ENGINEERING-PDH.com
ONLINE CONTINUING EDUCATION

FUNDAMENTALS OF SYSTEMS ENGINEERING PART 1

Main Category:	Project Management
Sub Category:	Systems Engineering
Course #:	PRJ-116
Course Content:	63 pgs
PDH/CE Hours:	5

OFFICIAL COURSE/EXAM

(SEE INSTRUCTIONS ON NEXT PAGE)

WWW.ENGINEERING-PDH.COM
TOLL FREE (US & CA): 1-833-ENGR-PDH (1-833-364-7734)
SUPPORT@ENGINEERING-PDH.COM

PRJ-116 EXAM PREVIEW

- TAKE EXAM! -

Instructions:

- At your convenience and own pace, review the course material below. When ready, click "Take Exam!" above to complete the live graded exam. (Note it may take a few seconds for the link to pull up the exam.) You will be able to re-take the exam as many times as needed to pass.
- Upon a satisfactory completion of the course exam, which is a score of 70% or better, you will be provided with your course completion certificate. Be sure to download and print your certificates to keep for your records.

Exam Preview:

1. According to the reference material, systems engineering is an interdisciplinary engineering management process that evolves and verifies an integrated, life-cycle balanced set of system solutions that satisfy customer needs.
    a. True
    b. False
2. Which of the following is NOT a major activity of Systems Engineering Management, according to the reference material?
    a. Development Phasing
    b. Field Implementation
    c. Life cycle Integration
    d. Systems Engineering Process
3. According to the reference material, there are 8 Primary Life Cycle Functions, which of the following functions is defined as "the activities necessary to provide operations support, maintenance, logistics, and material management."
    a. Support
    b. Training
    c. Verification
    d. Operation
4. According to Chapter 3 of the reference material, during which step of the Systems Engineering Process does one conduct Trade-Off studies?
    a. Verification
    b. Requirements Analysis
    c. Functional Analysis
    d. Systems Analysis and Control
5. Figure 4-2 shows 3 "inputs" that factor into requirements analysis, which of the following "descriptions" belongs to the controls input.

a. System analysis and control
b. Tech base and other constraints
c. Customer requirements
d. Multi-disciplinary product teams

6. According to the reference material, a one-line diagram is used to depict the hardware and software components and their interrelationships.
    a. True
    b. False
7. Figure 7-1 shows the Systems Engineering and Verification process, which of the following reviews must be completed before moving on to the "Fabricate, Integrate and Test" phase?
    a. PDR
    b. SVR
    c. CDR
    d. SFR
8. According to the reference material, one primary difference between Development Test and Operation Tests is that Operation tests are controlled by an independent agency.
    a. True
    b. False
9. Figure 8-3 outlines the different types of specifications for Program-Unique Specifications. Which category does the content "defines performance characteristics of CIs and CSCIs." Belong to?
    a. Item Detail Spec
    b. Process Spec
    c. System Spec
    d. Item Performance Spec
10. According to the reference material, the IEEE/EIA 12205, Software Life Cycle Processes, describes the U.S. implementation of the ISO standard on software processes.
    a. True
    b. False

PART 2

THE SYSTEMS ENGINEERING PROCESS

CHAPTER 1

INTRODUCTION TO SYSTEMS ENGINEERING MANAGEMENT

1.1 PURPOSE

The overall organization of this text is described in the Preface. This chapter establishes some of the basic premises that are expanded throughout the book. Basic terms explained in this chapter are the foundation for following definitions. Key systems engineering ideas and viewpoints are presented, starting with a definition of a system.

1.2 DEFINITIONS

A System Is …

Simply stated, a system is an integrated composite of people, products, and processes that provide a capability to satisfy a stated need or objective.

Systems Engineering Is…

Systems engineering consists of two significant disciplines: the technical knowledge domain in which the systems engineer operates, and systems engineering management. This book focuses on the process of systems engineering management.

Three commonly used definitions of systems engineering are provided by the best known technical standards that apply to this subject. They all have a common theme:

- A logical sequence of activities and decisions that transforms an operational need into a description of system performance parameters and a preferred system configuration. (MIL-STD-499A, *Engineering Management*, 1 May 1974. Now cancelled.)

- An interdisciplinary approach that encompasses the entire technical effort, and evolves into and verifies an integrated and life cycle balanced set of system people, products, and process solutions that satisfy customer needs. (EIA Standard IS-632, *Systems Engineering*, December 1994.)

- An interdisciplinary, collaborative approach that derives, evolves, and verifies a life-cycle balanced system solution which satisfies customer expectations and meets public acceptability. (IEEE P1220, *Standard for Application and Management of the Systems Engineering Process*, [Final Draft], 26 September 1994.)

In summary, systems engineering is an interdisciplinary engineering management process that evolves and verifies an integrated, life-cycle balanced set of system solutions that satisfy customer needs.

Systems Engineering Management Is…

As illustrated by Figure 1-1, systems engineering management is accomplished by integrating three major activities:

- Development phasing that controls the design process and provides baselines that coordinate design efforts,

- A systems engineering process that provides a structure for solving design problems and

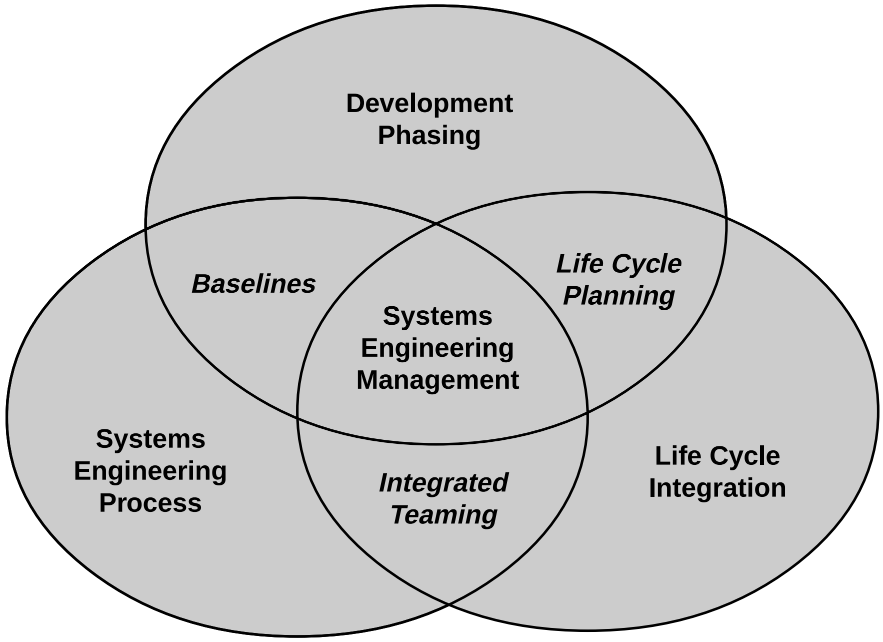

Figure 1-1. Three Activities of Systems Engineering Management

tracking requirements flow through the design effort, and

- Life cycle integration that involves customers in the design process and ensures that the system developed is viable throughout its life.

Each one of these activities is necessary to achieve proper management of a development effort. Phasing has two major purposes: it controls the design effort and is the major connection between the technical management effort and the overall acquisition effort. It controls the design effort by developing design baselines that govern each level of development. It interfaces with acquisition management by providing key events in the development process, where design viability can be assessed. The viability of the baselines developed is a major input for acquisition management Milestone (MS) decisions. As a result, the timing and coordination between technical development phasing and the acquisition schedule is critical to maintain a healthy acquisition program.

The systems engineering process is the heart of systems engineering management. Its purpose is to provide a structured but flexible process that transforms requirements into specifications, architectures, and configuration baselines. The discipline of this process provides the control and traceability to develop solutions that meet customer needs. The systems engineering process may be repeated one or more times during any phase of the development process.

Life cycle integration is necessary to ensure that the design solution is viable throughout the life of the system. It includes the planning associated with product and process development, as well as the integration of multiple functional concerns into the design and engineering process. In this manner, product cycle-times can be reduced, and the need for redesign and rework substantially reduced.

