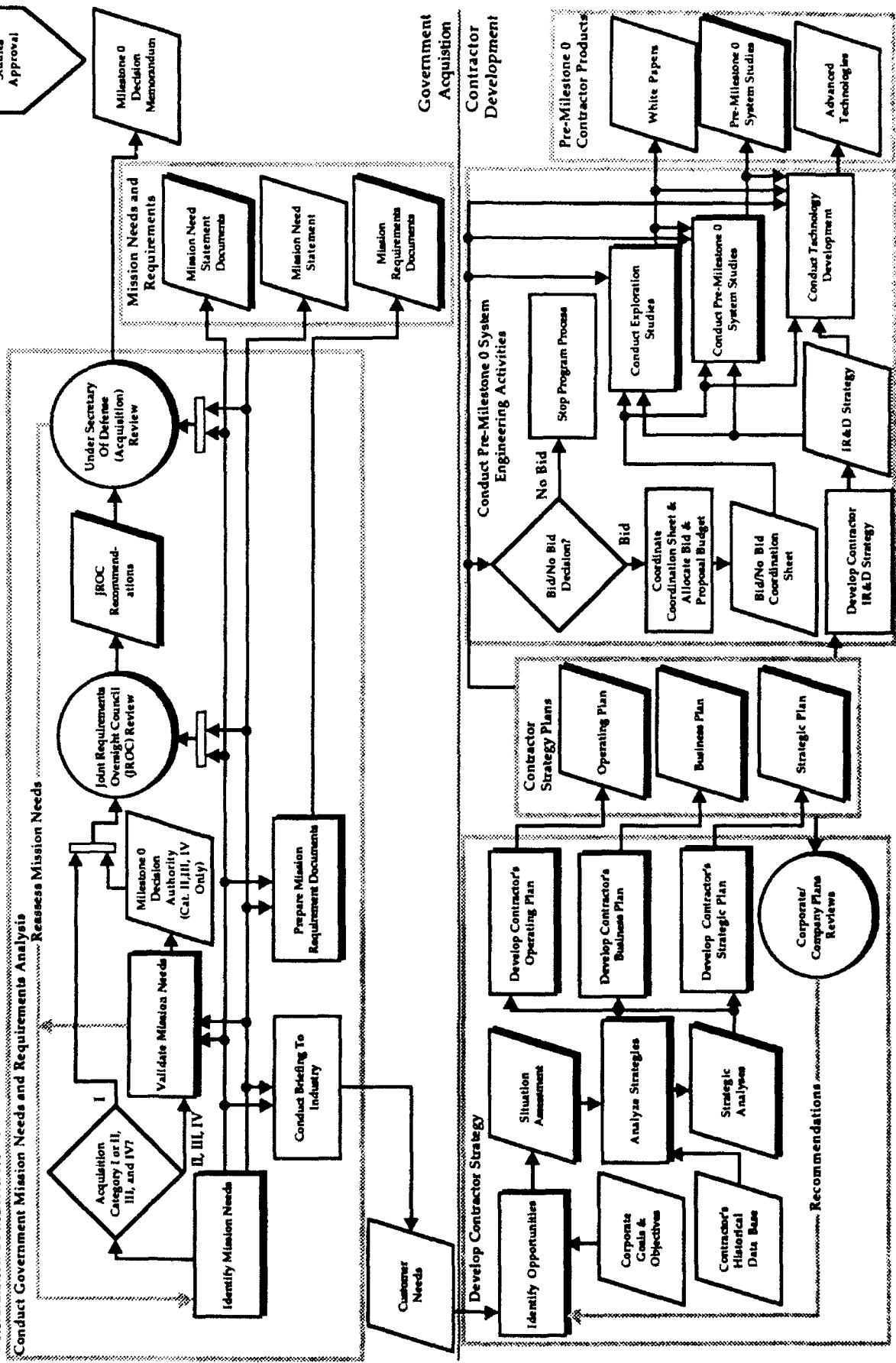
**3.2.1 Determination Of Mission Needs Phase**

The government establishes a clear Statement of Need (SON) and a set of Mission Need Documents in the Determination Of Mission Needs Phase. This phase defined the acquisition, strategic, operational, and tactical needs. Material and Non-Material solutions are evaluated to determine the best possible alternatives to satisfy the defined need. The Contractor identifies its business strategies and plans and aligns its research and development program goals with the established government needs.

It is important to understand that the contractor may or may not provide relevant technical data to government decision makers during this process. The

contractor must ascertain the value of the potential business and incorporate this decision into their business, operational, and strategic plans. Figure 3.2.1-1 illustrates the third level of detail of the Determination of Mission Needs Phase.

**Rome Laboratory**  
**Determination Of Mission Needs**  
**Figure 3.2.1-1 Determination Of Mission Needs**



**HELP!**

"Shadow Zoom" Selected

"Shadow Comment" Selected

"Shadow Launch" Selected

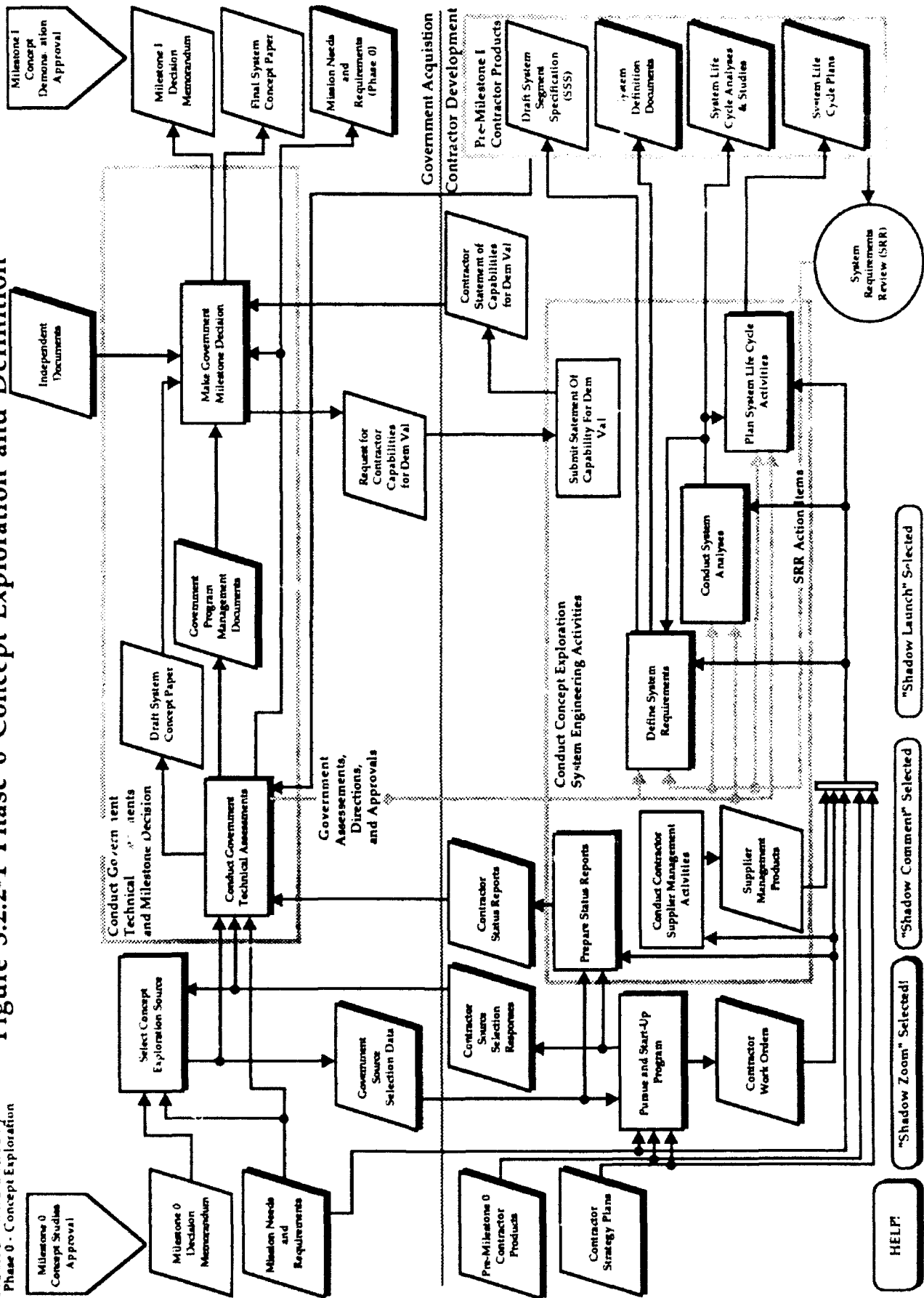
Double Click in the shadowed portion of the appropriate symbol to obtain the abstracted information at the next level of detail. The help button provides assistance in the utilization of the symbols and flows. Specific information about the desired symbol can be obtained by addressing the appropriate mode of the model.

### **3.2.2 Concept Exploration Phase**

In the Concept Exploration Phase, the government evaluates and defines various concepts to determine their feasibility. These studies are used to reduce risk and to identify high pay-off areas. The contractor develops some of these studies for the government and establishes a more detailed definition of the system. The system engineering process is used to provide the first level of definition for the system. A System Requirements Review (SRR) is held during this phase to determine the readiness and completeness of the system requirements. A Technical Performance Measurement (TPM) program is established to monitor and ascertain design characteristics. Figure 3.2.2-1 illustrates the third level of detail of the Concept Exploration Phase.

Rome Laboratory  
Phase 0 - Concept Exploration

### Figure 3.2.2-1 Phase 0 Concept Exploration and Definition

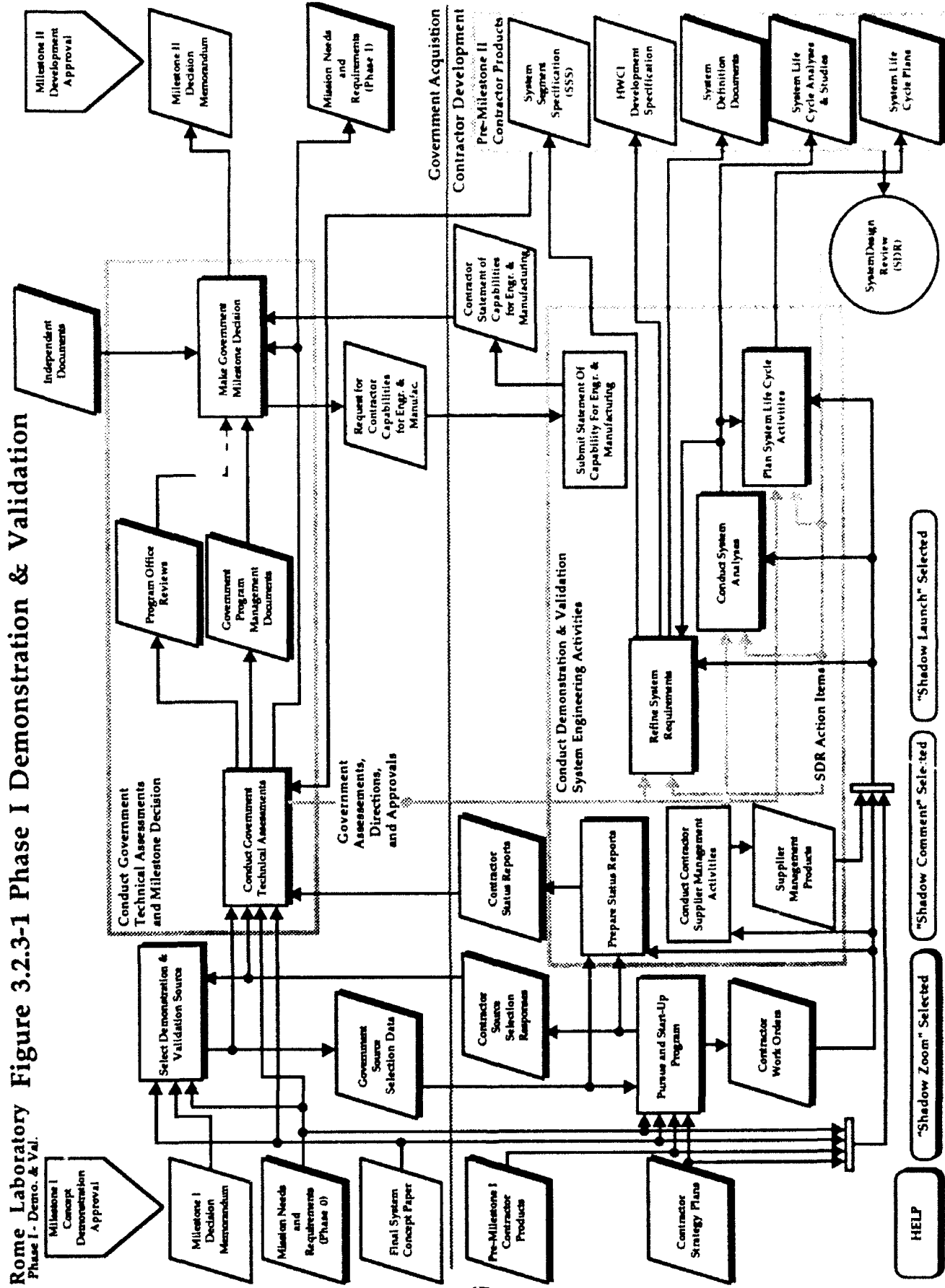


Double Click in the shadowed portion of the appropriate symbol to obtain the abstracted information at the next level of detailed. The help button provides assistance in the utilization of the symbols and flows. Specific information about the desired symbol can be obtained by addressing the appropriate mode of the model.

### **3.2.3 Demonstration And Validation Phase**

In the Demonstration and Validation Phase, the government establishes a developmental baseline and then prototypes and demonstrates various system technologies and concepts. Detailed designs allow for the refinement of the system requirements using the system engineering process. Risk is reduced by prototyping and integrating new technologies. This phase is not always feasible from a cost point of view. The processes in this phase support a Milestone II decision. A System Design Review (SDR) is held to determine how well the designs meet the established system requirements. Figure 3.2.3-1 illustrates the third level of detail of the Demonstration And Validation Phase.

Rome Laboratory Figure 3.2.3-1 Phase I Demonstration & Validation



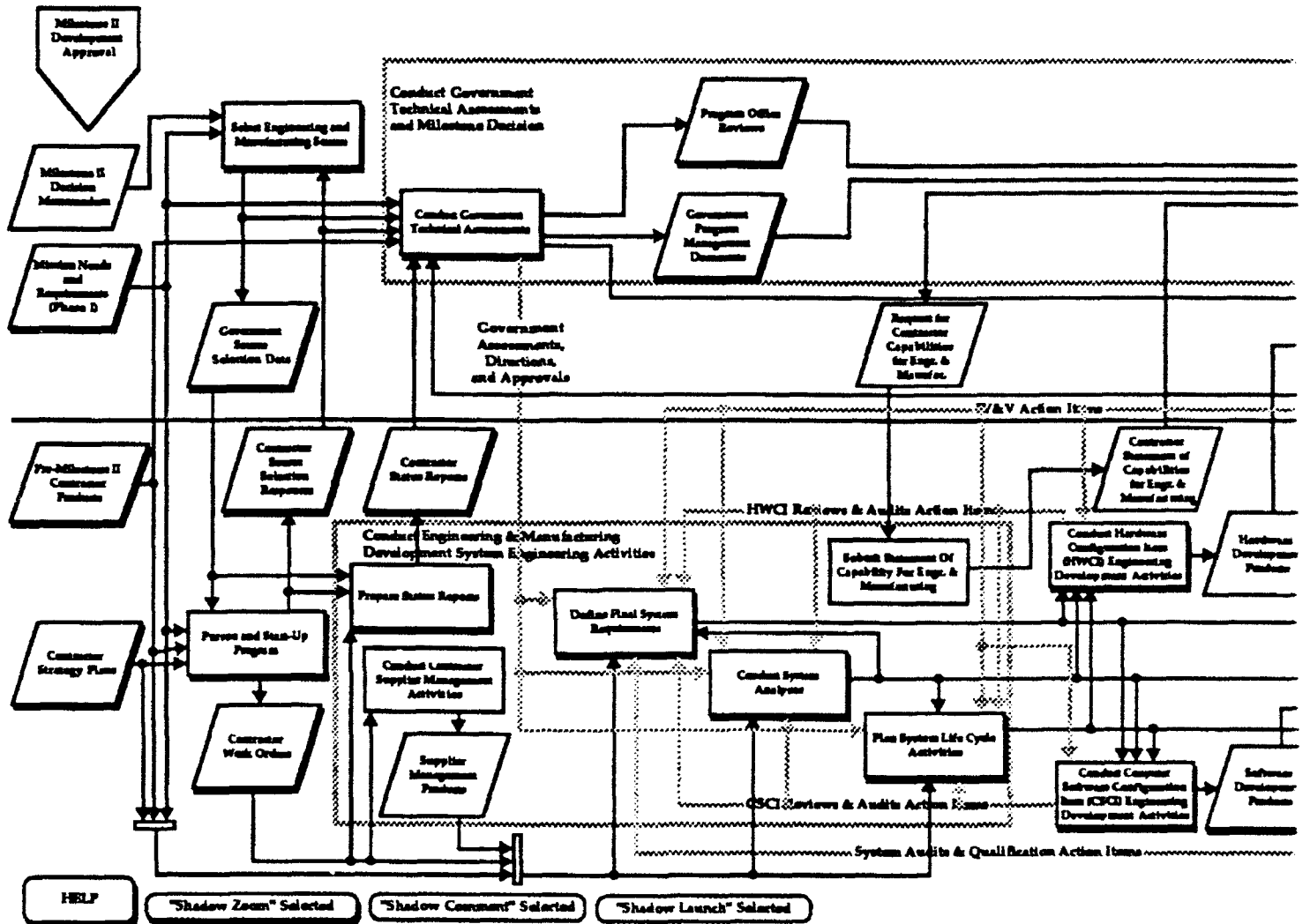
Double Click in the shadowed portion of the appropriate symbol to obtain the abstracted information at the next level of detailed. The help button provides assistance in the utilization of the symbols and flows. Specific information about the desired symbol can be obtained by addressing the appropriate mode of the model.



### **3.2.4 Engineering And Manufacturing Development Phase**

In the Engineering and Manufacturing Development Phase, the government and contractor establish a process and define the tools to produce the product. The system designs are finalized and a production baseline is established. Testing and integration is used to validate performance parameters for a production line. A Preliminary Design Review (PDR) and Critical Design Review (CDR) are held to baseline the design and test procedures and plans. A Functional Configuration Audit (FCA), Physical Configuration Audit (PCA), and a Formal Qualification Review (FQR) are held to finalize the production configuration of the system. The emphasis in this phase is in the system test, integration and manufacturing readiness. Figure 3.2.4-1 illustrates the third level of detail for the Engineering And Manufacturing Development Phase.

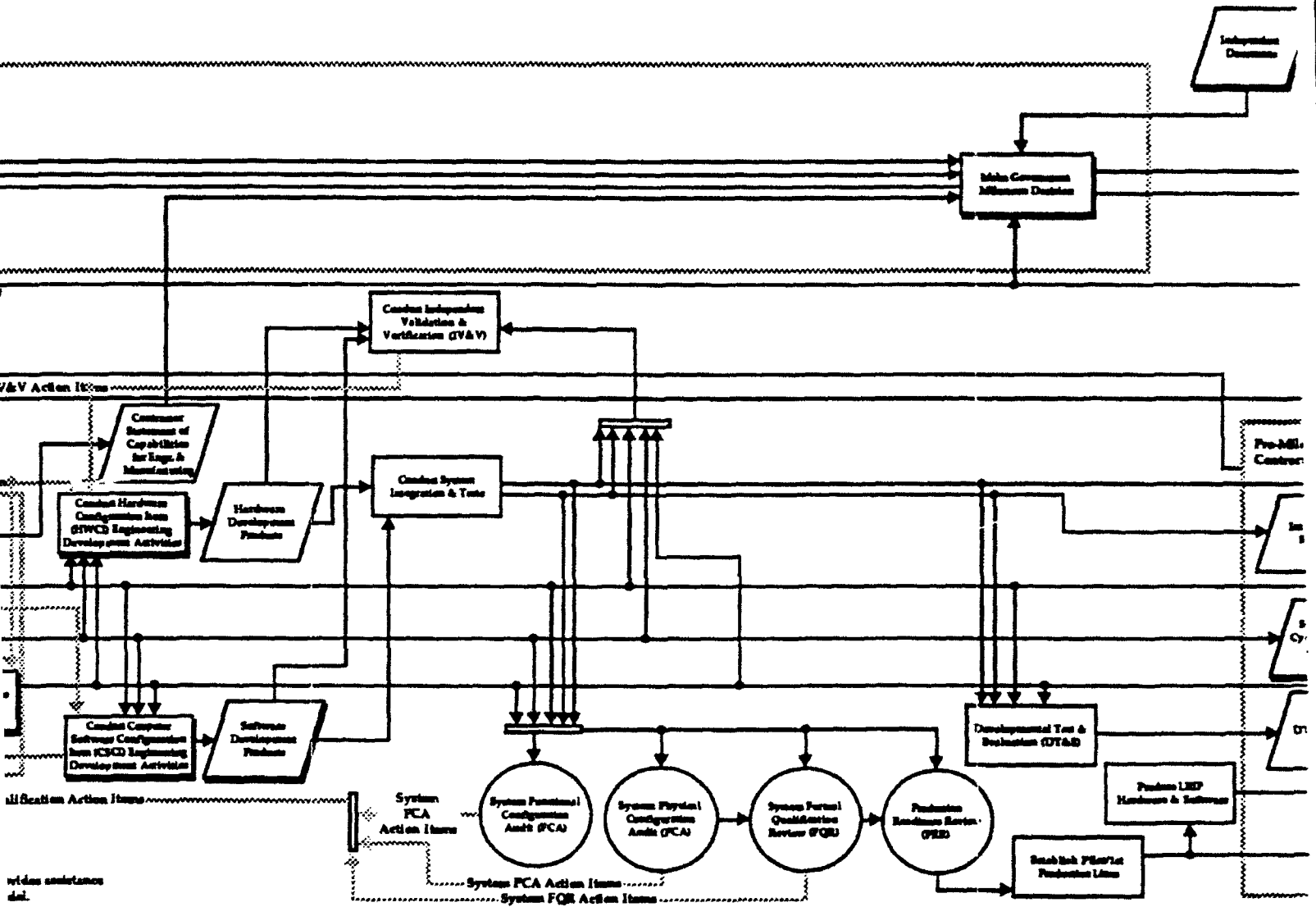
Figure 3.2.4-1 Phase II Engineering An



Double Click in the shadowed portion of the appropriate symbol to obtain the abstracted information at the next level of detailed. The help button provides assistance in the utilization of the symbols and flows. Specific information about the desired symbol can be obtain by addressing the appropriate mode of the model.

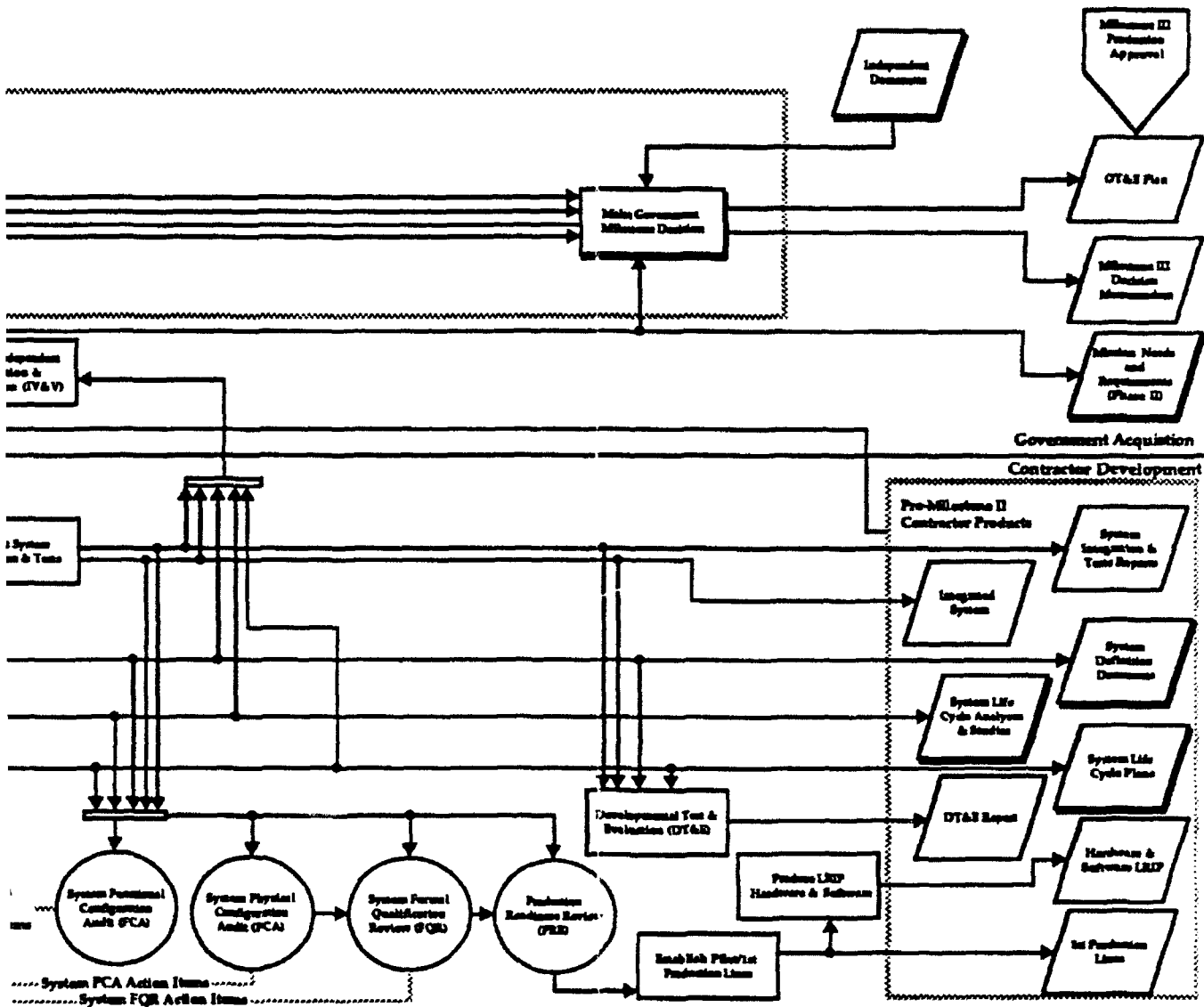
1083

# Phase II Engineering And Manufacturing Development



293

# uring Development



303

### 3.2.4.1 Hardware, Software And The System Engineering Process

In the Engineering and Manufacturing Development Phase, the system engineering process finalizes the system design definition by integrating the results of the hardware and software development processes with the system integration tests. We assumed that hardware /software development and the system integration/testing were specialty engineering disciplines and not the thrust of system engineering.

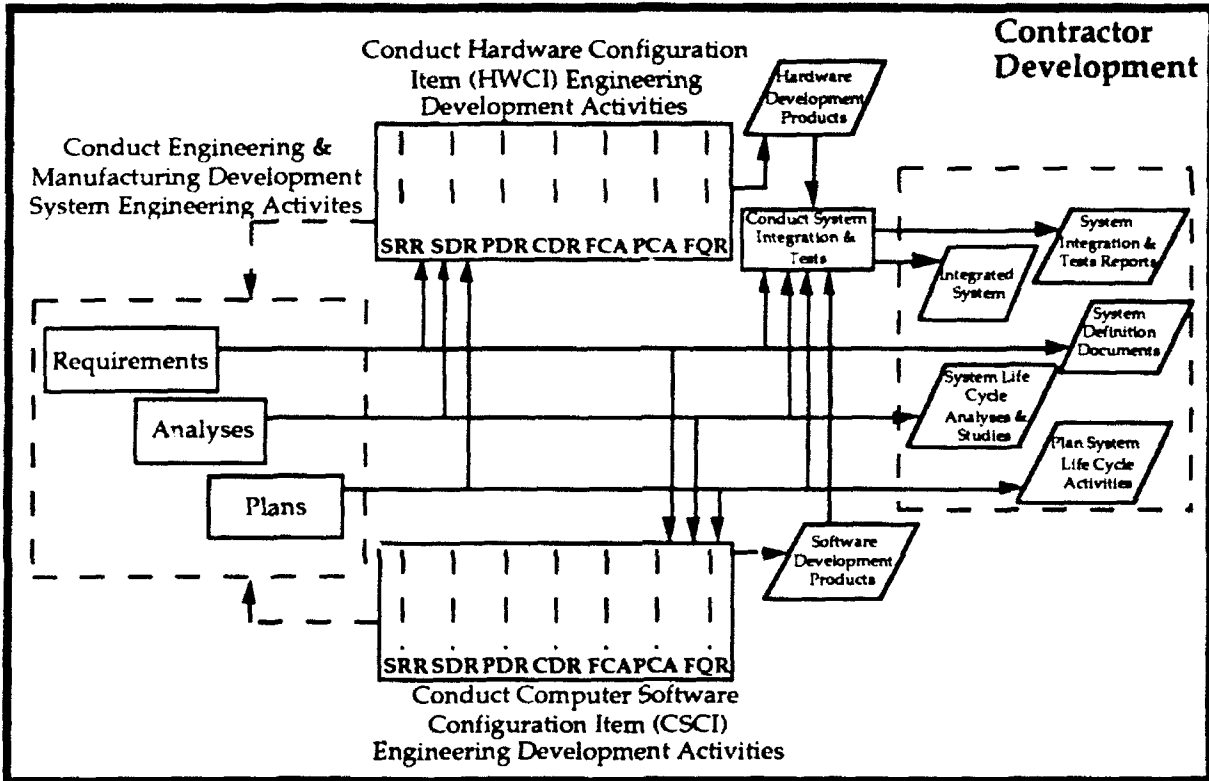


Figure 3.2.4.1.-1. Hardware, Software, and System Engineering Process

Figure 3.2.4.1-1 shows how the system engineering process is fed backwards by the hardware and software development processes. Since the output of the system engineering process defines the system, each formal hardware and software development review inputs a greater level of detail to the system engineering process. The reviews ensure consistency and refine the system definition work products. One worthy note is the possibility of using three different development approaches (i.e., spiral, incremental, concurrent) for the hardware, software, and system engineering development. The Integration and

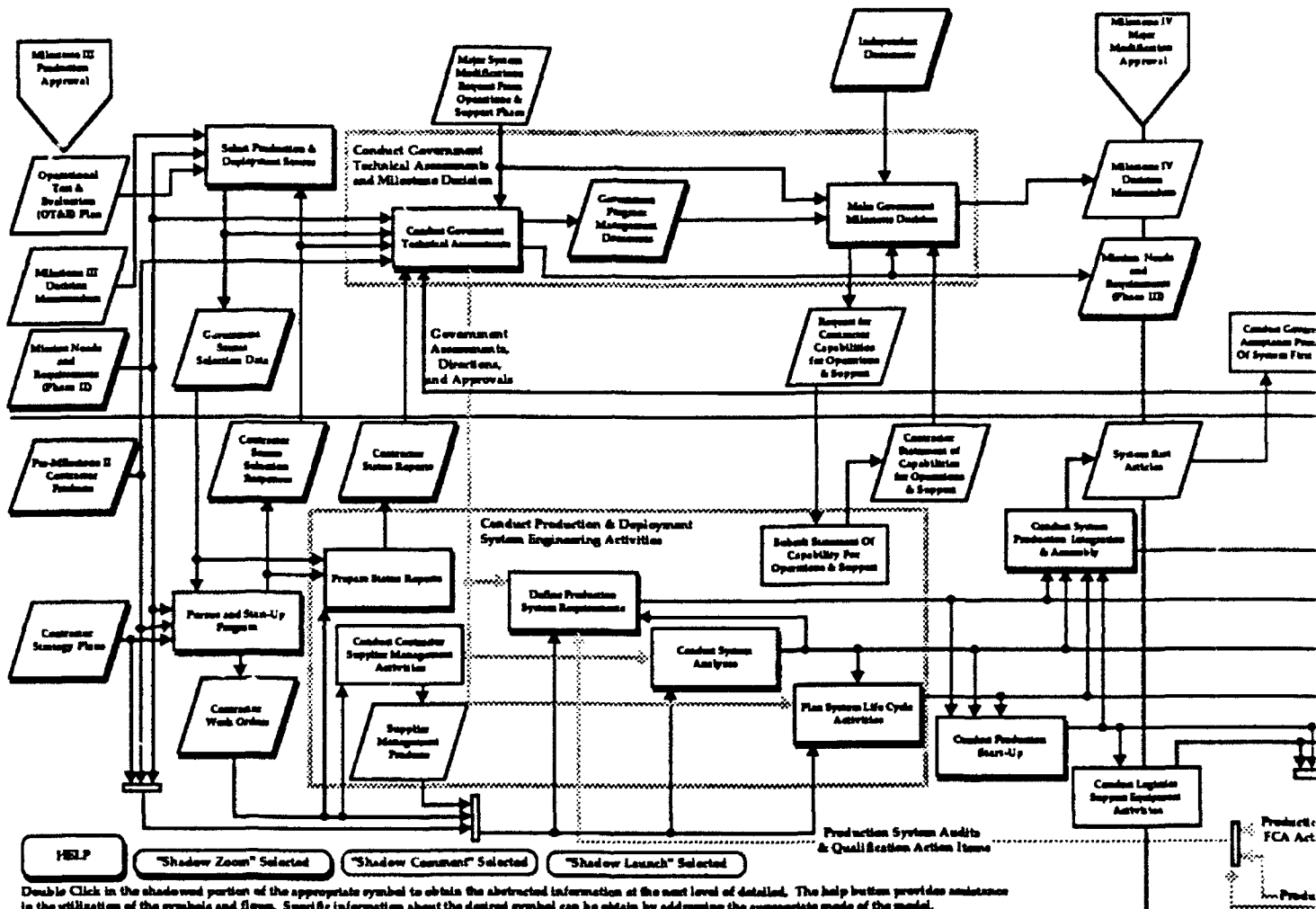
Test process puts these three development areas together. In real life, a key to a deliverable system is that Integration and Test specialists uncover numerous design, performance, and compatibility problems.

### **3.2.5 Production And Deployment Phase**

In the Production and Deployment Phase, the government and contractor establish a production line with sufficient verification and support testing to enable deployment. The production run could last 5 to 10 years depending on the system. During this period of time, engineering changes are made to suit operational needs or conform to original functional requirements. Typically, full functionality is achieved incrementally during production because of integration and testing constraints and schedule challenges. Since operational design changes can be incorporated during the Production And Deployment Phase, it is important to understand that Functional Configuration Audit, Physical Configuration Audit, and Formal Qualification Reviews are held to insure proper and controlled configuration of the system.

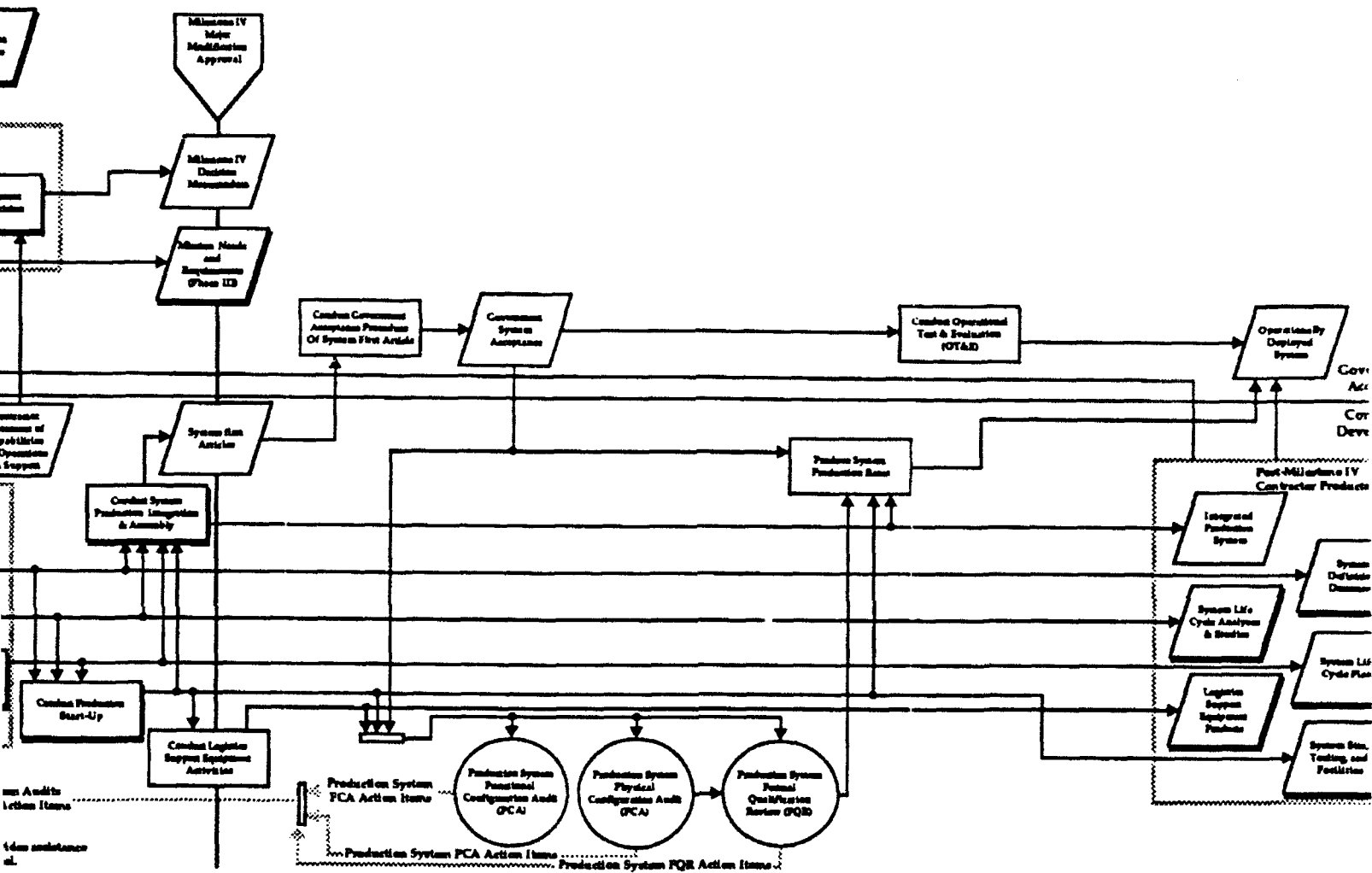
During this phase, major modifications to the system are approved and incorporated into the system production line. The emphasis of this phase is on the production configuration of the system and the adaptation of major modifications. Figure 3.2.5-1 illustrates the third level of detail of the Production and Deployment Phase.

Figure 3.2.5-1 Phase III Production & Deployment



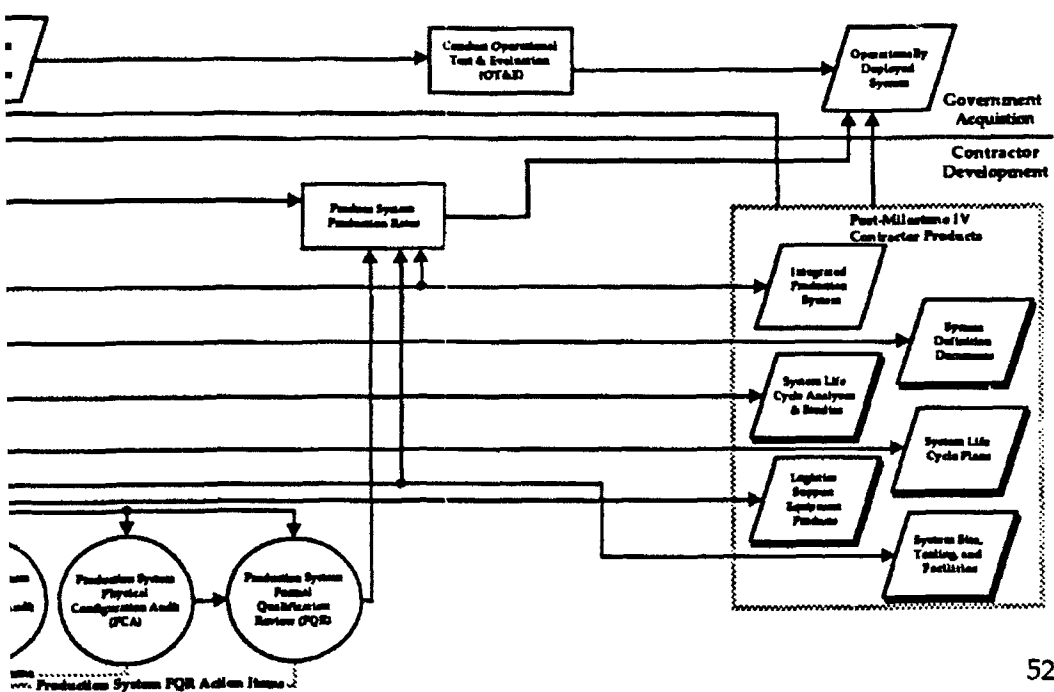
Double Click in the shadowed portion of the appropriate symbol to obtain the abstracted information at the next level of detailed. The help button provides assistance in the utilization of the symbols and flows. Specific information about the desired symbol can be obtain by addressing the appropriate mode of the model.