1.3 DEVELOPMENT PHASING

Development usually progresses through distinct levels or stages:

- Concept level, which produces a system concept description (usually described in a concept study);

- System level, which produces a system description in performance requirement terms; and

- Subsystem/Component level, which produces first a set of subsystem and component product performance descriptions, then a set of corresponding detailed descriptions of the products' characteristics, essential for their production.

The systems engineering process is applied to each level of system development, one level at a time, to produce these descriptions commonly called configuration baselines. This results in a series of configuration baselines, one at each development level. These baselines become more detailed with each level.

In the Department of Defense (DoD) the configuration baselines are called the functional baseline for the system-level description, the allocated baseline for the subsystem/ component performance descriptions, and the product baseline for the subsystem/component detail descriptions. Figure 1-2 shows the basic relationships between the baselines. The triangles represent baseline control decision points, and are usually referred to as technical reviews or audits.

Levels of Development Considerations

Significant development at any given level in the system hierarchy should not occur until the configuration baselines at the higher levels are considered complete, stable, and controlled. Reviews and audits are used to ensure that the baselines are ready for the next level of development. As will be shown in the next chapter, this review and audit process also provides the necessary assessment of system maturity, which supports the DoD Milestone decision process.

1.4 THE SYSTEMS ENGINEERING PROCESS

The systems engineering process is a top-down comprehensive, iterative and recursive problem

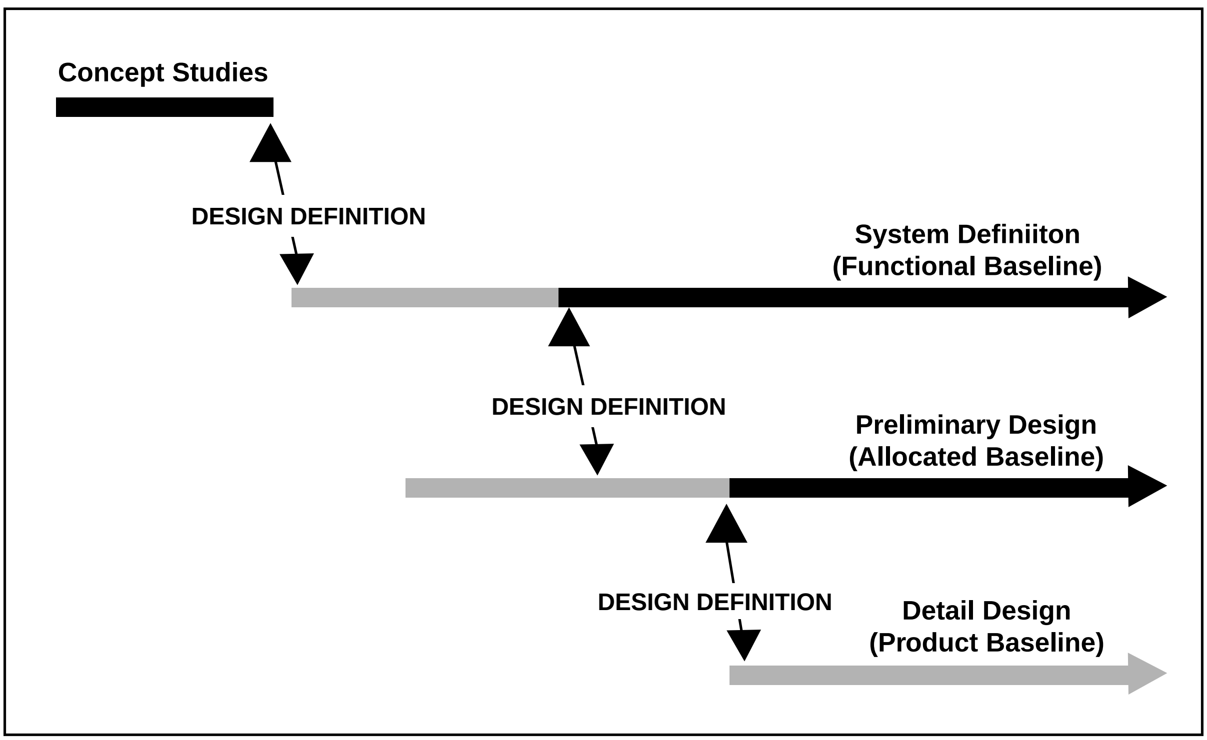

Figure 1-2. Development Phasing

solving process, applied sequentially through all stages of development, that is used to:

- Transform needs and requirements into a set of system product and process descriptions (adding value and more detail with each level of development),
- Generate information for decision makers, and
- Provide input for the next level of development.

As illustrated by Figure 1-3, the fundamental systems engineering activities are Requirements Analysis, Functional Analysis and Allocation, and Design Synthesis—all balanced by techniques and tools collectively called System Analysis and Control. Systems engineering controls are used to track decisions and requirements, maintain technical baselines, manage interfaces, manage risks, track cost and schedule, track technical performance, verify requirements are met, and review/audit the progress.

During the systems engineering process architectures are generated to better describe and understand the system. The word “architecture” is used in various contexts in the general field of engineering. It is used as a general description of how the subsystems join together to form the system. It can also be a detailed description of an aspect of a system: for example, the Operational, System, and Technical Architectures used in Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR), and software intensive developments. However, Systems Engineering Management as developed in DoD recognizes three universally usable architectures that describe important aspects of the system: functional, physical, and system architectures. This book will focus on these architectures as necessary components of the systems engineering process.

The *Functional Architecture* identifies and structures the allocated functional and performance requirements. The *Physical Architecture* depicts the

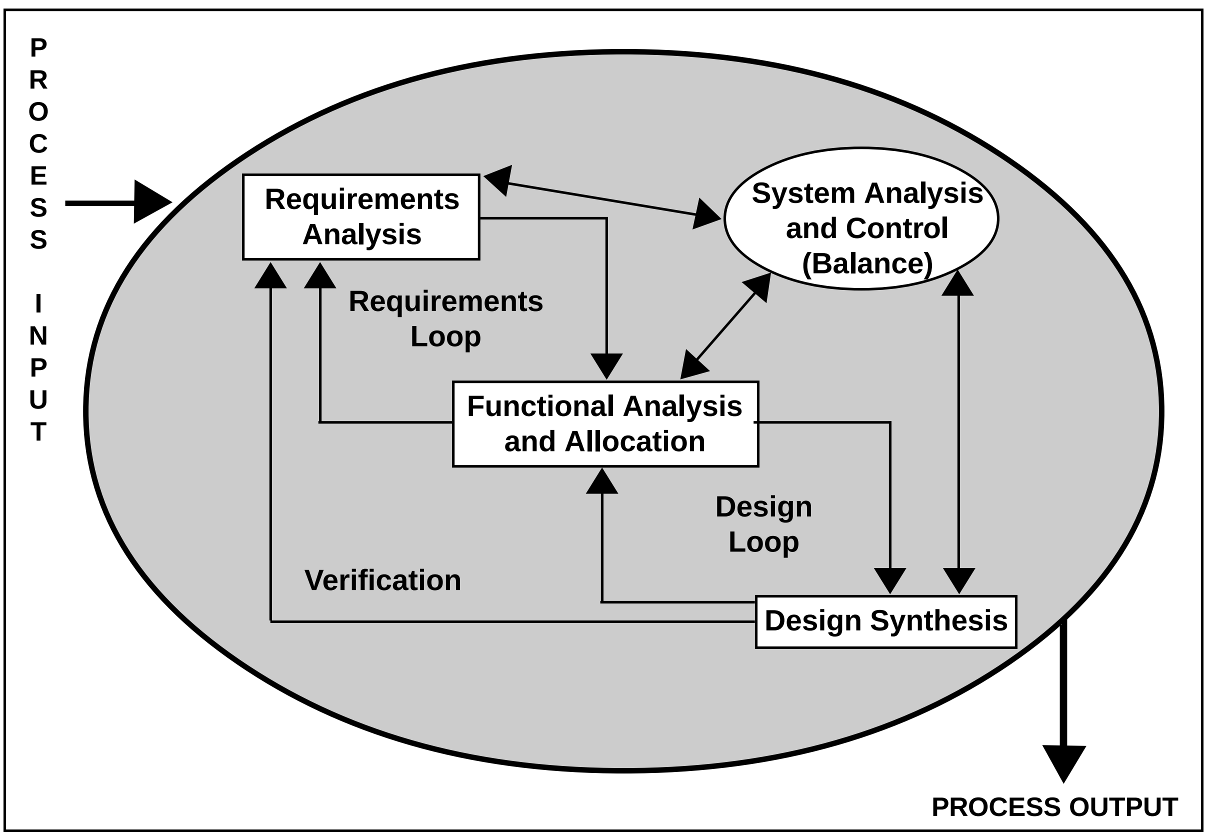

Figure 1-3. The Systems Engineering Process

system product by showing how it is broken down into subsystems and components. The *System Architecture* identifies all the products (including enabling products) that are necessary to support the system and, by implication, the processes necessary for development, production/construction, deployment, operations, support, disposal, training, and verification.

Life Cycle Integration

Life cycle integration is achieved through integrated development—that is, concurrent consideration of all life cycle needs during the development process. DoD policy requires integrated development, called Integrated Product and Product Development (IPPD) in DoD, to be practiced at all levels in the acquisition chain of command as will be explained in the chapter on IPPD. Concurrent consideration of all life cycle needs can be greatly enhanced through the use of interdisciplinary teams. These teams are often referred to as Integrated Product Teams (IPTs).

The objective of an Integrated Product Team is to:

- Produce a design solution that satisfies initially defined requirements, and
- Communicate that design solution clearly, effectively, and in a timely manner.

Multi-functional, integrated teams:

- Place balanced emphasis on product and process development, and
- Require early involvement of all disciplines appropriate to the team task.

Design-level IPT members are chosen to meet the team objectives and generally have distinctive competence in:

- Technical management (systems engineering),
- Life cycle functional areas (eight primary functions),
- Technical specialty areas, such as safety, risk management, quality, etc., or
- When appropriate, business areas such as finance, cost/budget analysis, and contracting.

Life Cycle Functions

Life cycle functions are the characteristic actions associated with the system life cycle. As illustrated by Figure 1-4, they are development, production and construction, deployment (fielding), operation, support, disposal, training, and verification. These activities cover the “cradle to grave” life cycle process and are associated with major functional groups that provide essential support to the life cycle process. These key life cycle functions are commonly referred to as the eight primary functions of systems engineering.

The customers of the systems engineer perform the life-cycle functions. The system user’s needs are emphasized because their needs generate the requirement for the system, but it must be remembered that all of the life-cycle functional areas generate requirements for the systems engineering process once the user has established the basic need. ***Those that perform the primary functions also provide life-cycle representation in design-level integrated teams.***

Primary Function Definitions

Development includes the activities required to evolve the system from customer needs to product or process solutions.

Manufacturing/Production/Construction includes the fabrication of engineering test models and “brass boards,” low rate initial production, full-rate production of systems and end items, or the construction of large or unique systems or subsystems.

Deployment (Fielding) includes the activities necessary to initially deliver, transport, receive, process, assemble, install, checkout, train, operate, house, store, or field the system to achieve full operational capability.

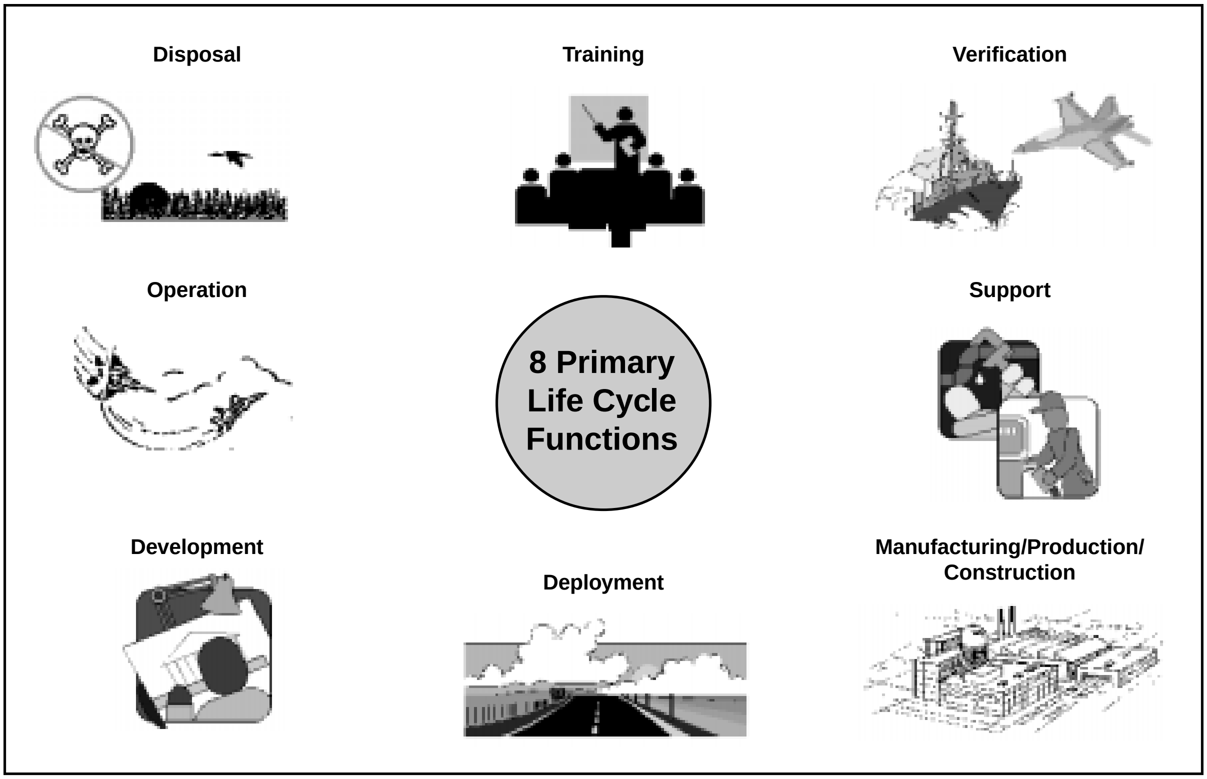

Figure 1-4. Primary Life Cycle Functions

Operation is the user function and includes activities necessary to satisfy defined operational objectives and tasks in peacetime and wartime environments.

Support includes the activities necessary to provide operations support, maintenance, logistics, and material management.

Disposal includes the activities necessary to ensure that the disposal of decommissioned, destroyed, or irreparable system components meets all applicable regulations and directives.

Training includes the activities necessary to achieve and maintain the knowledge and skill levels necessary to efficiently and effectively perform operations and support functions.

Verification includes the activities necessary to evaluate progress and effectiveness of evolving system products and processes, and to measure specification compliance.

Systems Engineering Considerations

Systems engineering is a standardized, disciplined management process for development of system solutions that provides a constant approach to system development in an environment of change and uncertainty. It also provides for simultaneous product and process development, as well as a common basis for communication.

Systems engineering ensures that the correct technical tasks get done during development through planning, tracking, and coordinating. Responsibilities of systems engineers include:

- Development of a total system design solution that balances cost, schedule, performance, and risk,
- Development and tracking of technical information needed for decision making,
- Verification that technical solutions satisfy customer requirements,

- Development of a system that can be produced economically and supported throughout the life cycle,
- Development and monitoring of internal and external interface compatibility of the system and subsystems using an open systems approach,
- Establishment of baselines and configuration control, and
- Proper focus and structure for system and major sub-system level design IPTs.

1.5 GUIDANCE

DoD 5000.2-R establishes two fundamental requirements for program management:

- It requires that an Integrated Product and Process approach be taken to design wherever practicable, and
- It requires that a disciplined systems engineering process be used to translate operational needs and/or requirements into a system solution.

Tailoring the Process

System engineering is applied during all acquisition and support phases for large- and small-scale systems, new developments or product improvements, and single and multiple procurements. The process must be tailored for different needs and/or requirements. Tailoring considerations include system size and complexity, level of system definition detail, scenarios and missions, constraints and requirements, technology base, major risk factors, and organizational best practices and strengths.