10/3



2013



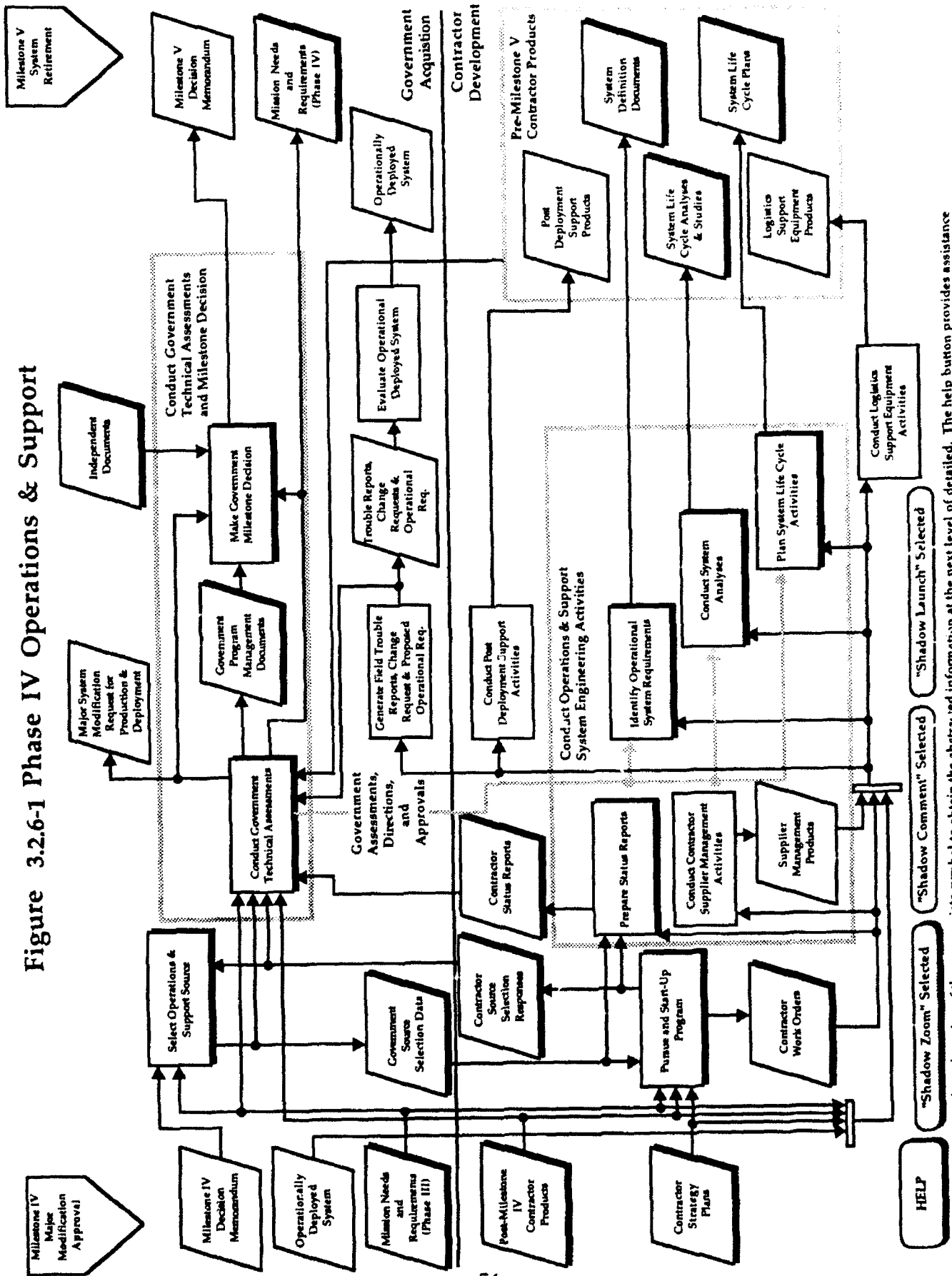


3073

### **3.2.6 Operations And Support Phase**

In the Operations and Support Phase, the government and contractor maintain the deployed system. To complete this service, a program is established which supports testing and provides spare parts. Another function of this phase is to address the field trouble reports. This report analyzes and provides the basis for modifications to the production system, as well as future field needs for a system. The system engineering process in this phase is focused on the operability and effectiveness of the system. If the system is not sufficiently efficient, the system must be retired and a new set of Statement of Needs (SON) and Mission Needs Documents must be developed. Notice how the field trouble reports are fed into the Production and Deployment Phase and the SON into the Determination of Mission Needs Phase. Figure 3.2.6-1 illustrates the third level of detail of the Engineering and Manufacturing Development Phase.

Figure 3.2.6-1 Phase IV Operations & Support



Double Click in the shadowed portion of the symbol to obtain the abstract information at the next level of detail. The help button provides assistance in the utilization of the symbols and flows. Specific information about the desired symbol can be obtained by addressing the appropriate mode of the model.

### 3.2.6.1 Production & Deployment And Operation & Support Phases

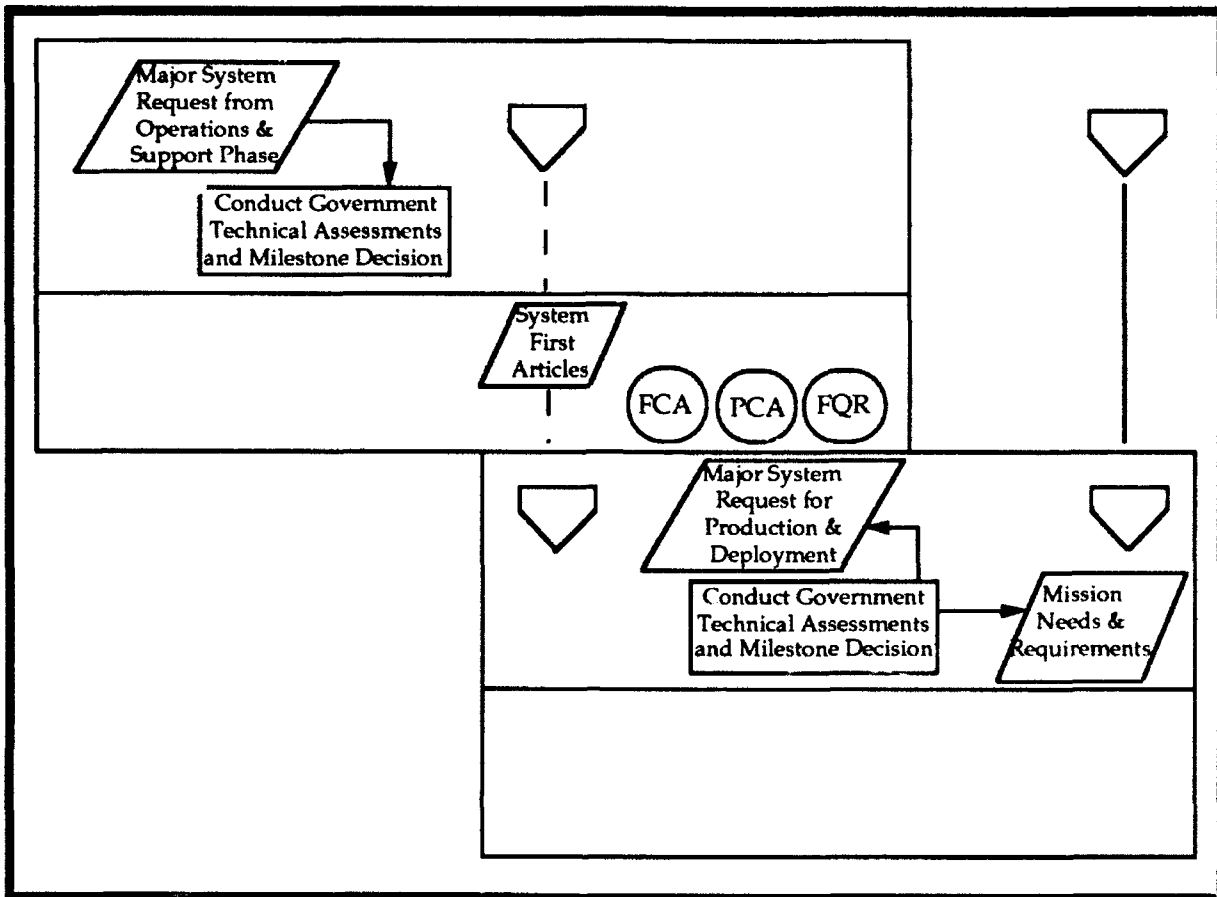


Figure 3.2.6.1.-1. Process Model Production & Deployment and Operation & Support Phases

Figure 3.2.6.1-1. illustrates the Production & Deployment and the Operations & Support phases. The milestone connection between the previously mentioned phases is "Major Modification Approval." As we enter the Production & Deployment phase, the Government must assess the requests it receives from the field to establish operational effectiveness. Figure 3.2.6.1.-1. illustrates this point by showing an output from the Government assessment process in the Operations & Support Phase and the Production and Deployment Phase. Notice the system's first article output is aligned with the "Major Modification Approval" milestone. Therefore, it is possible to deploy a system straight from the production line; in which case, field operations and support must provide feedback for any needed modification of the system. As the production line continues, changes to the system can be implemented. The Process Model

accounts for this possibility in its representation. Any changes to the system must be formally qualified, hence the reason for the FCA, PCA, and FQR reviews at the later part of the Production & Deployment phase. The Mission Needs & Requirements output contains the Statement Of Need (SON) used at the beginning of the system life cycle. The last milestone (left to right) in Figure 3.2.6.1-1. is "*System Retirement*" at which point the SON identifies the need for a suitable replacement.

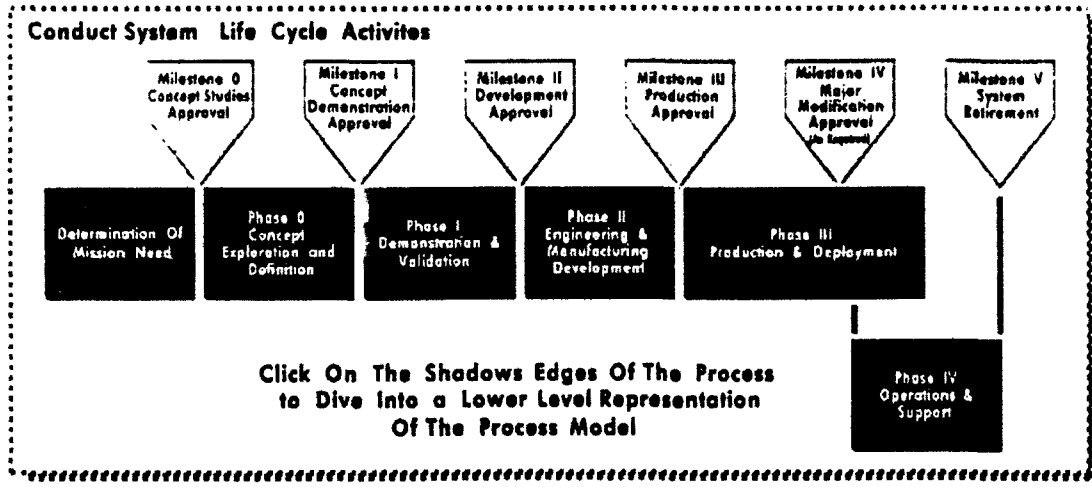
### **3.3 Abstracted View**

Abstraction is the most difficult part of modeling because it requires a full understanding of the process and its interactions within the system life cycle. It is not a mere input-process-output representation. It requires visualization, imagination, and experience in the process. Abstraction is a subjective discipline and skill. The modeler requires a clear and precise understanding of the representation uniformity and concept theme. The message being communicated is essential in order to effectively abstract the process to the desired level. Abstraction remains the most critical part of any representation whether the form of modeling is mathematical, graphical, textual or another type. The Process Model representation emphasizes information; the aforementioned steps are based on that emphasis.

During the SECD National Council On System Engineering (NCOSE) presentation session, the upper three levels of abstraction were introduced in a 5' by 8' color wall chart. In some areas, the Process Model contains up to eight levels of abstraction. In this report, abstraction refers to the hierarchical representation of a process. To illustrate the concept of abstraction, Figures 3.3a, 3.3b, and 3.3c represent the upper two levels of the Process Model. (These figures have been split into two foldouts because they are not readable when placed on one foldout.)

We followed a set of abstraction guidelines in the development of the Process Model. These guidelines include specific level of uniformity throughout the model, visibility to major formal reviews, and reasonable process functional flow and work products. The Process Model integrates documented (derived from standards) and "*As-Is*" (derived from field interviews) activities, processes, and work products by using a set consistency rules which produce a functional flow.

# System Acquisition and Development Process with Emphasis on System Engineering



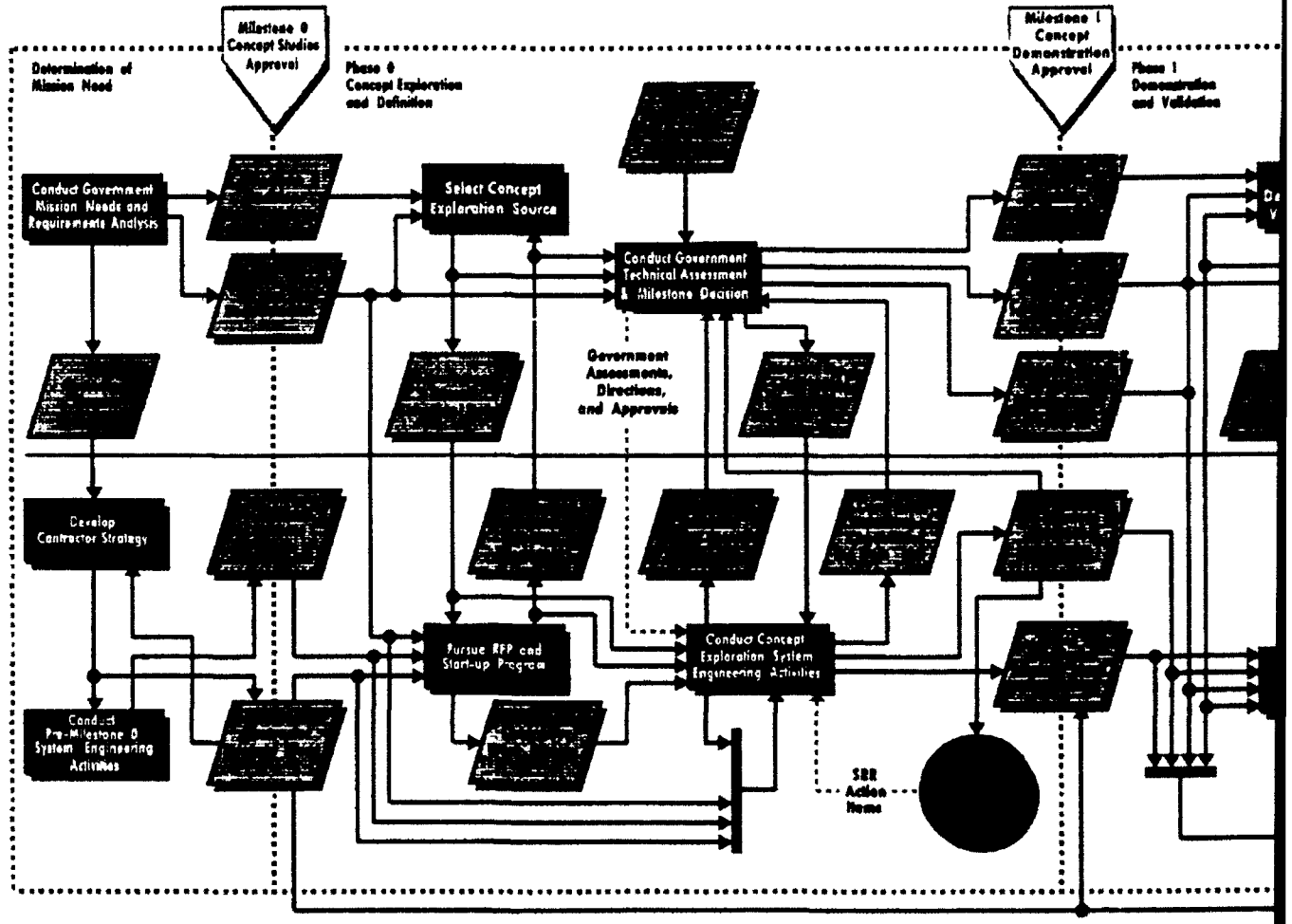
**Process Model Precedence**

- DoD 5000.2, "DoD System Acquisition Process"
- MIL-STD-15218, "System Engineering Handbook"
- MIL-STD-973, "Common System Engineering Handbook"
- MIL-STD-499A, "System Engineering Handbook"
- MIL-STD-499B, "System Engineering Handbook"
- DoD-STD-2167A, "System Engineering Handbook"
- AFR 800-14, "Life Cycle Management"
- S & E INST 5451.2

**HELP** "Shadow Zoom" Selected

Double Click in the shadowed part in the utilization of symbols and flow

# System Acquisition and Development Top Level with Emphasis on System Engineering



**HELP** "Shadow Zoom" Selected "Shadow Comment" Selected "Shadow Launch" Selected

Double Click in the shadowed part in the utilization of symbols and flow

1062

Development Process Model  
Engineering Activities

**Figure 3.3a System Engineering Acquisition And Development  
Upper Two Level Process Model With Emphasis On  
System Engineering Activities**

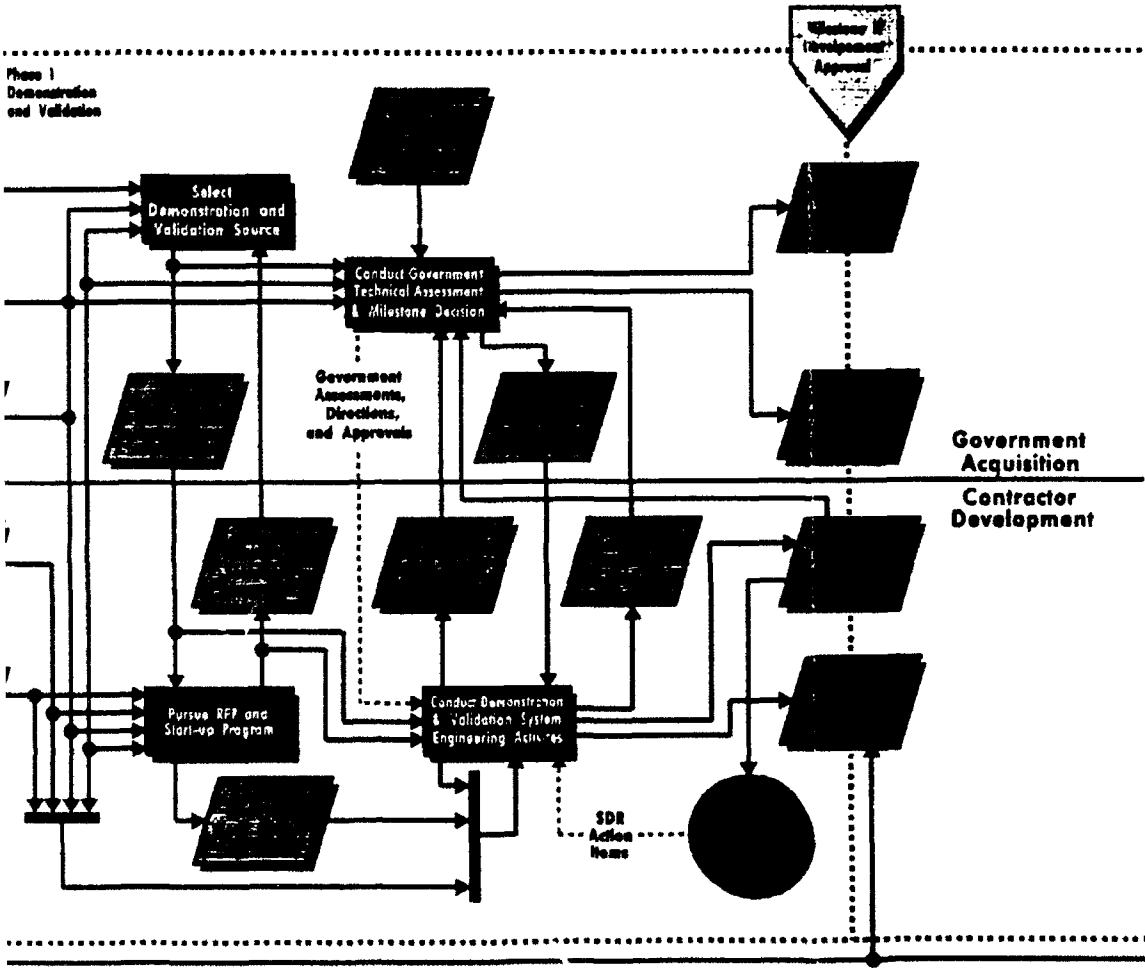
Model Precedence List Of Standards:

- 15000.2, " Defense Acquisition Management Policies and Procedures ", June 4, 1991
- STD-1521B, " Technical Reviews and Audits for Systems, Equipment, and Computer Software ", June 4, 1991
- STD-973, " Configuration Management ", Undated Draft
- STD-499A, " Engineering Management " May 1, 1974
- STD-499B, " System Engineering ", Undated Draft
- STD-2167A, " Defense System Software Development ", February 29, 1988
- 300-14, " Lifecycle Management Of Computer Resources In Systems ", September 29, 1986
- E INST 5451.2, " System Engineering Technical Support Implementation ", September 21, 1987

Shadow Zoom Selected  
 Shadow Comment Selected  
 Shadow Launch Selected

Click in the shadowed portion of the appropriate symbol to obtain the abstracted information at the next level. The help button provides assistance in the shadowed portion of the appropriate symbol to obtain the abstracted information at the next level. The help button provides assistance in the shadowed portion of the appropriate symbol to obtain the abstracted information at the next level. The help button provides assistance in the shadowed portion of the appropriate symbol to obtain the abstracted information at the next level.