For example, systems engineering of software should follow the basic systems engineering approach as presented in this book. However, it must be tailored to accommodate the software development environment, and the unique progress tracking and verification problems software development entails. In a like manner, all technology domains are expected to bring their own unique needs to the process.

This book provides a conceptual-level description of systems engineering management. The specific techniques, nomenclature, and recommended methods are not meant to be prescriptive. Technical managers must tailor their systems engineering planning to meet their particular requirements and constraints, environment, technical domain, and schedule/budget situation.

However, the basic time-proven concepts inherent in the systems engineering approach must be retained to provide continuity and control. For complex system designs, a full and documented understanding of what the system must do should precede development of component performance descriptions, which should precede component detail descriptions. Though some parts of the system may be dictated as a constraint or interface, in general, solving the design problem should start with analyzing the requirements and determining what the system has to do before physical alternatives are chosen. Configurations must be controlled and risk must be managed.

Tailoring of this process has to be done carefully to avoid the introduction of substantial unseen risk and uncertainty. Without the control, coordination, and traceability of systems engineering, an environment of uncertainty results which will lead to surprises. Experience has shown that these surprises almost invariably lead to significant impacts to cost and schedule. Tailored processes that reflect the general conceptual approach of this book have been developed and adopted by professional societies, academia, industry associations, government agencies, and major companies.

1.6 SUMMARY POINTS

- Systems engineering management is a multi-functional process that integrates life cycle functions, the systems engineering problem-solving process, and progressive baselining.

- The systems engineering process is a problem-solving process that drives the balanced development of system products and processes.

- Integrated Product Teams should apply the systems engineering process to develop a life cycle balanced-design solution.

- The systems engineering process is applied to each level of development, one level at a time.

- Fundamental systems engineering activities are Requirements Analysis, Functional Analysis/Allocation, and Design Synthesis, all of which are balanced by System Analysis and Control.

- Baseline phasing provides for an increasing level of descriptive detail of the products and processes with each application of the systems engineering process.

- Baselining in a nut shell is a concept description that leads to a system definition which, in turn, leads to component definitions, and then to component designs, which finally lead to a product.

- The output of each application of the systems engineering process is a major input to the next process application.

CHAPTER 3

SYSTEMS ENGINEERING PROCESS OVERVIEW

3.1 THE PROCESS

The Systems Engineering Process (SEP) is a comprehensive, iterative and recursive problem solving process, applied sequentially top-down by integrated teams. It transforms needs and requirements into a set of system product and process descriptions, generate information for decision makers, and provides input for the next level of development. The process is applied sequentially, one level at a time, adding additional detail and definition with each level of development. As shown by Figure 3-1, the process includes: inputs and outputs; requirements analysis; functional analysis and allocation; requirements loop; synthesis; design loop; verification; and system analysis and control.

Systems Engineering Process Inputs

Inputs consist primarily of the customer's needs, objectives, requirements and project constraints.

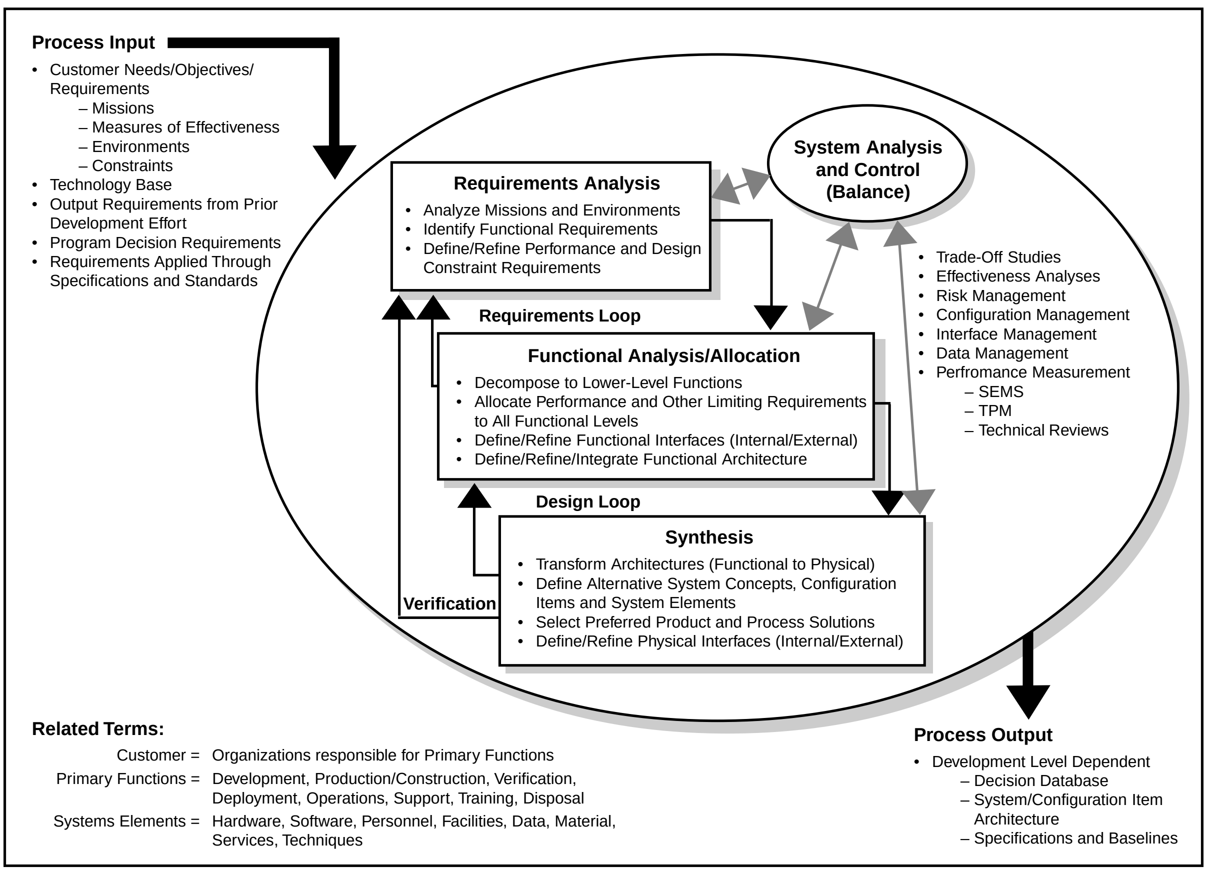

Figure 3-1. The Systems Engineering Process

Inputs can include, but are not restricted to, missions, measures of effectiveness, environments, available technology base, output requirements from prior application of the systems engineering process, program decision requirements, and requirements based on "corporate knowledge."

Requirements Analysis

The first step of the Systems Engineering Process is to analyze the process inputs. Requirements analysis is used to develop functional and performance requirements; that is, customer requirements are translated into a set of requirements that define what the system must do and how well it must perform. The systems engineer must ensure that the requirements are understandable, unambiguous, comprehensive, complete, and concise.

Requirements analysis must clarify and define functional requirements and design constraints. Functional requirements define quantity (how many), quality (how good), coverage (how far), time lines (when and how long), and availability (how often). Design constraints define those factors that limit design flexibility, such as: environmental conditions or limits; defense against internal or external threats; and contract, customer or regulatory standards.

Functional Analysis/Allocation

Functions are analyzed by decomposing higher-level functions identified through requirements analysis into lower-level functions. The performance requirements associated with the higher level are allocated to lower functions. The result is a description of the product or item in terms of what it does logically and in terms of the performance required. This description is often called the functional architecture of the product or item. Functional analysis and allocation allows for a better understanding of what the system has to do, in what ways it can do it, and to some extent, the priorities and conflicts associated with lower-level functions. It provides information essential to optimizing physical solutions. Key tools in functional analysis and allocation are Functional Flow Block Diagrams, Time Line Analysis, and the Requirements Allocation Sheet.

Requirements Loop

Performance of the functional analysis and allocation results in a better understanding of the requirements and should prompt reconsideration of the requirements analysis. Each function identified should be traceable back to a requirement. This iterative process of revisiting requirements analysis as a result of functional analysis and allocation is referred to as the requirements loop.

Design Synthesis

Design synthesis is the process of defining the product or item in terms of the physical and software elements which together make up and define the item. The result is often referred to as the physical architecture. Each part must meet at least one functional requirement, and any part may support many functions. The physical architecture is the basic structure for generating the specifications and baselines.

Design Loop

Similar to the requirements loop described above, the design loop is the process of revisiting the functional architecture to verify that the physical design synthesized can perform the required functions at required levels of performance. The design loop permits reconsideration of how the system will perform its mission, and this helps optimize the synthesized design.

Verification

For each application of the system engineering process, the solution will be compared to the requirements. This part of the process is called the verification loop, or more commonly, Verification. Each requirement at each level of development must be verifiable. Baseline documentation developed during the systems engineering process must establish the method of verification for each requirement.

Appropriate methods of verification include examination, demonstration, analysis (including modeling and simulation), and testing. Formal test and evaluation (both developmental and operational) are important contributors to the verification of systems.

Systems Analysis and Control

Systems Analysis and Control include technical management activities required to measure progress, evaluate and select alternatives, and document data and decisions. These activities apply to all steps of the sysems engineering process.

System analysis activities include trade-off studies, effectiveness analyses, and design analyses. They evaluate alternative approaches to satisfy technical requirements and program objectives, and provide a rigorous quantitative basis for selecting performance, functional, and design requirements. Tools used to provide input to analysis activities include modeling, simulation, experimentation, and test.

Control activities include risk management, configuration management, data management, and performance-based progress measurement including event-based scheduling, Technical Performance Measurement (TPM), and technical reviews.

The purpose of Systems Analysis and Control is to ensure that:

- Solution alternative decisions are made only after evaluating the impact on system effectiveness, life cycle resources, risk, and customer requirements,
- Technical decisions and specification requirements are based on systems engineering outputs,
- Traceability from systems engineering process inputs to outputs is maintained,
- Schedules for development and delivery are mutually supportive,
- Required technical disciplines are integrated into the systems engineering effort,
- Impacts of customer requirements on resulting functional and performance requirements are examined for validity, consistency, desirability, and attainability, and,
- Product and process design requirements are directly traceable to the functional and performance requirements they were designed to fulfill, and vice versa.

Systems Engineering Process Output

Process output is dependent on the level of development. It will include the decision database, the system or configuration item architecture, and the baselines, including specifications, appropriate to the phase of development. In general, it is any data that describes or controls the product configuration or the processes necessary to develop that product.

3.2 SUMMARY POINTS

- The system engineering process is the engine that drives the balanced development of system products and processes applied to each level of development, one level at a time.
- The process provides an increasing level of descriptive detail of products and processes with each system engineering process application. The output of each application is the input to the next process application.

CHAPTER 4

REQUIREMENTS ANALYSIS

4.1 SYSTEMS ENGINEERING PROCESS INPUTS

The inputs to the process include the customer's requirements and the project constraints. Requirements relate directly to the performance characteristics of the system being designed. They are the stated life-cycle customer needs and objectives for the system, and they relate to how well the system will work in its intended environment.

Constraints are conditions that exist because of limitations imposed by external interfaces, project support, technology, or life cycle support systems. Constraints bound the development teams' design opportunities.

Requirements are the primary focus in the systems engineering process because the process's primary purpose is to transform the requirements into designs. The process develops these designs within the constraints. They eventually must be verified to meet both the requirements and constraints.

Types of Requirements

Requirements are categorized in several ways. The following are common categorizations of requirements that relate to technical management:

Customer Requirements: Statements of fact and assumptions that define the expectations of the system in terms of mission objectives, environment, constraints, and measures of effectiveness and suitability (MOE/MOS). The customers are those that perform the eight primary functions of systems engineering (Chapter 1), with special emphasis on the operator as the key customer. Operational requirements will define the basic need and, at a minimum, answer the questions posed in Figure 4-1.

Operational distribution or deployment: Where will the system be used?

Mission profile or scenario: How will the system accomplish its mission objective?

Performance and related parameters: What are the critical system parameters to accomplish the mission?

Utilization environments: How are the various system components to be used?

Effectiveness requirements: How effective or efficient must the system be in performing its mission?

Operational life cycle: How long will the system be in use by the user?

Environment: What environments will the system be expected to operate in an effective manner?

Figure 4-1. Operational Requirements – Basic Questions

Functional Requirements: The necessary task, action or activity that must be accomplished. Functional (what has to be done) requirements identified in requirements analysis will be used as the top-level functions for functional analysis.

Performance Requirements: The extent to which a mission or function must be executed; generally measured in terms of quantity, quality, coverage, timeliness or readiness. During requirements analysis, performance (how well does it have to be done) requirements will be interactively developed across all identified functions based on system life cycle factors; and characterized in terms of the degree of certainty in their estimate, the degree of criticality to system success, and their relationship to other requirements.

Design Requirements: The "build to," "code to," and "buy to" requirements for products and "how to execute" requirements for processes expressed in technical data packages and technical manuals.

Derived Requirements: Requirements that are implied or transformed from higher-level requirement. For example, a requirement for long range or high speed may result in a design requirement for low weight.

Allocated Requirements: A requirement that is established by dividing or otherwise allocating a high-level requirement into multiple lower-level requirements. Example: A 100-pound item that consists of two subsystems might result in weight requirements of 70 pounds and 30 pounds for the two lower-level items.

Attributes of Good Requirements

The attributes of good requirements include the following:

- A requirement must be achievable. It must reflect need or objective for which a solution is technically achievable at costs considered affordable.
- It must be verifiable—that is, not defined by words such as excessive, sufficient, resistant, etc. The expected performance and functional utility must be expressed in a manner that allows verification to be objective, preferably quantitative.
- A requirement must be unambiguous. It must have but one possible meaning.
- It must be complete and contain all mission profiles, operational and maintenance concepts, utilization environments and constraints. All information necessary to understand the customer's need must be there.
- It must be expressed in terms of need, not solution; that is, it should address the "why" and "what" of the need, not how to do it.
- It must be consistent with other requirements. Conflicts must be resolved up front.
- It must be appropriate for the level of system hierarchy. It should not be too detailed that it constrains solutions for the current level of design. For example, detailed requirements relating to components would not normally be in a system-level specification.

4.2 REQUIREMENTS ANALYSIS

Requirements analysis involves defining customer needs and objectives in the context of planned customer use, environments, and identified system characteristics to determine requirements for system functions. Prior analyses are reviewed and updated, refining mission and environment definitions to support system definition.

Requirements analysis is conducted iteratively with functional analysis to optimize performance requirements for identified functions, and to verify that synthesized solutions can satisfy customer requirements. The purpose of Requirements Analysis is to:

- Refine customer objectives and requirements;
- Define initial performance objectives and refine them into requirements;
- Identify and define constraints that limit solutions; and
- Define functional and performance requirements based on customer provided measures of effectiveness.

In general, Requirements Analysis should result in a clear understanding of:

- Functions: What the system has to do,
- Performance: How well the functions have to be performed,
- Interfaces: Environment in which the system will perform, and
- Other requirements and constraints.

The understandings that come from requirements analysis establish the basis for the functional and physical designs to follow. Good requirements analysis is fundamental to successful design definition.

Inputs

Typical inputs include customer needs and objectives, missions, MOE/MOS, environments, key performance parameters (KPPs), technology base, output requirements from prior application of SEP, program decision requirements, and suitability requirements. (See Figure 4-2 for additional considerations.)

Input requirements must be comprehensive and defined for both system products and system processes such as development, manufacturing, verification, deployment, operations, support, training and disposal (eight primary functions).