Development Top Level Process Model  
Engineering Activities

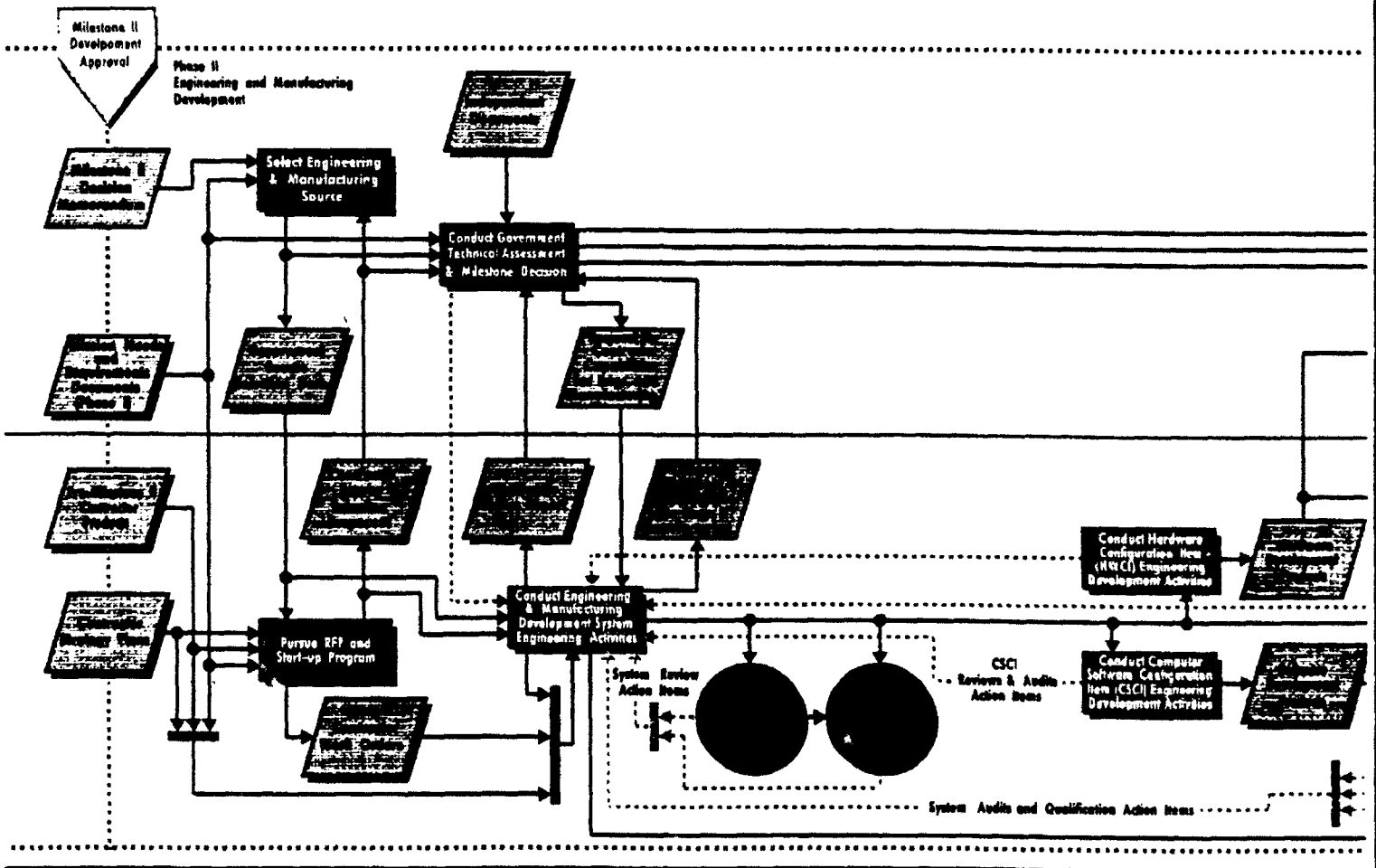


Click in the shadowed portion of the appropriate symbol to obtain the abstracted information at the next level. The help button provides assistance in the shadowed portion of the appropriate symbol to obtain the abstracted information at the next level. The help button provides assistance in the shadowed portion of the appropriate symbol to obtain the abstracted information at the next level.

*2072*

**Figure 3.3b System Engineering Acquisition  
Upper Two Level Process Model  
System Engineering Activities**

**System Acquisition and Development Top Level  
with Emphasis on System Engineering**

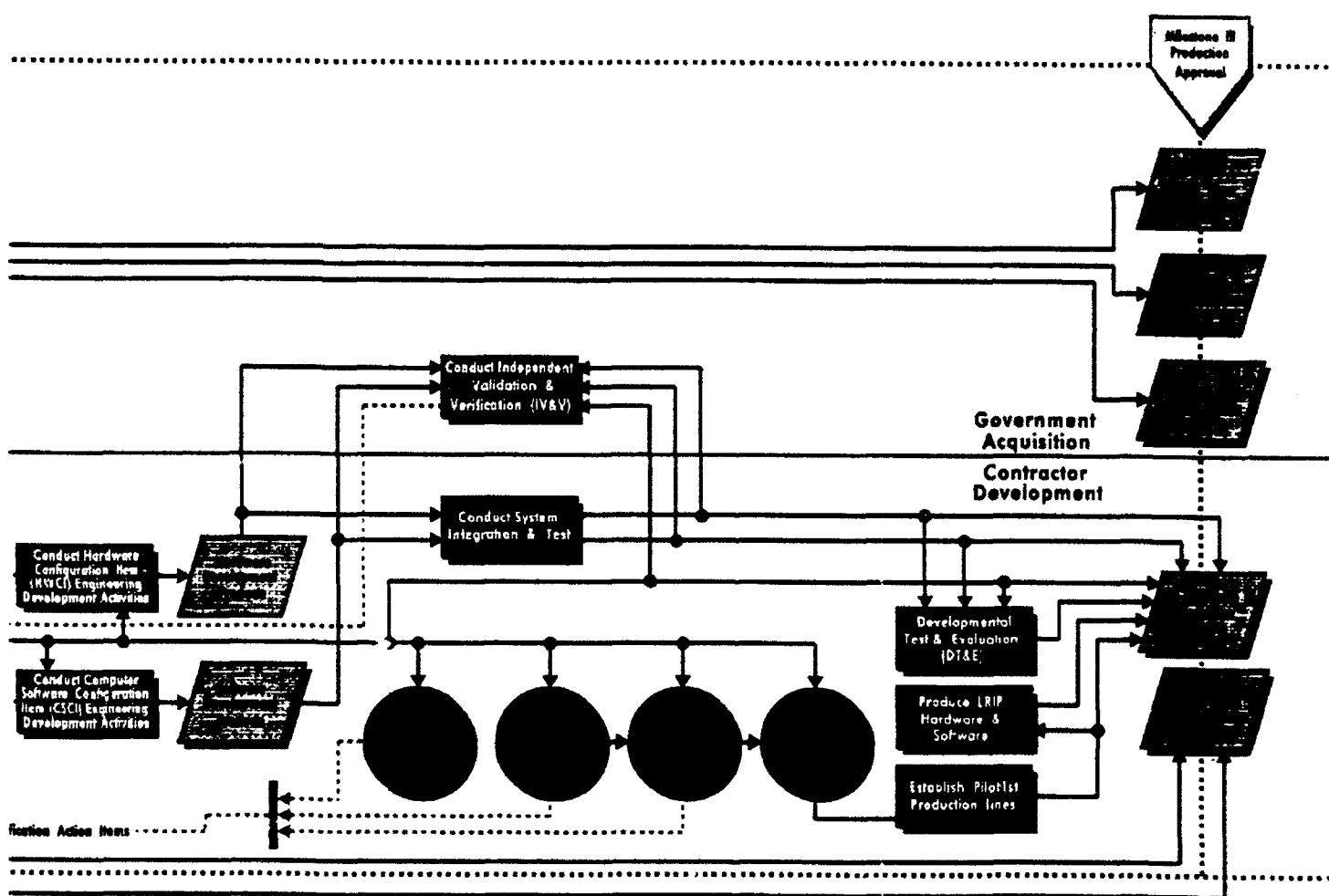


1072



**Engineering Acquisition And Development  
 Top Level Process Model With Emphasis On  
 Engineering Activities**

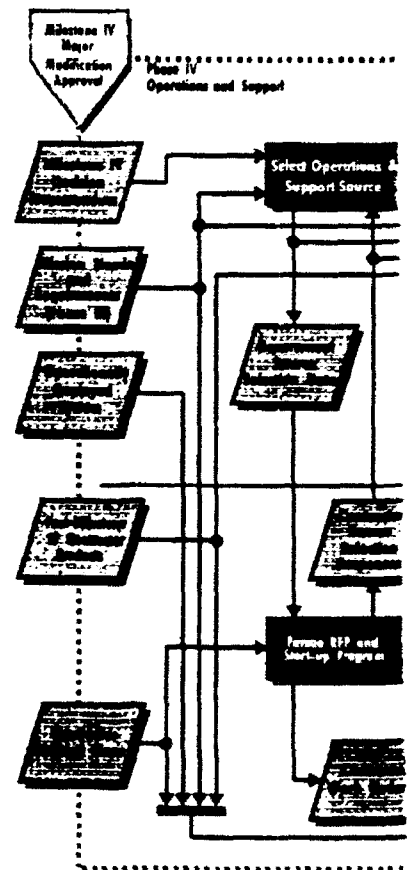
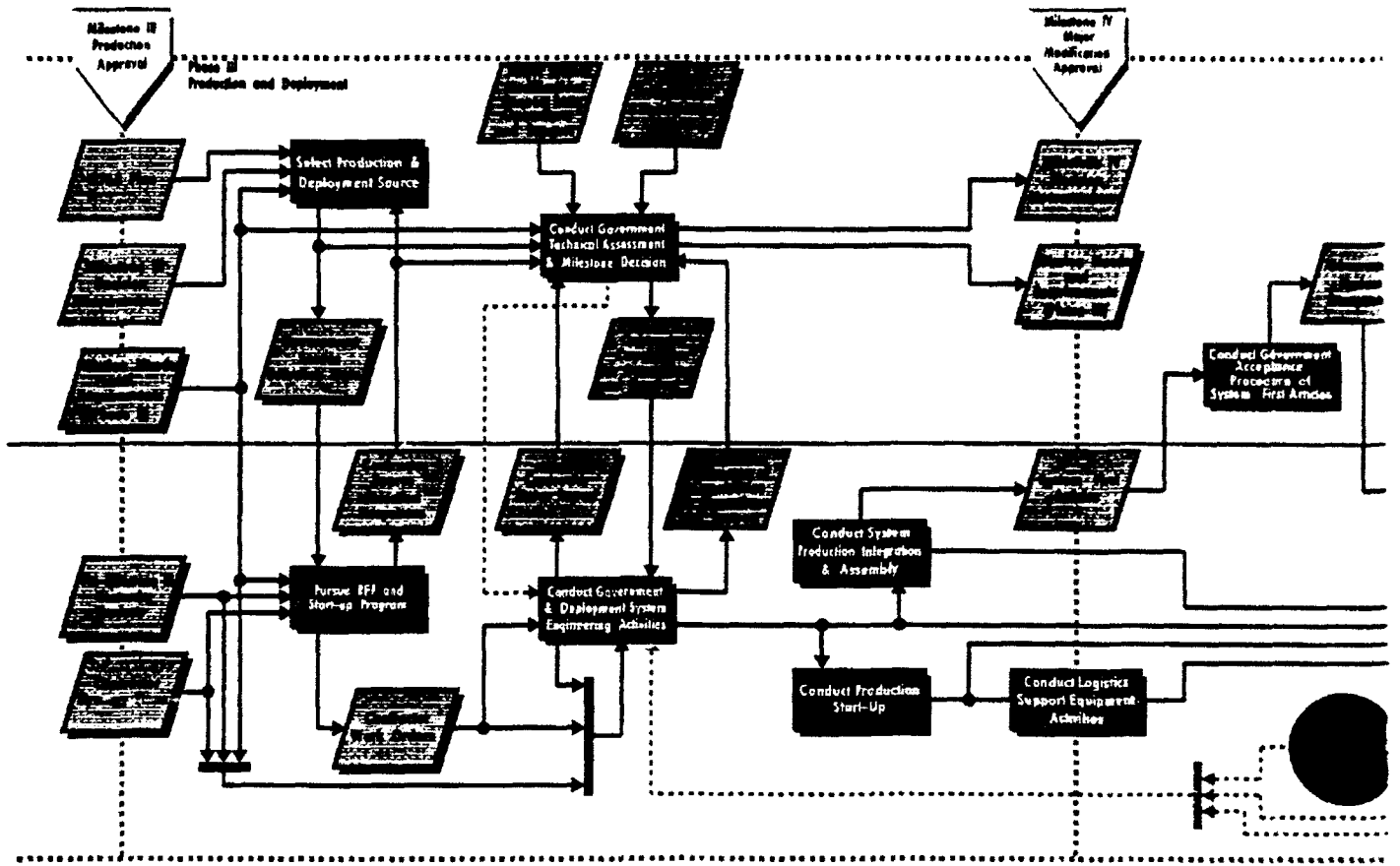
**Development Top Level Process Model  
 System Engineering Activities**



Double Click in the shadowed portion of the appropriate symbol to obtain the abstracted information of the next level. The help button provides assistance in the utilization of symbols and flows. Specific information about the desired symbol can be obtained by addressing the appropriate mode of the model.

2 of 2

# System Acquisition and Development Top Level Process with Emphasis on System Engineering Activities



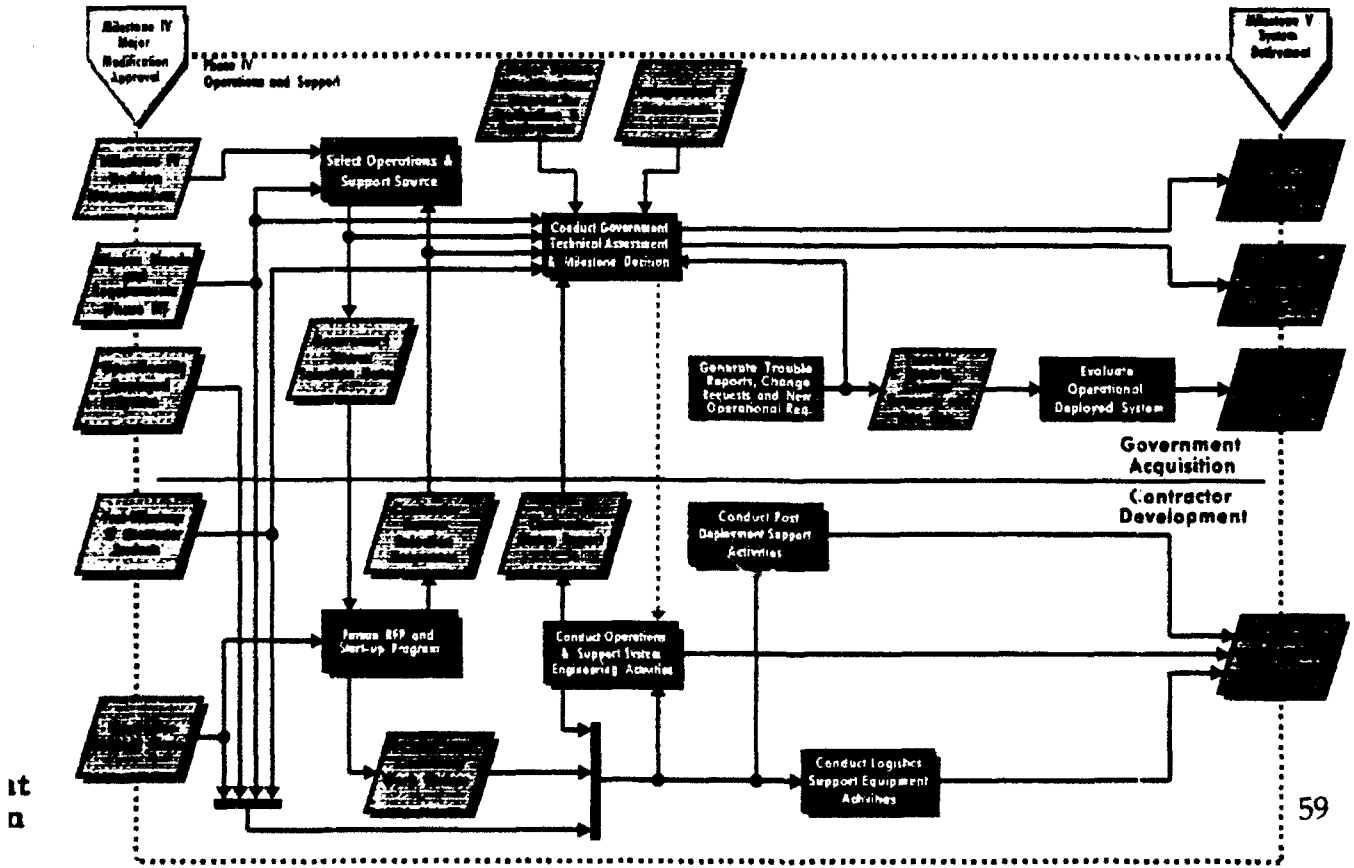
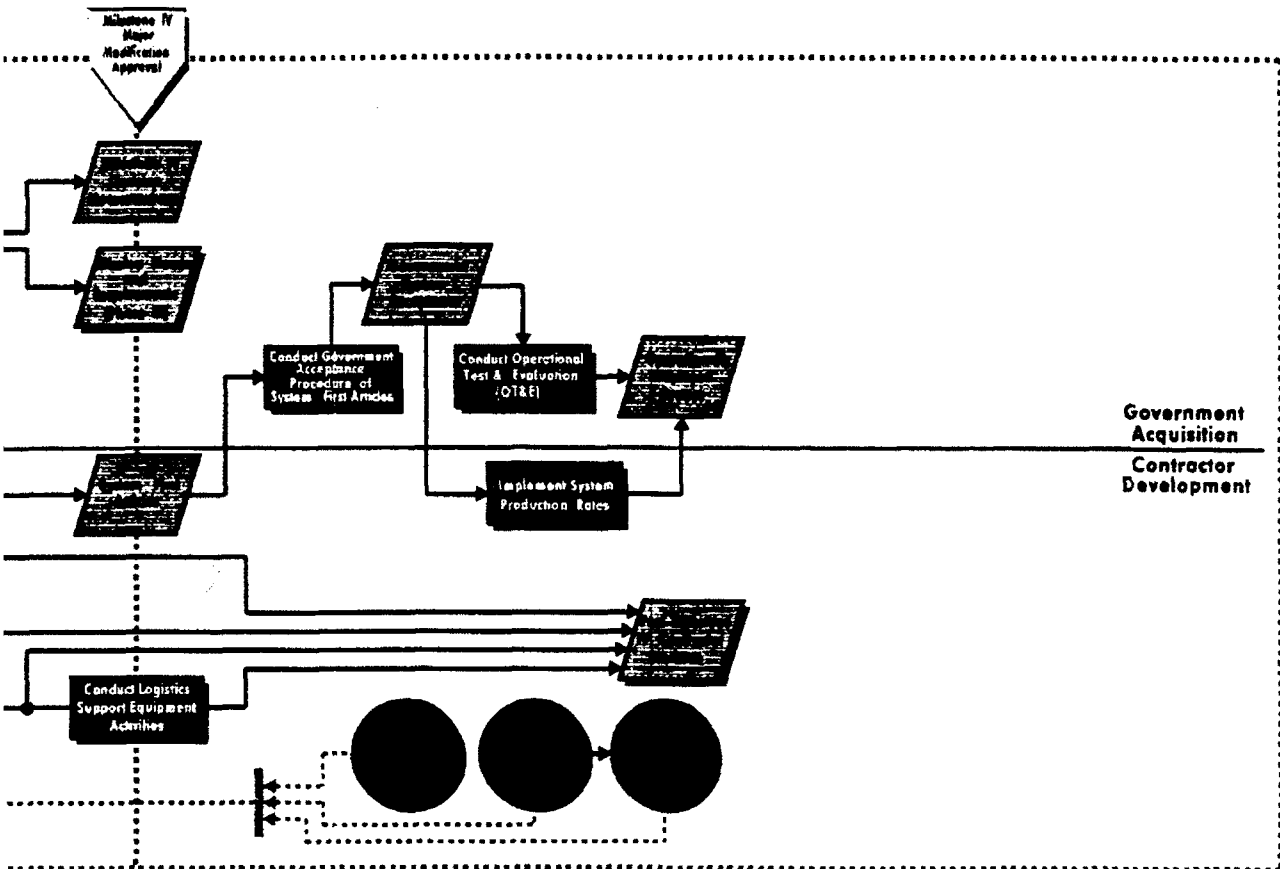
HELP

Double Click in the shaded portion of the appropriate symbol to obtain the abstracted information at the next level. The help button provides assistance in the utilization of symbols and flows. Specific information about the desired symbol can be obtained by addressing the appropriate menu of the model.