Role of Integrated Teams

The operator customers have expertise in the operational employment of the product or item being developed. The developers (government and contractor) are not necessarily competent in the operational aspects of the system under development. Typically, the operator's need is neither clearly nor completely expressed in a way directly

- **Inputs converted to putputs:**
  - Customer requirements
  - Mission and MOEs (MNS, ORD)
  - Maintenance concept and other life-cycle function planning
  - SE outputs from prior development efforts
- **Controls:**
  - Laws and organizational policies and procedures
  - Military specific requirements
  - Utilization environments
  - Tech base and other constraints
- **Enablers:**
  - Multi-disciplinary product teams
  - Decision and requirements database including system/configuration item descriptions from prior efforts
  - System analysis and control

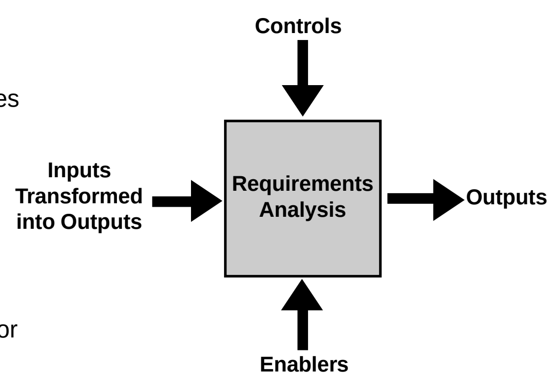

Figure 4-2. Inputs to Requirements Analysis

usable by developers. It is unlikely that developers will receive a well-defined problem from which they can develop the system specification. Thus, teamwork is necessary to understand the problem and to analyze the need. It is imperative that customers are part of the definition team.

On the other hand, customers often find it easier to describe a system that attempts to solve the problem rather than to describe the problem itself. Although these "solutions" may be workable to some extent, the optimum solution is obtained through a proper technical development effort that properly balances the various customer mission objectives, functions, MOE/MOS, and constraints. An integrated approach to product and process development will balance the analysis of requirements by providing understanding and accommodation among the eight primary functions.

Requirements Analysis Questions

Requirements Analysis is a process of inquiry and resolution. The following are typical questions that can initiate the thought process:

- What are the reasons behind the system development?
- What are the customer expectations?
- Who are the users and how do they intend to use the product?
- What do the users expect of the product?
- What is their level of expertise?
- With what environmental characteristics must the system comply?
- What are existing and planned interfaces?
- What functions will the system perform, expressed in customer language?
- What are the constraints (hardware, software, economic, procedural) to which the system must comply?
- What will be the final form of the product: such as model, prototype, or mass production?

This list can start the critical, inquisitive outlook necessary to analyze requirements, but it is only the beginning. A tailored process similar to the one at the end of this chapter must be developed to produce the necessary requirements analysis outputs.

4.3 REQUIREMENTS ANALYSIS OUTPUTS

The requirements that result from requirements analysis are typically expressed from one of three perspectives or views. These have been described as the Operational, Functional, and Physical views. All three are necessary and must be coordinated to fully understand the customers' needs and objectives. All three are documented in the decision database.

Operational View

The Operational View addresses how the system will serve its users. It is useful when establishing requirements of "how well" and "under what condition." Operational view information should be documented in an operational concept document that identifies:

- Operational need definition,
- System mission analysis,
- Operational sequences,
- Operational environments,
- Conditions/events to which a system must respond,
- Operational constraints on system,
- Mission performance requirements,
- User and maintainer roles (defined by job tasks and skill requirements or constraints),

- Structure of the organizations that will operate, support and maintain the system, and

- Operational interfaces with other systems.

Analyzing requirements requires understanding the operational and other life cycle needs and constraints.

Functional View

The Functional View focuses on WHAT the system must do to produce the required operational behavior. It includes required inputs, outputs, states, and transformation rules. The functional requirements, in combination with the physical requirements shown below, are the primary sources of the requirements that will eventually be reflected in the system specification. Functional View information includes:

- System functions,

- System performance,
  - Qualitative — how well
  - Quantitative — how much, capacity
  - Timeliness — how often

- Tasks or actions to be performed,

- Inter-function relationships,

- Hardware and software functional relationships,

- Performance constraints,

- Interface requirements including identification of potential open-system opportunities (potential standards that could promote open systems should be identified),

- Unique hardware or software, and

- Verification requirements (to include inspection, analysis/simulation, demo, and test).

Physical View

The Physical View focuses on HOW the system is constructed. It is key to establishing the physical interfaces among operators and equipment, and technology requirements. Physical View information would normally include:

- Configuration of System:
  - Interface descriptions,
  - Characteristics of information displays and operator controls,
  - Relationships of operators to system/ physical equipment, and
  - Operator skills and levels required to perform assigned functions.

- Characterization of Users:
  - Handicaps (special operating environments), and
  - Constraints (movement or visual limitations).

- System Physical Limitations:
  - Physical limitations (capacity, power, size, weight),
  - Technology limitations (range, precision, data rates, frequency, language),
  - Government Furinished Equipment (GFE), Commercial-Off-the-Shelf (COTS), Nondevelopmental Item (NDI), reusability requirements, and
  - Necessary or directed standards.

4.4 SUMMARY POINTS

- An initial statement of a need is seldom defined clearly.

- A significant amount of collaboration between various life cycle customers is necessary to produce an acceptable requirements document.

- Requirements are a statement of the problem to be solved. Unconstrained and nonintegrated requirements are seldom sufficient for designing a solution.

- Because requirements from different customers will conflict, constraints will limit options, and resources are not unlimited; trade studies must be accomplished in order to select a balanced set of requirements that provide feasible solutions to customer needs.

SUPPLEMENT 4-A

A PROCEDURE FOR REQUIREMENTS ANALYSIS

The following section provides a list of tasks that represents a plan to analyze requirements. Part of this notional process is based on the 15 requirements analysis tasks listed in IEEE P1220. This industry standard and others should be consulted when preparing engineering activities to help identify and structure appropriate activities.

As with all techniques, the student should be careful to tailor; that is, add or subtract, as suits the particular system being developed. Additionally, these tasks, though they build on each other, should not be considered purely sequential. Every task contributes understanding that may cause a need to revisit previous task decisions. This is the nature of all System Engineering activities.

Preparation: Establish and Maintain Decision Database

When beginning a systems engineering process, be sure that a system is in place to record and manage the decision database. The decision database is an historical database of technical decisions and requirements for future reference. It is the primary means for maintaining requirements traceability. This database decision management system must be developed or the existing system must be reviewed and upgraded as necessary to accommodate the new stage of product development. A key part of this database management system is a Requirements Traceability Matrix that maps requirements to subsystems, configuration items, and functional areas.

This must be developed, updated, and reissued on a regular basis. All requirements must be recorded. ***Remember: If it is not recorded, it cannot be an approved requirement!***

The 15 Tasks of IEEE P1220

The IEEE Systems Engineering Standard offers a process for performing Requirements Analysis that comprehensively identifies the important tasks that must be performed. These 15 task areas to be analyzed follow and are shown in Figure 4-3.

1. **Customer expectations**
2. **Project and enterprise constraints**
3. **External constraints**
4. **Operational scenarios**
5. **Measure of effectiveness (MOEs)**
6. **System boundaries**
7. **Interfaces**
8. **Utilization environments**
9. **LIfe cycle**
10. **Functional requirements**
11. **Performance requirements**
12. **Modes of operation**
13. **Technical performance measures**
14. **Physical characteristics**
15. **Human systems integration**

Figure 4-3. IEEE P1220 Requirements Analysis Task Areas

Task 1. Customer Expectations

Define and quantify customer expectations. They may come from any of the eight primary functions, operational requirements documents, mission needs, technology-based opportunity, direct communications with customer, or requirements from a higher system level. The purpose of this task is to determine what the customer wants the system to accomplish, and how well each function must be accomplished. This should include natural and induced environments in which the product(s) of the system must operate or be used, and constraints (e.g. funding, cost, or price objectives, schedule, technology, nondevelopmental and reusable items, physical characteristics, hours of operation per day, on-off sequences, etc.).

Task 2. Project and Enterprise Constraints

Identify and define constraints impacting design solutions. Project specific constraints can include:

- Approved specifications and baselines developed from prior applications of the Systems Engineering Process,
- Costs,
- Updated technical and project plans,
- Team assignments and structure,
- Control mechanisms, and
- Required metrics for measuring progress.