**Figure 3.3c System Engineering Acquisition And Development Upper Two Level Process Model With Emphasis On System Engineering Activities**

1062

# Development Top Level Process Model in System Engineering Activities



It  
a

*2072*

Uniformity was achieved throughout the model by counting the number of symbols represented at each level. In areas where readability or understanding could be compromised, additional symbols were added or deleted. Uniformity was an acceptability criteria factor as well as an abstraction guideline.

Another abstraction guideline is the visibility of major formal reviews. It has been discovered that the majority of people understand the system lifecycle process by following major milestones and formal reviews. This particular abstraction was very difficult because the Process Model represents interaction between Government and Contractor processes. These processes are implemented at a higher level than formal reviews. A judgement was made in each case as to whether the process should be included.

Two more abstraction guidelines were logical functional flow and work products. A series of functional flow revisions were made in order to implement this abstraction guideline. Practitioner process recognition and acceptability were targets of this guideline. Completeness of work products was inspected against standards and flows. Before the Process Model was baselined, many reviews were conducted.

While studying Figures 3.3a, 3.3b, and 3.3c, one should refer back to Section One, representation methodology, for a brief explanation of the Process Model symbols, colors and shapes.

### **3.4 Discriminating Factors**

Few models of the System Engineering process have been completed over the system lifecycle. Most of the previous work isolated the system engineering process from its natural domain. As a result, many critical interactions and flows between activities, process, and work products were lost; hence, the usefulness of the model was diminished. Other process modeling work focused solely on a particular process in a phase of the system life cycle. A particular solution was advocated for representation and abstraction and the result was communication of the captured information was lost. In other cases, the work was so generic that no information is conveyed about the process or the products. Although process modeling is an emerging technology, its emphasis must shift away from semantics to communications.

The SECD Process Model has a number of discriminating factors. The most prevailing is its emphasis in communicating information about the process. This factor is accomplished by providing a highly readable and understandable set of semantics. The following is a list of the Process Model discriminating factors:

- a) Emphasis in communicating information about the process.

- b) Interactions between Government Acquisition and Contractor Development captured.
- c) Eight hierarchical levels of detail represented.
- d) Interactive multi-dimensional view of the processes represented.
- e) System life cycle perspectives are modeled, including the system engineering process.
- f) *Generic representation selected*
- g) Integration of various system engineering standards into a cohesive and understandable view accomplished.

Although a detailed and definitive start, the SECD Process Model should not be considered the final answer. The objective of any model is to streamline the process. The SECD Process Model is not an exception to this rule. As we discern our work from others, we would like to encourage further experimentation with the Process Model. Many of the aforementioned discriminating factors separate our works from others. We believe the most discriminating factor is that the SECD Process Model provides essential information about process interaction.

**This page intentionally left left blank.**

## **4.0 Conclusions**

Building the SECD Process Model has broken new ground in our understanding of the System Engineering process. An interactive model of the activities of the lifecycle, the Process Model is extremely comprehensive, flexible, and extensive. It was designed to support the degrees of freedom needed to accommodate currently identified and future life cycles standards. Its strength lies in the identified interfaces between government acquisition and contractor development and the interaction between the System Engineering process and the system life cycle activities.

### **4.1 System Engineering**

One of the biggest issues we addressed during this effort was the definition of the term 'System Engineering'. The following list provides the definitions we identified for this term; they are not listed in any order of priority or importance:

- 1) System Engineering is both a technical and management process used in the solution of a problem or development of a system.
- 2) System Engineering comprises the entire life cycle of the product and entails cost, schedule and technical performance of the system.
- 3) System Engineering is a qualitative and quantitative mechanism for program managers in the decision making process. It is a team approach to problem solving.
- 4) Specialty engineering disciplines are selected according to the size, type, performance and level of System Engineering required.
- 5) Emphasis on technical and management aspects are dependent upon a particular organization and product.
- 6) The System Engineering role can be divided into three categories of activities: engineering, management, and communications.
- 7) System engineers develop models that are represented in various forms such as textual, graphics, scripts, and other notation languages.
- 8) System Engineering includes Concurrent Engineering (CE), Reverse-Engineering, Re-Engineering, Requirements Engineering, Integrated Process Development (IPD), Design To Life Cycle Costs (DTLCC), Technical Program Management, System Architecture, System Analysis, and Commercial Systems. System Engineering does not include hardware or software development or system integration. These particular disciplines are considered specialties within the Process Model,

and they have a big impact in the System Engineering process and its work products.

9) System Engineering is a problem solving technique that is applied at all levels of system development, acquisition, and management.

## **4.2 System Engineering Process**

After composing a solid definition for System Engineering, the next task was to ascertain the variability of the System Engineering process. The following is a list of observations about the System Engineering process.

- 1) The System Engineering process is contained within the system development process which in turn is contained within the system acquisition process.
- 2) The System Engineering process cannot be separated from the system development process. The interfaces between system acquisition and system development must be identified so the System Engineering process is meaningful.
- 3) The System Engineering process varies from one organization to another. Its variation encompasses the entire system life cycle horizontally and vertically.
- 4) The System Engineering process can be divided into three areas: requirements, analysis, and planning.
- 5) Automation of the System Engineering process can be achieved through flexibility and adaptability.
- 6) System Engineering processes and methods are derived from accepted industry standards and include best practices for implementation. The Process Model is the means by which the users are empowered to tailor and improve system processes and methods.

## **4.3 Process Model**

After defining the term 'System Engineering' and studying the System Engineering process, we were ready to address the issue of representation and the Process Model. The following is a list of observations about the Process Model.

- 1) Processes are multi-dimensional in nature; they have a relationship to a larger framework.
- 2) The Process Model must be generic in order to support the variation and abstraction needs of the Process Model.



- 3) The order of the symbology had an impact on the Process Model's final perspective.
- 4) The Process Model emphasizes the encapsulated information, not the representation. The acceptability criteria ensures this emphasis is maintained throughout the representation.
- 5) The Process Model can be constructed to incorporate system metrics.
- 6) The Process Model by virtue of its extensibility and completeness can be used to customize systems engineering processes unique to each individual development approach.
- 7) The Process Model can be used as a Reference Model to map System Engineering methods and assess their completeness and effectiveness in the development of a system.

The Process Model developed for SECD demonstrated coverage of the System Engineering process in *Catalyst* requirements and completeness of the System Segment/Specification (SSS) document. It supported the development of the prototype demonstration scenarios by providing a guideline to the System Engineering process.

#### **4.4 Lessons Learned**

Process modeling is a complex and mentally intensive task. It is not a task that can be completed by one person. The weaving together of numerous elements involved in process modeling requires a cohesive team of people with domain knowledge.

The following lessons learned provides general insight into the challenges of process modeling.

- 1) Concrete goals, objectives and a detailed plan of action (that is consistent with the customer needs) are necessary to complete the process modeling task.
- 2) The main focus of the development team should be on information gathering, representation methods, automated tool selection, and desired level of abstraction.
- 3) Abstraction and scoping are the key issues of process modeling. These issues influence the representation form and the depth of detail required. These issues should be continuously revised to ensure the proper direction is being followed throughout development.

- 4) Various representation approaches should be examined. This activity actually reduces risk and will help the developers anticipate trouble areas and construct their solution..
- 5) Develop or chose the representation method and automated tool early in the process.
- 6) Set the domain boundaries to the problem and make assumptions based on gathered information.
- 7) Reconcile conflicting information by setting a precedence list to avoid ambiguity.
- 8) The objective of any process model is to streamline the process.
- 9) Processes, like standards and living documents, should adapt to new technologies, practices, businesses, and design constraints. To successfully complete the Process Model, set a measure of completeness at the very beginning.
- 10) Involve the customer in the review of the process model.

## **5.0 Future Applications And Directions**

This section outlines the future potential applications for the process model. Applications include the addition of metrics and instrumentation, the mapping of system methodologies into the model and the utilization of the model as a seamless interface for other works in the system engineering area. This section discusses using the model as a process knowledge engine and shows how additional levels of abstraction would produce generic system engineering procedures.

Potential areas of application for the *Catalyst* Process Model include training, tracking, tracing, planning, guiding and benchmarking system engineering methodologies.

The Process Model is envisioned as a spin off platform for applications and experiments in the areas of reference model, management, training, knowledge bases, and metrics. The following sections describe these potential applications.

### **5.1 Reference Model**

A need exists today for the means to identify, validate, and improve the system engineering life cycle process. This need has been expressed by government, industry and academia. The Process Model used in combination with other models represents, brought insight and discipline to processes, methodologies, and environments. Example models to be considered could be the Software Engineering Institute (SEI's) Capability Maturity Model and the National Institute of Standards & Technology (NIST) Reference Model for Frameworks of Software Engineering Environments recently released in January 1992.

### **5.2 Process Management and Guide**

The Process Model's features of breadth and detail allow it to be applied to an existing program. The model provides the foundation of identifying and managing the numerous activities in a particular phase of the life cycle. By helping plan and direct activities in a program, the Process Model is being used as a management tool.

Since the Process Model is based on applicable standards, it also becomes a graphical representation of various integrated textual standards. In the future, the Process Model could be defined visually to the procedure level. With insight to this level, the user could observe the development of the system parts as the system is being completed. Cost, schedule, and requirements state vectors would be integrated to provide customized statistical information about the

system and its process. "Catalyst" will be the basis for the framework of this application.

### **5.3 Training Tool**

The Process Model has the potential to become the first self-taught, system engineering tool. We envision using a multi-functional, touch and pen screen, color display with multi-media capabilities. This display would provide sample system engineering tutorials to students with different levels of experience. Integrated processes and work products could be used to test and ascertain changes to the system development process. Because the Process Model already works interactively, this area promises to be very successful.

### **5.4 Process Knowledgeable**

The Process Model is envisioned to provide the basis for the development of an enactable process knowledge-based rule paradigm. This paradigm will support varied levels of system engineering function. It would also provide valuable information to Catalyst which currently requires more knowledge of the system engineering process.

To enact process knowledgeable rules, we foresee a parametric engine of the system engineering process throughout the life cycle. This engine will be based on a state vector approach. Experimentation, however, should be conducted in this area to determine the feasibility of the idea.

### **5.5 Metrics And Instrumentation**

Metrics and instrumentation is a highly promising application area of the Process Model. As discussed earlier, it is possible to instrument the Process Model with predictive metrics and to measure these parameters in a mathematical form. To apply these metrics effectively, the Process Model needs to be abstracted downwards to develop a set of generic system engineering procedures.

The remainder of this section discusses the instrumentation and application of metrics to the Process Model. The text and corresponding figures were provided by Mr. Mike Carrio, MTM Engineering Inc., McLean VA.

The Process Model enables the collection of predictive metrics in a cohesive and consistent manner. Predictive metrics, unlike their traditional counterpart, post-facto metrics (e.g., complexity metrics), can now be employed effectively in the early development and acquisition phases to provide risk mitigation strategies and approaches. Figure 5.5-1. illustrates the predictive metrics over the life cycle

phases. In the past, the lack of viable process models has precluded early life cycle instrumentation, thus, employing an extensive set of predictive metrics.

Additionally, the Process Model provides a contextual framework for the association and mapping of development and design methodologies. Use of formal and defacto methodologies can now be mapped to corresponding processes and work products to assess their breadth and completeness. This mapping in turn provides the basis for quantification of product, usage, and process metrics. Furthermore, relationships between process and product metrics can be algorithmically established to provide greater synergism with the predictive metric domain.

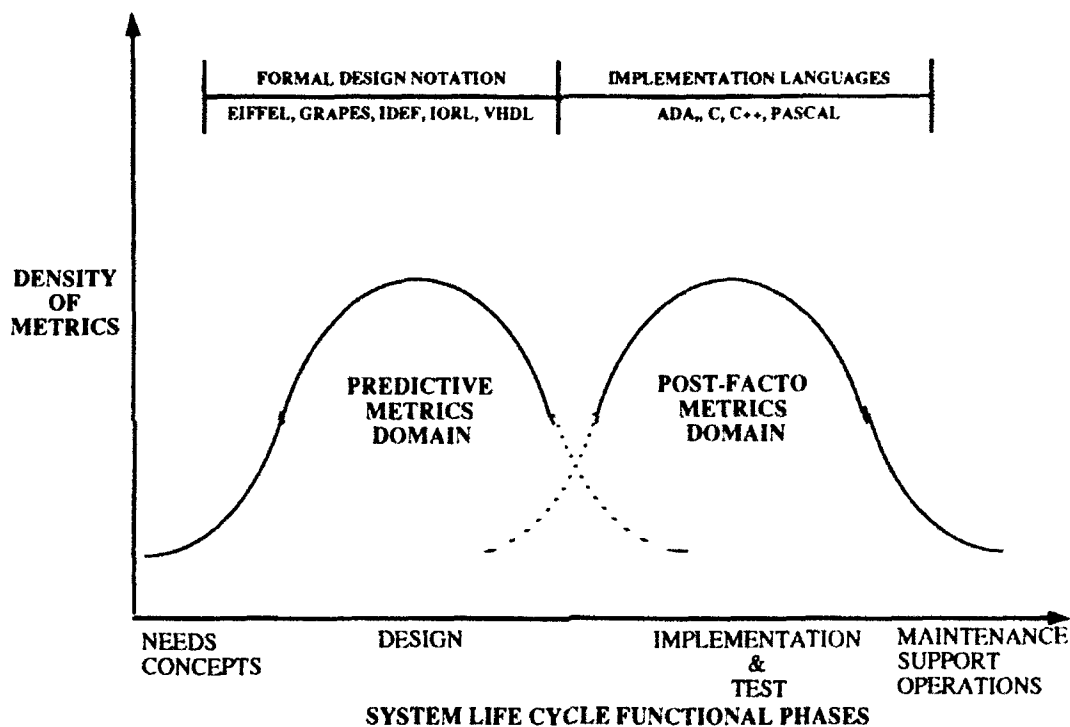


Figure 5.5-1. Density Of Metrics Versus System Life Cycle Functional Phases

The Process Model, with its capacity for instrumentation, can be utilized to effectively establish requirement state vectors with the inherent ability to determine requirements and process maturity, stability and completeness. When used, formal methodologies (i.e., methodologies supported by a formal grammar or executable notation) can be closely aligned to those processes or activities they support. The formal notation can then be utilized as an

instrumentation vehicle to collect particular measured information. In turn, the measured information can be incorporated into metrics for subsequent quantification. This procedure is identical to using a software implementation language as the vehicle for collecting metrics. Thus, metrics collection can be extended from the software implementation process into the systems engineering process and beyond. Quantification has allowed for cost effective implementation trade-offs much earlier in the acquisition/development phases than previously.

One of the barriers to predictive metrics and early risk mitigation strategies has been the inability to instrument and collect this type of information due to its informal, undisciplined and undefined nature. Requirements state vectors consists of cost, schedule, complex algorithms and maturity and stability dimensions; hence, these vectors lend themselves to be easily depicted and simplified. These simplified representations can then be effective program management tools and be early warning alarms that provide management information about critical activities. Program managers and their staffs, who use the same underlying technical units of measurement as those of systems and software engineers, view the same issues as their team members. However, their frame of reference may be a cost and schedule view which does not require knowledge of the state vector coefficients or algorithmic components.

This concept is somewhat analogous to the Parnas' concept of information hiding where software package functionality can be separated from the details of package implementation in order to enable effective communication between components at different levels of abstraction. The latter concept is one of the maturing signs of software engineering as a science, providing synergism to the realm of system engineering.

The Road Map Components are intended to identify the many views and elements required for viable design synthesis. The Process Model, in addition to facilitating process and product quantification, enables extensive and comprehensive methodology and element mappings to be performed. The mappings in turn can be effectively used to compare methodologies and ascertain how they relate to each other. Figure 5.5-2. illustrates the road map components of a system methodology.

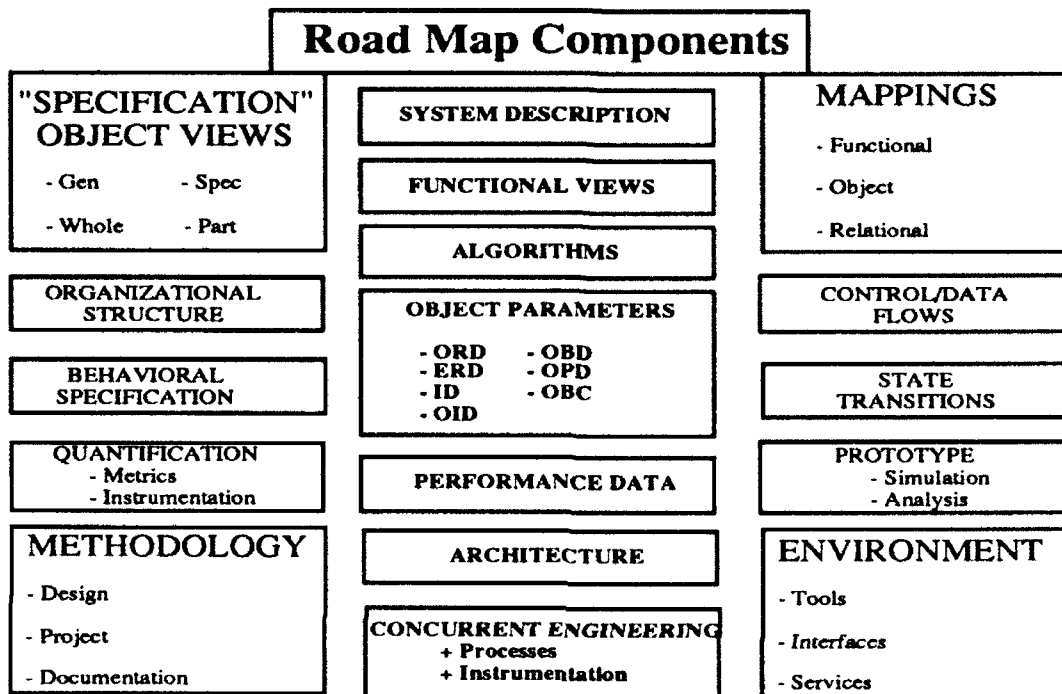


Figure 5.5-2. Road Map Components

## References

- [CHE65] Harold Chesnut, "Systems Engineering Tools." John Wiley & Sons, Inc., New York, 1965.
- [COM92] Comer, E.R., "Catalyst: Automating System Engineering in the 21st Century", Proceedings of the Second Annual International Symposium of the National Council On Systems Engineering (NCOSE), Seattle WA, 20-22 July 1992.
- [COR66] Robert E. Corrigan and Roger A. Kaufman. "Why System Engineering" published by Fearon Publishers, Inc. 2165 Park Boulevard, Palo Alto CA, 94306
- [HUM82] Watts S. Humphrey, "Japanese Management Perspectives for U.S. Engineering", Manufacturing Engineering, April 1982.
- [HUM89] Watts S. Humphrey, "Managing the Software Process", published by Addison-Wesley Publishing Company, Inc; October 1989. This book is part of the Software Engineering Institute (SEI) Series in Software Engineering.
- [KEL88] Marc I. Kellner and Gregory A. Hansen, "Software Process Modelling," Technical Report CMU/SEI-88-TR-9;ESD-TR-88-010, May 1988. Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213-3890
- [KEL89] Marc I Kellner, "Experience with Enactable Software Process Models" *Proceeding of the 5th International Software Process Workshop: Experience With Software Process Models*, October 10-13, 1989, Kennebunkport, Maine , USA. Published by ACM
- [KEL89\*] Marc I. Kellner, "Comparison of NAVAIR "Regulations" Process with F-14A "As-Implemented" Process," dated March 28, 1989. Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213-3890
- [KEL89\*\*] Marc I. Kellner, "Software Process Modeling Example," dated August 1989. *Proceeding of the 5th International Software Process Workshop: Experience With Software Process Models*, October 10 - 13, 1989, Kennebunkport, Maine, USA. Published by the ACM
- [KEL90] Marc I. Kellner, "Overview of the F-16 A/B Post Deployment Software Support Process Model," dated February 23, 1990. Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213-3890
- [KEL90\*] Marc I. Kellner, "Supporting Software Processes Through Software Process Modeling." dated August 1990. *Proceedings of the 6th International Software Process Workshop: Support for the Software Process*, October 29-31, 1990 Hakodate, Japan. Published by ACM
- [LaM92] LaMonica, F., "System Engineering Concept Demonstration Overview", Proceedings of the Second Annual International Symposium of the National Council On Systems Engineering (NCOSE), Seattle WA, 20-22 July 1992.
- [LAV89] Jonah Z. Lavi and Michael Winokur. "ECSAM - A Method for the analysis of complex embedded computer systems & their software," *Proceedings of the Structured Techniques Association Conference STA5*, Chicago Ill. May 1989



- [MDC73] MDC Report J1501-4, dated 21 March 1973; Long Range Patrol Aircraft, Volume II, Book 4, System Engineering. Douglas Aircraft Company, 3855 Lakewood Blvd, Long Beach CA, 90846
- [MOR59] "Integration of System Engineering with Component Development", by J.A. Morton, Electrical Manufacturing, Vol. 64, pp. 85-92, August 1959.
- [ORT92] Ortiz, A. "System Engineering Concept Demonstration - Process Model", Proceedings of the Second Annual International Symposium of the National Council On Systems Engineering (NCOSE), Seattle WA, 20-22 July 1992.
- [ORT92] Ortiz, A., "SECD Process Model Supporting Document, Final Technical Report, Volume 3, Rome Laboratory, Griffiss Air Force Base NY, May 1992
- [OST87] Leon Osterweil. "Software Processes Are Software Too." Proceedings of the 9th International Conference on Software Engineering, IEEE, 1987, pp2-12.
- [OST91] Leon Osterweil, "Software Process Research at UCI", dated 18 January 1991. Irvine Research Unit in Software (IRUS) Inagural Convocation.
- [SEMC90] Robert M. Pedraglia, "Natural Working Group Summary Findings On SystemEngineering Management Capabilities". This is an unpublsh presentation
- [SEMG90] System Engineering Management Guide, dated January 1990; developed by the Defense System Management College and printed by the Superintendent Of Documents, U.S. Printing Office, Washington, D.C. 20402.
- [SE&A90] Blanchard, Benjamin S. and Fabrycky, Walter J., "System Engineering and Analysis", second edition; published by Prentice Hall Inc., A Division of Simon & Schuster Englewood Cliffs, New Jersey 07632
- [STS86] Standard Taxonomy for Software Engineering Standards (P1002/D08), dated 14 October 1986 (REVCOM Version); Sponsor by Technical Committee on Software Engineering of the IEEE Computer Society; Prepared by Software Engineering Standards Taxonomy Working Group of the IEEE Computer Society
- [TMSA81] Technical Marketing Society Of America, "System Engineering Advanced Techniques", November 3, 1981

## **Appendix A - Literature Survey Of Existing System Engineering Processes And Models**

The system life cycle encompasses the system acquisition process, the system development process, and the System Engineering process. Each process is contained within the other respectively– acquisition into development and development into engineering. The process models surveyed in this appendix encompass information about these three system life cycle processes. It is important to examine these models carefully as they provide insight into the activities and work products of these processes.

Although the System Engineering process has many objectives and criterion, its main thrust is to solve a problem. Complicating this process however are two difficulties: figuring out the exact nature of the expected results and choosing one of the numerous alternatives that exist to obtain the desired result. The successful solution of a System Engineering problem tends to follow a general approach in which an iterative method is employed to discover and refine the expected results, as the initial assumptions are compared with the derived data. Solutions to System Engineering problems can be postulated with a set of questions. These questions determine requirements, establish the objectives, goals, constraints, and determine the weighting functions which place emphasis on the various system requirements.

The system requirements are determined from a consideration of the user's stated needs, (e.g., from specifications or previous experience, or from a general knowledge of the same or similar processes). These include the answers to questions such as [MOR59]:

- What should the system do?
  - Performance (size, weight, efficiency, appearance, etc.)
  - Cost (absolute, relative, or competitive situations)
  - Time (when product is wanted, time required to produce)
  - Reliability (life, failure rate, etc.)
- What environment does it have to operate in?
  - Home, commercial, military
  - Power supply variations, maintenance, service

- Physical (size; power available; mechanical, electrical, chemical conditions)
- Existing equipment (what can be changed, what cannot be changed)
- What environment is the product to be made in?
  - Engineering skills and facilities
  - Manufacturing skills and facilities
  - What other products are being engineered and built
  - What materials are available

The weighting factors used, with the aforementioned information described above, should be determined correctly. Significant factors in formulating the problem are the answers to such questions as:

- What information is available?
  - Given
  - Required
  - Known
  - Unknown
- What are the inputs?
  - Signal sources (amplitude, time, and/or frequency responses tolerances, accuracy)
  - Power sources
  - Disturbances and noises
  - Regulation of input sources
- What are the output characteristics?
  - Signal distribution (amplitude, frequency)
  - Noise limitation (amplitude, frequency)

- Accuracy
- Loads and impedances of output

The preceding questions are aimed at defining the system. The system engineer must be careful not to overlook significant features while, at the same time, exclude nonessential or time-consuming details.

Chestnut identifies five standards for system judgment by which the customer determines the overall worth of a system. These standards should be used as a guideline to customer satisfaction [CHE65]:

1. Performance
2. Cost
3. Reliability
4. Time (to build and install, life of system)
5. Maintainability

The system engineer has these design criterion to judge customer satisfaction. Typically, system engineers place emphasis on performance and cost, while the value of the system to the customer is measured by the above mentioned criteria. Therefore, the System Engineering process must also support the customer's value judgments.

The next section discusses the classical definitions of the System Engineering process in which operational requirements are transformed into optimal technical requirements. It was important to clearly identify the System Engineering process activities and decisions in order to achieve the desired results. At the same time, we examined other definitions that provided a different perspective about the System Engineering process.

### **A-1.0 System Engineering Management Guide**

This guide defines the System Engineering process as *"iteratively applied and consists of primarily four activities. These activities are functional analysis, synthesis, evaluation and decision, and a description of system elements"* [SEMG90]. Figure A-1.0 illustrates the high level generic System Engineering process found in [SEMG90]. The final product of this version of the System Engineering process is a set of production ready documents of all system elements. This definition of the process emphasizes the iterative nature of the method.

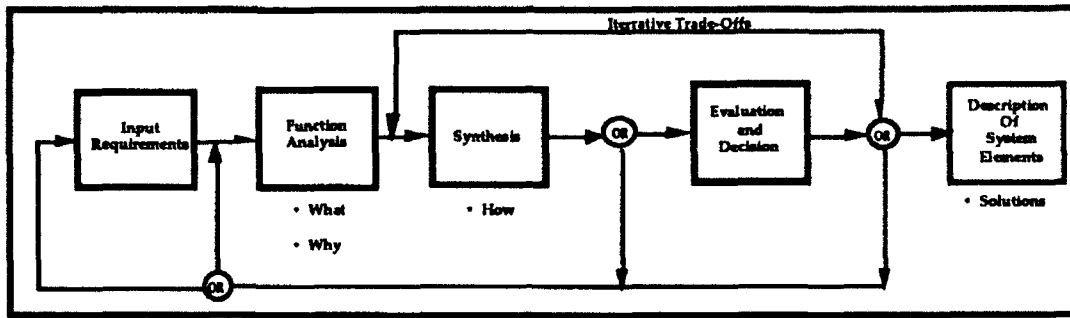


Figure A-1.0 [SEMG90] System Engineering Process

Figure A-1.0 expresses the term 'functional analysis' as the questions "What" and "Why" and synthesis as "How". This definition implies that during functional analysis we ask ourselves what the necessary functions are and why are they there. The synthesis subprocess collects the system functions together by answering the question of how we put these functions together in order to achieve our objectives? The next subprocess evaluates the alternatives (tradeoffs) and makes a decision. Notice the feedback loop, at the top of Figure A-1.0, which represents the various alternatives considered. The output of this process is a fully documented description of each system element.

## A-2.0 System Engineering Advanced Techniques

This report defines the System Engineering process as "A logical sequence of activities and decisions transforming an operational need into a description of a preferred system configuration and its performance parameters" [TMSA81].

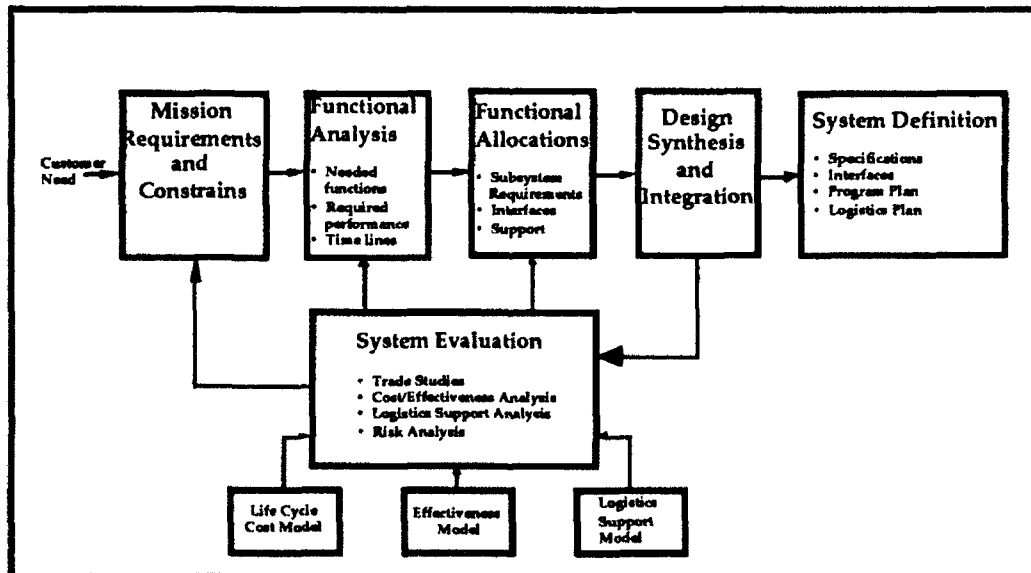


Figure A-2.0 [TMSA81] Generic System Engineering Process

Figure A-2.0 illustrates a high level generic System Engineering process. This process is a more detailed System Engineering process, as compared to the one shown in Figure A-1.0, because it provides more information about the basic System Engineering functions. The added value of the System Engineering process in Figure A-2.0 is the utilization of Life Cycle Cost, Effectiveness and Logistics Support models. Until now, the application of formal models for system evaluation was not mentioned. The Life Cycle Cost and Logistics Support are formal, approved models that are used widely for a specific type of system. An effectiveness model consists of a simulation of the system elements and its desired performance. This exercise (simulation) is typically performed to validate the design and to discover any latency errors. System simulations allow the measurement of effectiveness and other critical technical parameters. The effectiveness model also helps to assess reliability, maintainability, and safety in the analysis of the system performance.

### A-3.0 Long Range Patrol Aircraft, Volume II, Book 4, System Engineering

This report provides a complete System Engineering Implementation Plan (SEIP) for a long range patrol aircraft [MDC73]. Figure A-3.0 illustrates a general System Engineering process. This System Engineering process is more detailed than the one presented in Figures A-1.0 and A-2.0 and is specifically applicable to aircraft systems.

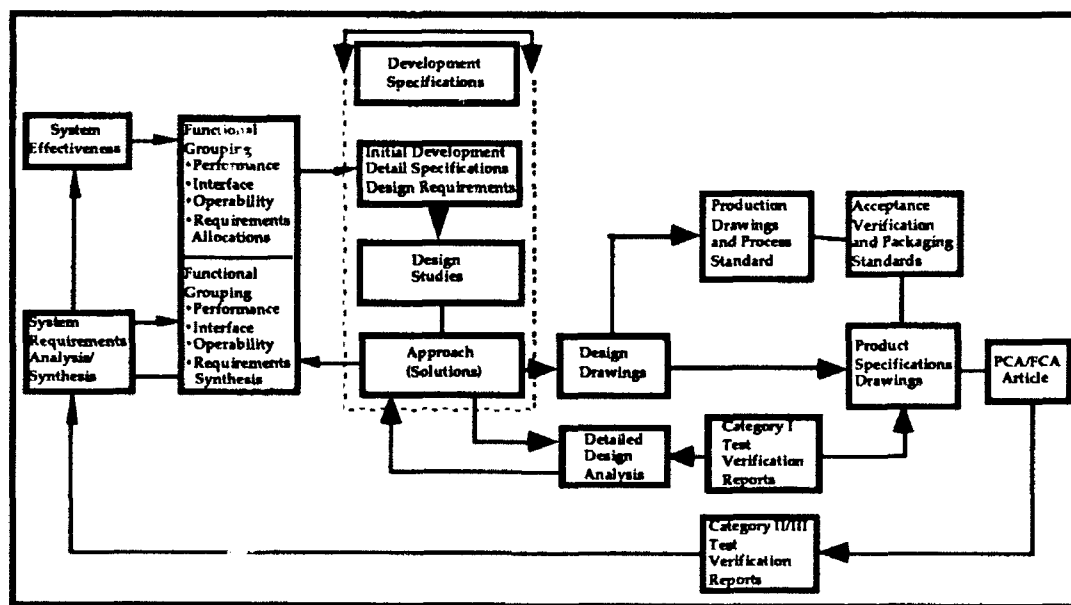


Figure A-3.0 [MDC73] General System Engineering Process

This System Engineering process for an aircraft includes Category I, II, and III test verification reports. The process output is the completion of the Functional Configuration Audit (FCA) and the Physical Configuration Audit (PCA). This particular process does not address the production, deployment or retirement phases.

#### **A-4.0 MDC-DAC System Engineering Process**

In June 1990 during the SECD program, an effort was made to document the MDC-DAC System Engineering Process. An MDC-DAC set of procedures and guidelines was identified and a Procedural Guideline Tree was developed. To combine these procedures and guidelines, an MDC-DAC System Engineering process model was developed.

The final MDC-DAC System Engineering Process Model extended to the early activities of the system life cycle (i.e., Concept Exploration to Full Scale Development). It emphasized the Concept Exploration Phase. The processes presented in the model are the author's interpretation of MDC-DAC procedures and guidelines. Other sources were used when needed to provide sufficient information to complete the documentation of these processes.

#### **A-5.0 Booz-Allen-Hamilton**

The Defense Science Board (DSB) established an Acquisition Task Force of which Booz-Allen-Hamilton (BAH) was contracted to support. The objective of this task force was to document and streamline the DoD system acquisition process, address relevant issues, and recommend an improved prototype acquisition process.

The BAH effort was divided into three phases. In the first phase, BAH surveyed over 150 companies across the United States. This effort produced a generic acquisition process, a timeline and specific drivers/accelerators of process time usage. In the second phase, BAH provided potential process changes to both government and contractor acquisition life cycle and assessed the impact of these recommended changes on the cycle time. In the third phase, BAH developed a formal plan to ensure the implementation of the recommendations generated in the second phase.

#### **A-6.0 EP1X Systems Engineering Report**

The EP1X System Engineering report defined and documented the System Engineering process at MDC-DAC from a procedural point of view. This report defined the system life cycle phases as it applies to the System Engineering

process, provided a historical perspective, and established System Engineering objectives, and definitions.

The report presented a historical perspective of System Engineering and defined the System Engineering problem and the need for a discipline with a systematic and methodical approach to design. The following definitions were provided in the report as a reference.

- **SYSTEMS** - An organized set of doctrine, ideas or principles intended to explain the arrangement or working of a systematic whole.
- **SUB-SYSTEM** - Those logical divisions which permit a system to be broken or sub-divided into sections.
- **INTEGRATION** - The portion of management and technology in which the interdependence of material, device, circuit, and system-design consideration is especially significant and is blended into a functioning or unified whole.

"The System Engineering (System Integration) method requests the the desired results and the translation of the requirements convertes them into a functional set of interrelated processes (units) to perform a specific function (tasks). The technologies necessary to achieve this goal are integrated at the beginning of the process (conception) and carried on through-out the life cycle of the product. Here, the laws of Physics, Chemistry, Electro technology and other scientific disciplines can not be violated without definite and certain failure. These disciplines enable the engineering processes to occur concurrently.

The Concurrent Engineering process, as implied by the name, is how we perform the product definition activities with sub-contractors, partners, suppliers and others based on the work package. It is intended to cause the developer to consider the related processes, the efficiency, the quality, the cost, the schedule and user requirements to support the process."

"When development programs were relatively simple to manage, engineering efforts could be directed by a few top managers. Communication between participants was uncomplicated; functions and responsibilities were easily stated; and decision-making in regard to cost performance and schedules were fairly straight-forward. As the state-of-the-art advanced, science and engineering expanded along highly specialized functional lines. They increased in importance and complexity, and required more sophisticated management.

The problems of communication, coordination, direction and control of these specialties among geographically separated personnel have become increasingly severe. Some specialists are grouped into "functional" organization. to coordinate the state-of-the-art across more than one program and to time-share between



programs. In other instances, specialists are divided into program oriented organizations or a compromise bilateral organization may be adopted, i.e., vertical build units and horizontal teams.

With advanced and increasingly complex new programs, there is a need for increased rigor in the following technical and managerial activities: (1) Control of the design interfaces among systems, equipment, personnel, facilities, and computer programs. (2) Use of trade-off analysis techniques in allocation of functions, selection among design approaches and resolution of conflicting design objectives constraints. (3) Assurance that the performance specifications, detail design, and production packages are consistent with the fundamental (mission requirements and with the "ilities" (i.e., producibility, operability, supportability, reliability, safety, compatibility,) interfacing with systems, equipment, personnel facilities and computer programs.

The development of solutions to the problems of communications, direction and control requires methodical analytical approaches to the development of a total system. These approaches are termed System Engineering. The sequential and iterative method for top-down development of a product and its technical program task elements is known as the System Engineering process.

The total design process encompasses system analysis, definition, synthesis of requirements, preliminary design and detail design. System analysis is the analysis and transformation of material requirements into a theoretical model with quantitative terms, and the manipulation of the model in simulation of the operational environment. Definition and synthesis of requirements is the translation of definition performance objectives of a selected system approach into design criteria (design-to-requirements) for the individual elements which will comprise that system. Preliminary design develops the design approach for the system and its elements based upon the criteria provided by definition and synthesis of requirements. Detail design translates the design approach into a manufacturing configuration which can be produced and supported within the state of existing economically achievable manufacturing technologies and support capabilities. System Engineering integrates the engineering effort throughout the design process."