Enterprise constraints can include:

- Management decisions from a preceding technical review,
- Enterprise general specifications,
- Standards or guidelines,
- Policies and procedures,
- Domain technologies, and
- Physical, financial, and human resource allocations to the project.

Task 3. External Constraints

Identify and define external constraints impacting design solutions or implementation of the Systems Engineering Process activities. External constraints can include:

- Public and international laws and regulations,
- Technology base,
- Compliance requirements: industry, international, and other general specifications, standards, and guidelines which require compliance for legal, interoperability, or other reasons,
- Threat system capabilities, and
- Capabilities of interfacing systems.

Task 4. Operational Scenarios

Identify and define operational scenarios that scope the anticipated uses of system product(s). For each operational scenario, define expected:

- Interactions with the environment and other systems, and
- Physical interconnectivities with interfacing systems, platforms, or products.

Task 5. Measures of Effectiveness and Suitability (MOE/MOS)

Identify and define systems effectiveness measures that reflect overall customer expectations and satisfaction. MOEs are related to how well the system must perform the customer's mission. Key MOEs include mission performance, safety, operability, reliability, etc. MOSs are related to how well the system performs in its intended environment and includes measures of supportability, maintainability, ease of use, etc.

Task 6. System Boundaries

Define system boundaries including:

- Which system elements are under design control of the performing activity and which fall outside of their control, and
- The expected interactions among system elements under design control and external and/or higher-level and interacting systems outside the system boundary (including open systems approaches).

Task 7. Interfaces

Define the functional and physical interfaces to external or higher-level and interacting systems, platforms, and/or products in quantitative terms (include open systems approach). Functional and physical interfaces would include mechanical, electrical, thermal, data, control, procedural, and other interactions. Interfaces may also be considered from an internal/external perspective. Internal interfaces are those that address elements inside the boundaries established for the system addressed. These interfaces are generally identified and controlled by the contractor responsible for developing the system. External interfaces, on the other hand, are those which involve entity relationships outside the established boundaries, and these are typically defined and controlled by the government.

Task 8. Utilization Environments

Define the environments for each operational scenario. All environmental factors (natural or induced) which may impact system performance must be identified and defined. Environmental factors include:

- Weather conditions (e.g., rain, snow, sun, wind, ice, dust, fog),
- Temperature ranges,
- Topologies (e.g., ocean, mountains, deserts, plains, vegetation),
- Biological (e.g., animal, insects, birds, fungi),
- Time (e.g., dawn, day, night, dusk), and
- Induced (e.g., vibration, electromagnetic, chemical).

Task 9. Life Cycle Process Concepts

Analyze the outputs of tasks 1-8 to define key life cycle process requirements necessary to develop, produce, test, distribute, operate, support, train, and dispose of system products under development. Use integrated teams representing the eight primary functions. Focus should be on the cost drivers and higher risk elements that are anticipated to impact supportability and affordability over the useful life of the system.

Task 10. Functional Requirements

Define what the system must accomplish or must be able to do. Functions identified through requirements analysis will be further decomposed during functional analysis and allocation.

Task 11. Performance Requirements

Define the performance requirements for each higher-level function performed by the system. Primary focus should be placed on performance requirements that address the MOEs, and other KPPs established in test plans or identified as interest items by oversight authorities.

Task 12. Modes of Operation

Define the various modes of operation for the system products under development. Conditions (e.g., environmental, configuration, operational, etc.) that determine the modes of operation should be included in this definition.

Task 13. Technical Performance Measures (TPMs)

Identify the key indicators of system performance that will be tracked during the design process. Selection of TPMs should be limited to critical technical thresholds and goals that, if not met, put the project at cost, schedule, or performance risk. TPMs involve tracking the actual versus planned progress of KPPs such that the manager can make judgments about technical progress on a by-exception basis. To some extent TPM selection is phase dependent. They must be reconsidered at each systems engineering process step and at the beginning of each phase.

Task 14. Physical Characteristics

Identify and define required physical characteristics (e.g., color, texture, size, weight, buoyancy) for the system products under development. Identify which physical characteristics are true constraints and which can be changed, based on trade studies.

Task 15. Human Factors

Identify and define human factor considerations (e.g., physical space limits, climatic limits, eye movement, reach, ergonomics) which will affect operation of the system products under development. Identify which human systems integration are constraints and which can be changed based on trade studies.

Follow-on Tasks

The follow-on tasks are related to the iterative nature of the Systems Engineering Process:

Integrate Requirements:

Take an integrated team approach to requirements determination so that conflicts among and between requirements are resolved in ways that result in design requirements that are balanced in terms of both risk and affordability.

Validate Requirements:

During Functional Analysis and Allocation, validate that the derived functional and performance can be traced to the operational requirements.

Verify Requirements:

- Coordinate design, manufacturing, deployment and test processes,
- Ensure that requirements are achievable and testable,
- Verify that the design-to-cost goals are achievable, and
- Verify that the functional and physical architectures defined during Functional Analysis/Allocation and Synthesis meet the integrated technical, cost, and schedule requirements within acceptable levels of risk.

CHAPTER 5

FUNCTIONAL ANALYSIS AND ALLOCATION

5.1 INTRODUCTION

The purpose of this systems engineering process activity is to transform the functional, performance, interface and other requirements that were identified through requirements analysis into a coherent description of system functions that can be used to guide the Design Synthesis activity that follows. The designer will need to know what the system must do, how well, and what constraints will limit design flexibility.

This is accomplished by arranging functions in logical sequences, decomposing higher-level functions into lower-level functions, and allocating performance from higher- to lower-level functions. The tools used include functional flow block diagrams and timeline analysis; and the product is a functional architecture, i.e., a description of the system—but in terms of functions and performance parameters, rather than a physical description. Functional Analysis and Allocation facilitates traceability from requirements to the solution descriptions that are the outcome of Design Synthesis.

Functions are discrete actions (use action verbs) necessary to achieve the system's objectives. These functions may be stated explicitly, or they may be derived from stated requirements. The functions will ultimately be performed or accomplished through use of equipment, personnel, facilities, software, or a combination.

5.2 FUNCTIONAL ANALYSIS AND ALLOCATION

Functional and performance requirements at any level in the system are developed from higher-level requirements. Functional Analysis and Allocation is repeated to define successively lower-level functional and performance requirements, thus defining architectures at ever-increasing levels of detail. System requirements are allocated and defined in sufficient detail to provide design and verification criteria to support the integrated system design.

This top-down process of translating system-level requirements into detailed functional and performance design criteria includes:

- Defining the system in functional terms, then decomposing the top-level functions into subfunctions. That is, identifying at successively lower levels what actions the system has to do,
- Translating higher-level performance requirements into detailed functional and performance design criteria or constraints. That is, identifying how well the functions have to be performed,
- Identifying and defining all internal and external functional interfaces,
- Identifying functional groupings to minimize and control interfaces (functional partitioning),
- Determining the functional characteristics of existing or directed components in the system and incorporating them in the analysis and allocation,
- Examining all life cycle functions, including the eight primary functions, as appropriate for the specific project,
- Performing trade studies to determine alternative functional approaches to meet requirements, and

- Revisiting the requirements analysis step as necessary to resolve functional issues.

The objective is to identify the functional, performance, and interface design requirements; it is not to design a solution…yet!

Functional Partitioning

Functional partitioning is the process of grouping functions that logically fit with the components likely to be used, and to minimize functional interfaces. Partitioning is performed as part of functional decomposition. It identifies logical groupings of functions that facilitate the use of modular components and open-system designs. Functional partitioning is also useful in understanding how existing equipment or components (including commercial) will function with or within the system.

Requirements Loop

During the performance of the Functional Analysis and Allocation process, it is expected that revisiting the requirements analysis process will be necessary. This is caused by the emergence of functional issues that will require re-examination of the higher-level requirements. Such issues might include directed components or standards that cause functional conflict, identification of a revised approach to functional sequencing, or, most likely, a conflict caused by mutually incompatible requirements.