Draper's set of objectives and guidelines provided a frame of reference for the system engineering process:

- "a. Ensure that the engineering effort is fully integrated and reflects adequate and timely consideration of design, test and demonstration, production, operation, and support of the system equipment.

- b. Ensure that the definition and design of the system or equipment item are conducted on a total system basis, reflecting equipment, facilities, personnel data, computer programs, and support requirements to achieve the required effectiveness in acceptable risk, cost, and schedule consideration.
- c. Integrate the design requirements and related efforts of reliability, maintainability, integrated logistics support, human factors engineering, health, safety, and other specialties with respect to each other as well as into the engineering effort.
- d. Ensure compatibility of all interfaces within the system, including the necessary supporting equipment and facilities; ensure the compatibility and proper interface of the system with other systems and equipment that will be present in the operational environment.
- e. Establish, control, and maintain an effective work breakdown structure throughout the life of the system project in accordance with applicable directives.
- f. Evaluate effects of changes on overall system performance, effectiveness, schedule, and cost; ensure that all affected activities participate in the evaluation of changes.
- g. Provide a framework of coherent system requirements to be used as performance, design, and test criteria; provide source data for development plans, contract work statements, specifications, test plans, design drawings, and other engineering documentation.
- h. Measure and judge technical performance for the timely identification of high risk areas.
- i. Document technical decisions made during the course of the programs."

In an organized process where scientific and engineering efforts are combined to produce the desired results, the laws of Physics, Chemistry, Electro technology and other scientific disciplines can not be violated without definite and certain failure. The application of scientific disciplines provide the following activities:

- Definition of operational need.
- Selection of the best configuration which satisfies the operational need.
- Transformation of an operational need into technical requirements and products.
- Flow-down of the technical requirements.

- Establishment of the measure's effectiveness.
- Integration of related technical parameters.
- Assurance of compatibility of all physical attribute (mass, size, etc.).
- Assurance of functionality.
- Identification of program interfaces in a manner which optimizes the total system.
- Integration of the efforts of all disciplines and specialties.
- Definition of the process to be used.

The life cycle consists of a sequence of events, with each event and its sequence important to understanding the need for an integrated process. The following steps define the complete process of building a product according to the EP1X report.

- LIFE CYCLE OF A PRODUCT:
  - NEEDS/REQUIREMENTS
  - CONCEPT
  - RESEARCH & ANALYSIS OF CONCEPT
  - DESIGNING
  - TOOLING
  - MANUFACTURING
  - TESTING & EVALUATING
  - DELIVERY
  - OPERATING
  - CUSTOMER SUPPORT
  - DISPOSAL

In addition, this report defines the System Engineering process from beginning to end as follows:

- Define The Process For:
  - Organization Phase
    - Funding
    - Staffing
  - Mission/Needs Phase

- Requirements
- Constraints
- Concept Exploration Phase
  - Research & Analysis
  - Modeling
  - Preliminary Design
  - Baseline
- Development/Production Phase
  - Full Scale Engineering Development (FSED)
  - Production
- Testing & Evaluation Phase -Demonstration -Validation  
Airworthiness
- Delivery - Acquisition Phase -Change from Manufacturer to Owner
- Operating Phase
  - In-Service
  - Performance
  - Reliability
- Customer Support Phase
- Maintenance
  - Modifications
  - Training
- Disposal
  - Dismantling
  - Recycling

#### **A-7.0 NAWC Instruction 5451.2**

This section discusses the Navy Air Warfare Center (NAWC), *Systems and Engineering Group Instruction (S&E INST) 5451.2* process model. This process representation encompasses the entire system life cycle and is used a guide and reference for the overall activities, processes, and work products. It represents a compendium of various standards and practices.

## A-8.0 Chestnut's Model

Chestnut provides a variety of System Engineering models in his book "System Engineering Tools" [CHE65]. One of the most interesting models researched was Chestnut's interrelationship of performance, reliability, and cost. This model meshes together a logical representation of performance, reliability, and cost loops with their respective constants for functional commonality between parameters. This mathematical representation allows derivation of equations in terms of functions.

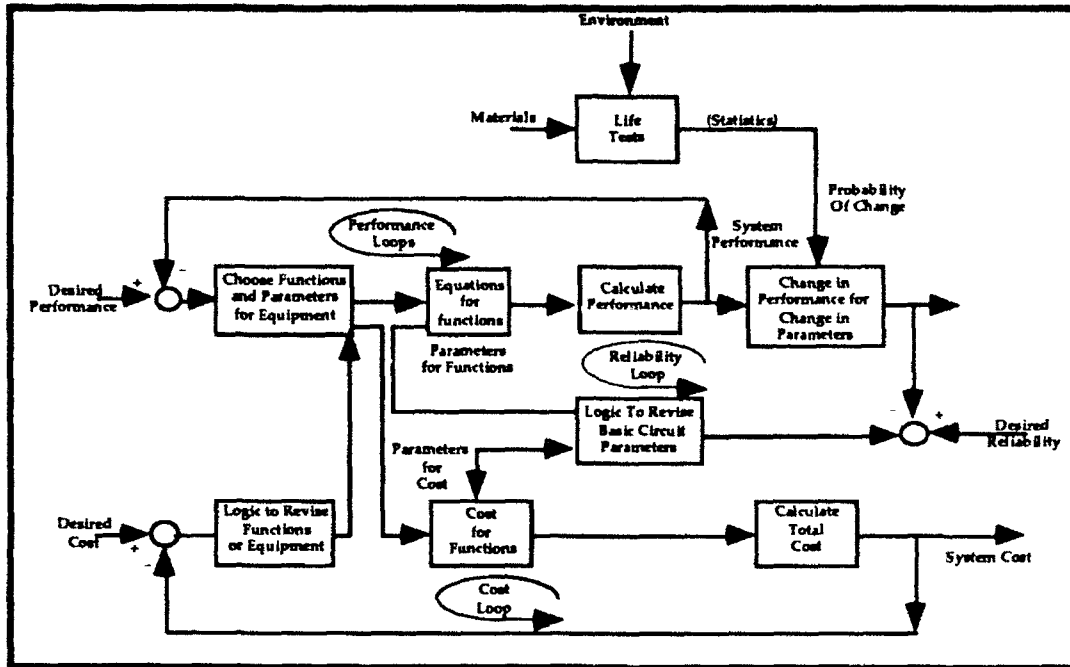


Figure A-8.0 Interrelationship of Performance, Reliability, and Cost Loops

Figure A-8.0 illustrates the relationships between the representations. Although this work was published in 1965, the concept and idea of functional interrelations of parameters and functions is still being studied today. Notice the parameters are the loops and not the functional blocks.

## **Appendix B Document Summaries**

This appendix provides summaries for three documents used in the development of the Process Model:

1. "A Systems Engineering Methodology for the Advanced Tactical Aircraft (ATA)" written by Stephen J. Kapureh from the NAVAL Post Graduate School.
2. "System Engineering Group Instruction 5451.2 from the Department of the Naval Air System Command" written by the Naval Air System Command Headquarters.
3. "Software Process Modeling: Value and Experience" written by Mark Kellner from the Software Engineering Institute.

### **B-1.0 A Systems Engineering Methodology For The Advanced Tactical Aircraft (ATA)**

Written by Steven J. Kapureh, this paper presents methods to apply a Systems Engineering approach throughout a project's lifecycle. The author presents these methods in the form of a methodology and a synopsis of principles, each of which can be utilized in the development of a Systems Engineering program.

This paper's main contribution to the SECD program was its definition of the terms 'system' and 'System Engineering' and a discussion of the two basic categories of System Engineering modeling. The insight learned from these definitions and ideas were incorporated into the defined terms for SECD, refer to Section One, and were used in the creation of the Process Model. The following paragraphs summarize the main points.

The author synthesized his definitions from ten different sources; hence, they provided very complete and accurate information. A system is defined as "a composition of requirements capable of performing and/or supporting an operational role. A complete system includes related facilities, equipment, material services, software, technical data, and personnel. A system requires each of these items to support itself as a self-sufficient unit within its intended operational and/support environment."

The author moves on to describe the System Engineering as follows. "System engineering begins with the identification of an operational requirement for a system. The next step is to identify the constraints and environment in which the system will be developed, produced, and operated. At this point, scientific and engineering skills can be utilized to transfer the qualitative operational requirement into quantitative parameters. These parameters will then be

decomposed level by level from system to subsystems to parts and finally to component levels. It then becomes an iterative process of analyzing the performance parameters, designing a solution, testing, and evaluation. Tradeoffs are then made on the subsystems based on weighted objectives established by the cost, schedule, and performance characteristics of the total system. These subsystems are then integrated into the total system."

System Engineering, the author also states, is the application of scientific and engineering efforts to

- I. Transform an operational need into a description of system performance parameters and a system configuration through the use of an iterative process of definition, synthesis, analysis, design, test and evaluation.
- II. Integrate related technical parameters and ensure compatibility of all physical, functional, and program interfaces in a manner that optimizes that total system definition and design.
- III. Integrate reliability, maintainability, safety, survivability, human and other such factors into the total engineering efforts to meet cost, schedule and technical performance objectives.

To show the correlation between Systems Engineering and a lifecycle, the author described the systems lifecycle for a typical weapon system acquisition. This lifecycle is divided into six basic phases:

- a. Mission Need Determination - objective phase; a large percentage of the costs become fixed during this phase.
- b. Concept Exploration - functional analyses phase; inherent in the analyses are cost and risk assessments.
- c. Demonstration and Validation - simulation phase; the team performs these activities and writes the System Engineering Management Plan (SEMP), which includes plans for verification and risk allocation.
- d. Full scale development - implementation phase; SEMPER is implemented so that detailed system simulation and models can be developed to predict system performance.
- e. Production and Development - modification phase.
- f. Retirement - lessons-learned phase; lessons learned from completed projects are documented so they can be applied to the new program. This phase helps to solve the problems of future systems.

After researching the above terms and lifecycles, the author states that his work revealed two basic categories of System Engineering models: quantitative model and qualitative model. The quantitative model uses mathematical representations to describe the system. The qualitative model uses words and symbols to portray the system. An example of each model was discussed.

An example of a quantitative model is one developed by Arnold and Stepler. Their model can be thought of as a three-tiered wheel. Systems Engineering is the first tier of the wheel. The second tier models a typical program office where the functional managers provide information to the first tier. The third tier depicts a number of tasks that must be executed in the development of a system.

An example of a qualitative model is one developed by the Army Management Training Agency. In this model each software phase corresponds to different states in the lifecycle of a program. System Engineering management encompasses the System Engineering process and the integration of all engineering activities and technical aspects of the system/project from requirement phase through to delivery phase.

As mentioned above, the author's definitions and model descriptions were helpful aids while developing the Process Model. The most important lesson learned from this paper is the importance of performing Systems Engineering to insure the success of large, complex systems.

### **B-2.0 System Engineering Group Instruction 5451.2 from the Department Of The Naval Air System Command**

This report, written by the Naval Air System Command Headquarters, establishes the requirements, procedures, and responsibilities for conducting systems engineering within the Naval Air Systems command for aircraft systems and weapons. The report establishes policy procedures, defines responsibilities and assigns action to be taken by appropriate personnel. It also includes checklists for:

- a. The Systems Engineering Process
- b. Systems Engineering Products
- c. Systems Engineering Implementation Plan Guide
- d. Review Questions and Checklist

This report establishes policy for the Systems Engineering Management, a group which ensures Systems Engineering compliance, and the Systems Engineering Technical Support Team, a group which plans and manages all top-level System Engineering and ensures it is performed at the system level.



The instructional documents provided in this report are the result of a System Engineering advocate position, AIR-05A1. This position was established within the NAVAIR to coordinate items of interest to the Systems Engineering Core Team (SECT) and to assist in the implementation and execution of systems engineering support efforts by SECT. The report provides detailed procedures to be followed, one of which was the System Engineering procedure. Figure B.2-1 illustrates the System Engineering Process and System Development Lifecycle.

The System Engineering procedure provided important information for our SECD research. This procedure explained the iterative System Engineering process activities and the products which result from these activities. This procedure also provides a Systems Engineering implementation guide. A summary of this procedure is provided in the following paragraphs.

The System Engineering process is an iterative process which consists of the following (refer to Figure B.2-2):

1. Mission requirements and constraints analysis
2. Functional analysis
3. Functional allocation
4. Design requirements
5. System Synthesis and integration
6. System evaluation

The System Engineering activities produce the products for the following four phases:

- (1) concept exploration phase
- (2) demonstration and validation phase
- (3) full-scale development phase
- (4) production and deployment phase

Sample products for each phase are listed below;

1. Concept Exploration Phase -

example products for this phase consist of the System Engineering Implementation Plan (SEIP), Mission Analysis, Trade studies, Position papers, Functional baseline, Lessons learned - Naval Aviation Lessons Learned Document (NALLD), Risk analysis report, Cost and schedule estimates, General system specification. (refer to Figure B.2-3)

## 2. Demonstration and Validation Phase -

example products for this phase are System Engineering Implementation Plan (SEIP) Update, Refined System Engineering requirements, Trade-Off Studies, Functional Baseline, Preliminary Allocated Baseline, Lessons learned, System/technical Interfaces, Risk analysis report. (Refer to Figure B.2-4)

## 3. Full-Scale Development Phase -

example products for this phase include System Engineering Implementation Plan (SEIP) Update, Trade-off alternatives, Development Monitor, Design reviews, Functional Configuration Audit, Risk Analysis Report, Data Package Review, Allocated Baseline, Product baseline. (Refer to Figure B.2-5)

## 4. Production and Deployment Phase:

example products for this phase include Systems Engineering Implementation Plan (SEIP) update, Engineering Change Proposal (ECP) evaluation/impact, Data Package Validation, Deficiency Reports, Waivers and deviations, Operational Safety and Improvement Program (OSIP), Physical Configuration Audit (PCA). (Refer to Figure B.2-6)

An implementation guide provides a detailed description of the subjects that should be addressed in each of the above phases. The following are examples of subjects in an implementation guide:

- (a) Illustration of the system, systems engineering and program Work Breakdown Structure (WBS) to at least the top three levels
- (b) Description of the System, Systems Engineering and Program WBS with assigned codes for each reference
- (c) Summary of major objectives
- (d) Major Milestones - with top level timeline
- (e) Program Master Schedule - System Engineering
- (f) Resource Summary
- (g) Critical Resource
- (h) Responsibilities - names and /or codes
- (i) Budget - identify the cost by acquisition phase

The findings of this report showed us that Systems Engineering is recognized as a discipline. The report's specific engineering procedures provided a precise guide of how to perform System Engineering during development.

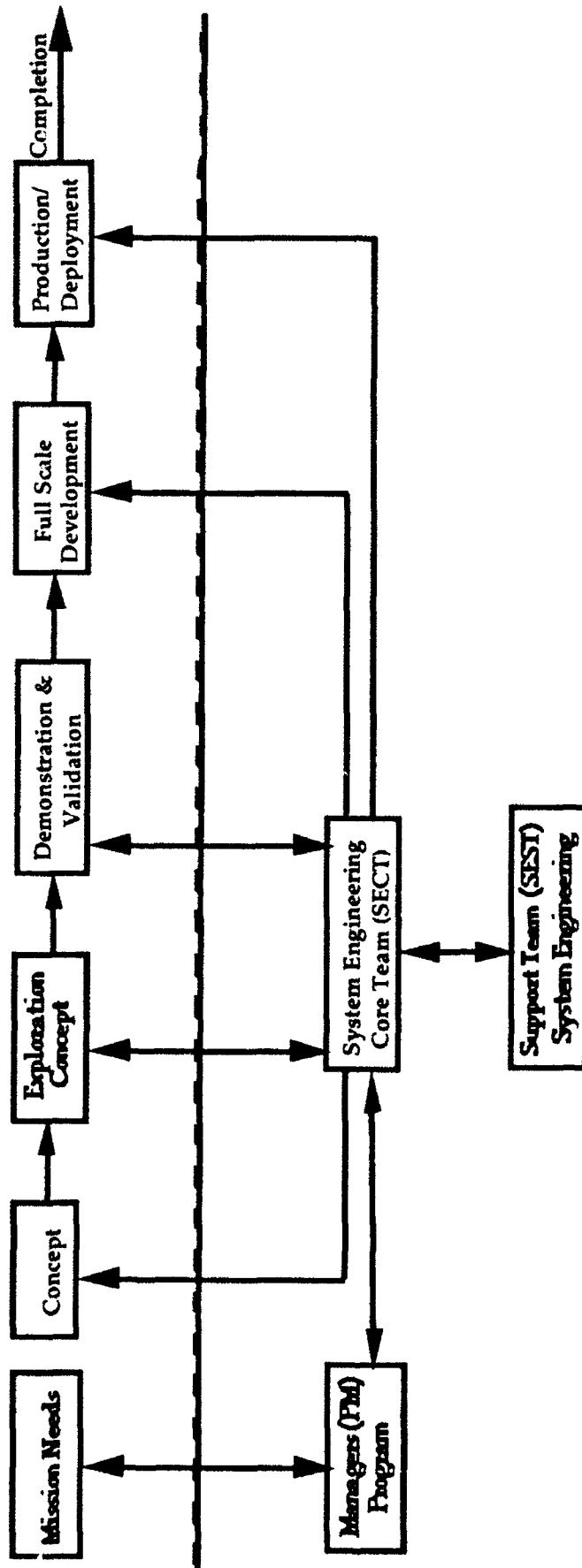


Figure B.2-1 System Engineering Process and System Development Life Cycle

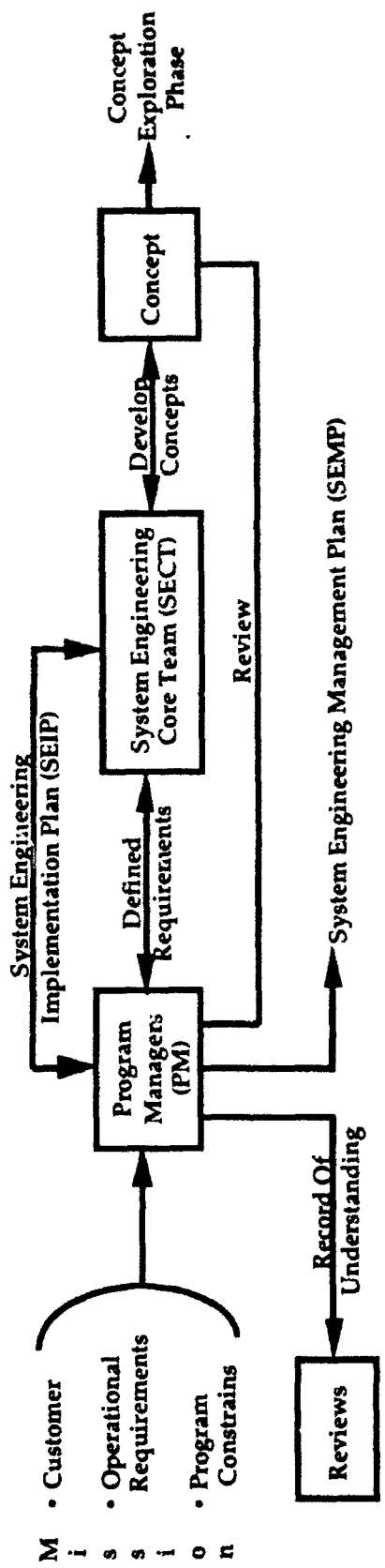
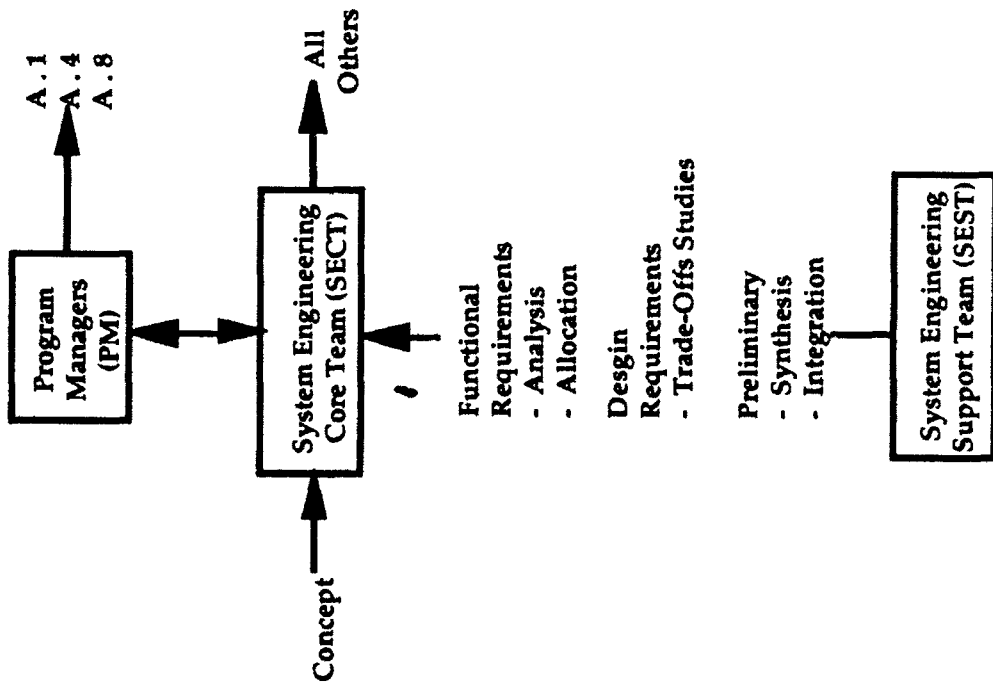


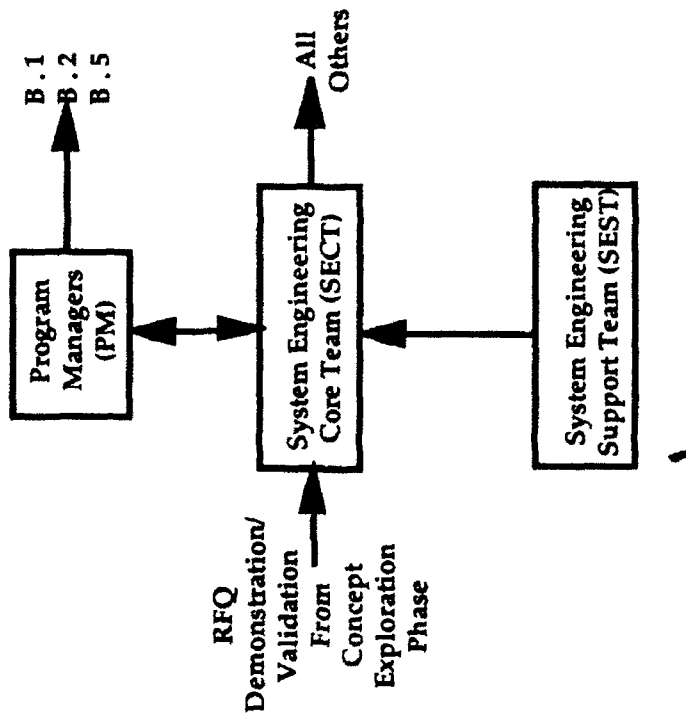
Figure B.2-2 System Development Mission Needs Interfaces with System Engineering Process



NUMBER	TITLE
A. 1	System Engineering Implementation Plan (SEIP)
A. 3	Trade-Offs Studies
A. 5	Functional Configuration Baseline
X. X	
A. 6	Lessons Learned
A. 7	Risk Analysis Report
A. 10	RFQ for Demonstration and Validation Phase
A. 12	Integrated Design Assessment Report ("ilities")
A. 13	Preliminary Test & Evaluation Plan (TEMP)
X. X	
A. 11	Technical Performance Measurements
A. 14	PrePlanned Product Improvement (P3I)
A. 15	Program History File
A. 16	Resource Requirements Definition
A. 17	Review & Briefing Material
A. 18	System Eng. Support Team & Core Team Support
X. X	
A. 2	Mission Analysis
A. 4	Position Papers
A. 8	Cost & Schedule Estimates
A. 9	General System Specification

Figure B.2- 3 System Engineering Process Concept Demonstration Products

D e m o n s t r a t i o n a n d V a l i d a t i o n

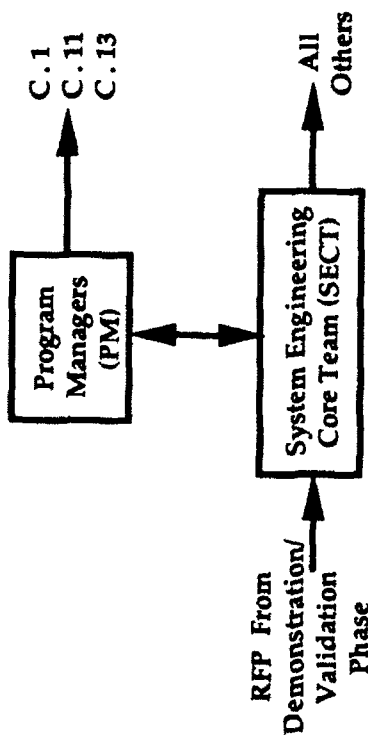


NUMBER	TITLE
B.1	System Engineering Implementation Plan (SEIP)
B.3	Trade-Offs Studies
B.4	Functional Baseline
B.5	Preliminary Allocated Base
B.6	Lessons Learned
B.8	Risk Analysis Report
B.13	RFQ for Full Scale Development Phase
B.14	Integrated Test Plan
B.15	Test & Evaluation Master Plan (TEMP)
B.16	Design Reviews
B.17	Technical Performance Measurements Tracking
B.18	PrePlanned Product Improvement (P3I) Update
B.20	Program History File
B.21	Resource Requirements Definition Update
B.22	Review & Briefing Material
B.23	System Eng. Support Team & Core Team Support
B.2	Refine System Requirements
B.7	System/Technical Interfaces
B.9	Proposal and Source Selection
B.10	Demonstrations (Prototypes)
B.11	Type "A" Specifications
B.12	Preliminary "B" Specifications
B.19	Producibility Validation

Figure B.2-4 System Engineering Process Demonstration and Validation Products

**PRODUCTS OF FULL SCALE DEVELOPMENT PHASE**

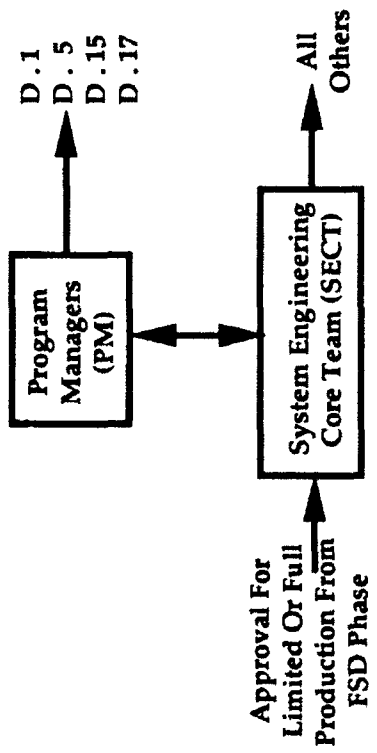
NUMBER	TITLE
C.1	System Engineering Implementation Plan (SEIP) Update
C.2	Trade-Offs Alternatives
C.9	Product Baseline
C.5	Allocated Base
C.10	Lessons Learned
C.6	Risk Analysis Report
X.X	
C.15	Integrated Test Plan
C.14	Test & Evaluation Master Plan (TEMP)
C.4	Design Reviews
C.16	Technical Performance Measurements Tracking
C.17	PrePlanned Product Improvement (P3I) Update
C.18	Program History File
C.19	Resource Requirements Definition Update
C.22	Review & Briefing Material
C.23	System Eng. Support Team & Core Team Support
C.7	Data Package Review
C.20	Safety Plan
C.3	Development Monitor
C.5	Functional Configuration Audit
C.21	GFE Specifications Review
C.11	Approval For Limited/ Full Production
C.12	Review DT & OT Test Plans & Monitor Results
C.13	Certification Of Readiness For OP EVAL.



F u l l S c a l e D e v e l o p m e n t

Figure B. 2- 5 System Engineering Process Demonstration and Validation Products





NUMBER	TITLE
D.1	System Engineering Implementation Plan (SEIP) Update
X.X	
X.X	
X.X	Lessons Learned
D.8	
X.X	
X.X	
X.X	
D.11	Test & Evaluation Master Plan (TEMP)
X.X	
X.X	
D.12	PrePlanned Product Improvement (P3I) Update
D.10	Program History File
X.X	
D.18	Review & Briefing Material
D.19	System Eng. Support Team & Core Team Support
D.3	Data Package Review
D.16	Safety Plan
D.2	Engineering Change Proposal (ECP) Evaluation/Impact
D.4	Deficiency Reports
D.7	Physical Configuration Audit (PCA)
D.9	Production Acceptance Test & Evaluation (PAT&E)
D.13	Development Test & Follow-On Test & Evaluation
D.14	Operational Flight Trainers
D.17	Fleet Interface
D.15	SLEP/CILOP
D.6	Operational, Safety & Improvement Program (OSIP)
D.5	Waivers & Deviations

Figure B.2-6 System Engineering Process Production/Deployment Products

### **B.3.0 Software Process Modeling: Value and Experience by Mark Kellner**

*"The Software Process Modeling: Value and Experience"* describes the Software Engineering Institute (SEI) experience with organizational process models. The paper presents processes in a technical and managerial framework for applying people, methods, tools and practices to a software development effort. It classifies the process maturity framework as Initial, Repeatable, Defined, Managed, Optimizing. These classifications are hypothesized to be representative of the historical phases in evolutionary improvements of actual organizations.

The paper develops the idea of process improvement with statements such as "Process improvements efforts have been successfully applied in manufacturing industries for decades." It follows this idea by introducing four basic steps for process improvement. These steps are "(1) Understanding the current process; (2) Identify and analyze areas of opportunity for process improvement; (3) Select the improvements to be implemented next, and put them into practice; and (4) Cycle back to (1) to evaluate and understand the impact of the implemented changes." The paper states that "these processes must be adjusted to accommodate changes to user requirements expectations; methods, techniques, and tools; skill level and training of professionals."

Four objectives of SEI software process modeling work are presented. These objectives are "(1) increasing understanding regarding a process; (2) Supporting evolutionary improvements to a process; (3) enabling processes to be formally defined and applied prescriptively; (4) Facilitate effective management of a process." The benefits of software process definition are presented in a symbolic form, for example "Analogous to football, such a definition embodies both the game plan and the play book for software work; it is important that the process participants (players) know the plays in detail (play book), and have a bigger picture of when to use them (game plan)."

Advocacy for an automated approach is made to implement software process modeling requirements. The paper suggests that the software process models must possess capabilities in Representation power, Comprehensive analysis, and Forecasting. This idea is elaborated by establishing the importance of representing software processes "as-is" and "to-be" and also identifying restrictions imposed by regulations, standards, and directives. The idea of testing models in areas of consistency, completeness, and correctness to determine its validity is introduced. A criteria is set for models to yield

predictions and the usage of this forecasting capability to answer "what if" type questions.

The paper explains that "The term "enactable" has been used to denote that a process model must be precise and capable of being executed, run, interpreted, etc.." A relation is established between the analysis and predictive capabilities as a consequence of the power of enactable process models.

The value of experience in software process modeling is presented by suggesting a prototype approach for the development of software process modeling techniques. The SEI approach to software process modeling is as follows:

1. Develop a list of important capabilities in software process modeling
2. Examine automated technologies suitable as modeling tools
3. Select a tool and applied it to a component of a real-world software process
4. Feedback the lessons learned into developing the approach

SEI's specific modeling approach was developed using a commercial tool called STATEMATE. This tool was originally developed to specify and design real-time reactive system. SEI's success in modeling organizational processes has enabled them to conclude that the tool is suitable for organizational process modeling. This approach represents a process from a functional, behavioral, and organizational prospective with Activity charts, State charts and Module Charts respectively. The author states "This concept of prospectives is analogous to the different viewpoints presented in engineering drawings of a three-dimensional (3-D) object." This section of the paper ended with an extended discussion of analysis and capabilities of the enactable process models developed using Statemate. Emphasis is given to the qualitative aspects of the Statemate models.

The paper presents a synopsis of the experience gained during the development of the process in use by the U.S. Navy to support operational software for the F-14A aircraft. A similar effort was conducted for the Air Force F-16A/B aircraft. These models depict the full software support process.

Model construction is explained as "The construction of the process varies according to the type of model under development." The process of developing a model was iterative over several months. SEI used a two person team who employed an informal interview format. A high level model was constructed after the 1st round of interviews. An intensive verification session with process experts was held. This verification consisted of a manual review of the model and all its parts.

Brief examples are drawn from SEI's Navy F-14A software support process. A high level view with detailed coverage of major steps in an activity chart was developed. This represents the functional perspective of the process. The behavioral perspective of the process is presented in a state chart. Concurrence of various high level activities is addressed here. A detail view of a portion of the process is presented to illustrate the modeling approach. A description of the process of assigning an engineer to investigate the problem is given with examples of semantics used to define, describe, and model the process.

An observation is made about the level of detailed presented in the F-14A model examples and the fact that they do not dive into a lower level where creativity intuition are key components. The model captures judgments such as determination to end investigation. It is pointed out that "our aim is to depict actual behavior, substantial portions of which are guided by professional judgment." Qualitative aspects included in the model are individual work and group interactions to a limited extent. Activities such as design reviews, board reviews, proposed changes, impact analysis, and scheduling are represented in the model. The outcomes of the modeling process are listed as follows:

1. Understanding of the process being modeled by participants
2. Visualization of interrelation among process components
3. Prove modeling is a valuable communication vehicle
4. Identify problems and area of process improvements
5. Recommendations for modifications of methods and procedures and technology insertions can be made

As a result of the modeling effort, SEI identified 14 issues for possible process improvement and offered over 30 recommendations for modifications to methods, procedures and technology usage. The sponsor of the effort, Mr. Barry Corson, of the Navy Air System Command stated "Software process modeling is an important first step toward success with applying the principles to Total Quality Management. This is the best method we've found to identify and define the processes that we are sponsoring." The author concludes the section with the hope that in the future these models will be quantified, allowing them to predict manpower requirements and schedules.

The paper concludes by stating that using an existing tool instead of building it yields rapid progress in the construction of real-world models for large scale software processes.

## **Appendix C Representation Experiments**

Several representation experiments were conducted during the development of the SECD Process Model. For representative purposes, we have included a few examples in the following sections.

The SECD Process Model must be complete, lend itself to automation, and also be changeable and understandable by a wide range of users. In order to meet these requirements, several well-known and lesser known modeling techniques were applied to the developed system engineering process.

The selection of a modeling technique is always complicated by the emergence of new techniques. In the following sections, various modeling techniques are briefly discussed. The advantages and disadvantages are explained for each one. In some cases, the modeling technique could be ruled out quickly because of obvious shortcomings. In other cases, a modeling technique was closely examined and top level models developed to fully understand and explore its potential benefits and shortcomings. Modeling techniques which could not represent known representation/notation standards (i.e., ANSI) were dismissed in an effort to develop a readable representation/notation which would be highly applicable.

### **C-1.0 IDEF (ICAM Definition Language)**

The Air Force ICAM (Integrated Computer Aided Manufacturing) program produced the first of a series of IDEF methods for the representation of system analysis, engineering and design. The IDEF method was originally developed to provide a clearer picture of how existing systems performed, allowing better communication among those people responsible for the integration of the systems. IDEF<sub>0</sub> is a method for describing a process from a functional point of view. The functionality of a process or system and the relationships between functions in the form of inputs, outputs, control and mechanism are defined and successively decomposed into greater detail in a series of diagrams with a rigorously defined syntax. The IDEF<sub>1</sub> method was developed to provide users with a clear representation of how the data and information important to a process can be managed to accomplish the process goals. The IDEF<sub>2</sub> method was developed to represent the behavior of process resources as a function of time suitable for mathematical modeling to be employed in simulations.

The IDEF methods have been widely employed to provide detailed descriptions of many types of process and systems. The use of IDEF to provide a complete, simulatable description of an all encompassing process such as "system engineering" would require the development of three separate models for

functions, data, and behavior. The size of the models would quickly become unmanageable and increasingly harder to comprehend. The final iterative version of the model could not have been possible with the IDEF tool, although it is certainly feasible to represent the model using any of the IDEF tools.

## **C-2.0 MacDraw**

The Apple Macintosh software tool called "MacDraw" was used to represent a data model view and a functional view of the AFR 800-14. The representations however lacked the ability to hierarchically decompose the processes. Both efforts produced models which were extremely complicated and difficult to read; therefore, it was decided that the MacDraw tool could not support useful representations.

### **C-2.0.1 AFR 800-14 Air Force Organization Representation**

This representation model was developed to gain understanding of the Air Force organization as specified in AFR 800-14. A modeling technique called "Delta Charts" was used to represent an integrated data view of the Process Model. The Delta Charts representation consists of a box representing the function and a rectangle representing an organization performing the function. The integrated data view consists of a composite view of system engineering standards which are represented in a flow chart. The Delta Chart representation method was used to ascertain the usability of these representations for the SECD Process Model.

The Delta Chart representation is similar to a functional block representation, but uses different semantics. In a complete representation, it labels the upper part of an activity with the name of the organization responsible for that activity. In this manner activities are correlated to organizations in a functional view.

### **C-2.0.2 AFR 800-14 Standard Process Model**

This representation consisted of a functional block diagram of each of the phases of the system life cycle as described by AFR 800-14. In this model, the effort is to identify activities and data flows. This standard provides a detailed description of the process, activities, and flows. Many of the processes in the AFR 800-14 are referenced in the SECD Process Model. Section Two of this document provides a summary of this standard as part of a precedence list.

This functional representation unveiled numerous work products as well as commonly used data items. Conflicts between DoDI 5000.2 and AFR 800-14 were also discovered. For example, the label names are not consistent with each other and activities overlap in many cases. Both of these documents are acquisition standards, but AFR 800-14 has more developmental information.

The overall organization of this representation is input/process/output. A series of drawings were developed separating each phase of the system life cycle for clarity and readability. The major drawback of this model is that it lacks cohesiveness with other standards.

### **C-2.0.3 MIL-STD-499A Standard Model**

A functional view of MIL-STD-499A was developed using MacDraw or COTS Apple Macintosh applications. The MIL-STD-499A standard model representation technique is the functional block diagram. This diagram provides information about the system engineering process over the entire system life cycle. The majority of the information in MIL-STD-499A is generic in nature and difficult to implement. Tailorable examples of data items are provided for implementation support.

This data model provides insight to the system engineering process activities. It is divided into life cycle phases and its name labels are consistent with other standards. Each phase of the life cycle has been represented separately to ensure clarity and readability. Section Two of this document provides a summary of this standard as part of a precedence list.

This representation's major drawback is the lack of inclusion of acquisition process, activities, and data flows. Its strength is in the management of the engineering process, not the development of engineering work products. This model as well as the AFR 800-14 model provided the means to organize a large amount of information into a graphical form. From this standpoint, this data model served its purpose for the final abstraction of the SECD Process Model.

### **C-3.0 i-Logix STATEMATE**

STATEMATE, in reality, is an automated tool which supports a methodology developed for the analysis and design of real time, embedded computer systems. This methodology is called Embedded Computer Systems Analysis Method (ECSAM). Dr. Mark I Kellner, Software Engineering Institute, Carnegie Mellon University developed a method to represent software processes using STATEMATE. Other users have found that the STATEMATE tool is readily adaptable for modeling any type of system including organizations, processes, environments, as well as real time, embedded computer system.

A STATEMATE model is comprised of three separate views of a system (functional, physical, and behavioral), each with a rigid set of syntax rules. These views of the system can be successively decomposed into greater levels of detail. The separate views are required to be consistent through the use of data flow and control logic which describes the operation of the system. The resulting

STATEMATE model can be simulated interactively or in batch mode on a desktop computer.

The STATEMATE tool is a very powerful method of presenting and simulating a process model. The cost involved in utilizing STATEMATE with regards to the learning curve for modeling and model interpretation and initial purchase of the capability essentially eliminated this modeling method for the SECD Program. However, the STATEMATE tool would be useful for the verification and validation of a developed SECD model. STATEMATE's syntax, data, behavior, and function error checking capabilities along with the simulation capability could be employed to verify and ensure consistency in the SECD Model. The major difficulty with STATEMATE and other tools in this category was the readability of the model by common users. STATEMATE did not meet our readability or understandability requirements.

It is our belief that STATEMATE and other similar tools can be used to develop an enactable engine to support the infrastructure of any process. Due to semantics, STATEMATE should be used as an engine to a highly graphical interface.