Figure 5-1 gives an overview of the basic parameters of Functional Analysis and Allocation. The output of the process is the functional architecture. In its most basic form, the functional architecture is a simple hierarchical decomposition of the functions with associated performance requirements. As the architecture definition is refined and made more specific with the performance of the

- **Outputs:**
  - Functional architecture and supporting detail
- **Inputs:**
  - Outputs of the Requirements Analysis
- **Enablers:**
  - Multi-discipline product teams, decision database; Tools & Models, such as QFD, Functional Flow Block Diagrams, IDEF, N2 charts, Requirement Allocation Sheet, Timelines, Data Flow Diagrams, State/Mode Diagrams, Behavior Diagrams
- **Controls:**
  - Constraints; GFE, COTS, & Reusable S/W; System concept & subsystem choices; organizational procedures
- **Activities:**
  - Define system states and modes
  - Define system functions & external interfaces
  - Define functional interfaces
  - Allocate performance requirements to functions
  - Analyze performance
  - Analyze timing and resources
  - Analyze failure mode effects and criticality
  - Define fault detection and recovery behavior
  - Integrate functions

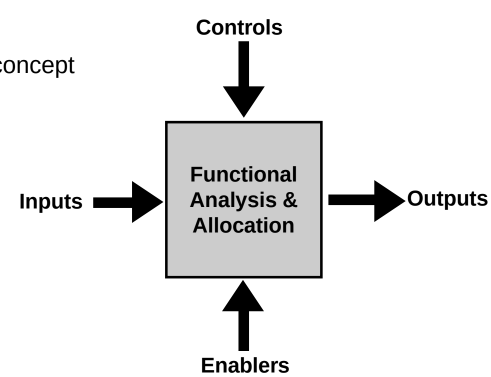

Figure 5-1. Functional Analysis and Allocation

activities listed in Figure 5-1, the functional architecture becomes more detailed and comprehensive. These activities provide a functional architecture with sufficient detail to support the Design Synthesis. They are performed with the aid of traditional tools that structure the effort and provide documentation for traceability. There are many tools available. The following are traditional tools that represent and explain the primary tasks of Functional Analysis and Allocation (several of these are defined and illustrated beginning on page 49):

- Functional flow block diagrams that define task sequences and relationships,
- IDEF0 diagrams that define process and data flows,
- Timeline analyses that define the time sequence of time critical functions, and
- Requirements allocation sheets that identify allocated performance and establish traceability of performance requirements.

5.3 FUNCTIONAL ARCHITECTURE

The functional architecture is a top-down decomposition of system functional and performance requirements. The architecture will show not only the functions that have to be performed, but also the logical sequencing of the functions and performance requirements associated with the functions. It also includes the functional description of existing and government-furnished items to be used in the system. This may require reverse engineering of these existing components.

The functional architecture produced by the Functional Analysis and Allocation process is the detailed package of documentation developed to analyze the functions and allocate performance requirements. It includes the functional flow block diagrams, timeline sheets, requirements allocation sheets, IDEF0 diagrams, and all other documentation developed to describe the functional characteristics of the system. However, there is a basic logic to the functional architecture, which in its preliminary form is presented in the example of Figure 5-2. The Functional Analysis and Allocation process would normally begin with the

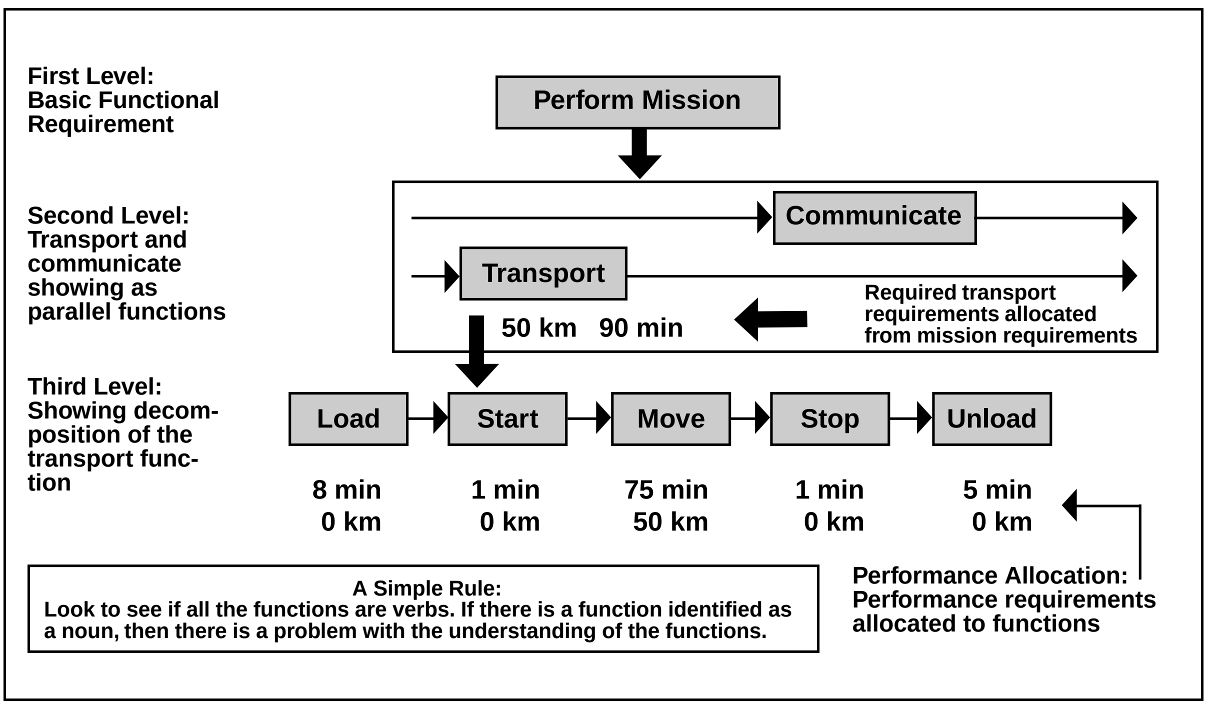

Figure 5-2. Functional Architecture Example

IPT drafting such a basic version of the architecture. This would generally give the IPT an understanding of the scope and direction of the effort.

Functional Architecture Example

The Marine Corps has a requirement to transport troops in squad-level units over a distance of 50 kilometers. Troops must be transported within 90 minutes from the time of arrival of the transport system. Constant communication is required during the transportation of troops. Figure 5-2 illustrates a preliminary functional architecture for this simple requirement.

5.4 SUMMARY POINTS

Functional analysis begins with the output of requirements analysis (that is, the identification of higher-level functional and performance requirements). Functional Analysis and Allocation consists of decomposition of higher-level functions to lower-levels and then allocation of requirements to those functions.

- There are many tools available to support the development of a Functional Architecture, such as: functional-flow block diagrams, timeline analysis sheet, requirements allocation sheet, Integrated Definition, and others.
- Use of the tools illustrated in this chapter is not mandatory, but the process they represent is:
  - Define task sequences and relationships (functional flow block diagram (FFBD)),
  - Define process and data flows (IDEF0 diagrams),
  - Define the time sequence of time-critical functions (timeline analysis sheets (TLS)), and
  - Allocate performance and establish traceability of performance requirements (requirements allocation sheets (RAS)).

SUPPLEMENT 5-A

FUNCTIONAL FLOW BLOCK DIAGRAM

The purpose of the functional flow block diagram (FFBD) is to describe system requirements in functional terms.

Objectives

The FFBD is structured to ensure that:

- All life cycle functions are covered.
- All elements of system are identified and defined (e.g. prime equipment, training, spare parts, data, software, etc.).
- System support requirements are identified to specific system functions.
- Proper sequencing of activities and design relationships are established including critical design interfaces.

Characteristics

The FFBD is functionally oriented—not solution oriented. The process of defining lower-level functions and sequencing relationships is often referred to as functional decomposition. It allows traceability vertically through the levels. It is a key step in developing the functional architecture from which designs may be synthesized.

Figure 5-3 shows the flow-down structure of a set of FFBDs and Figure 5-4 shows the format of an FFBD.

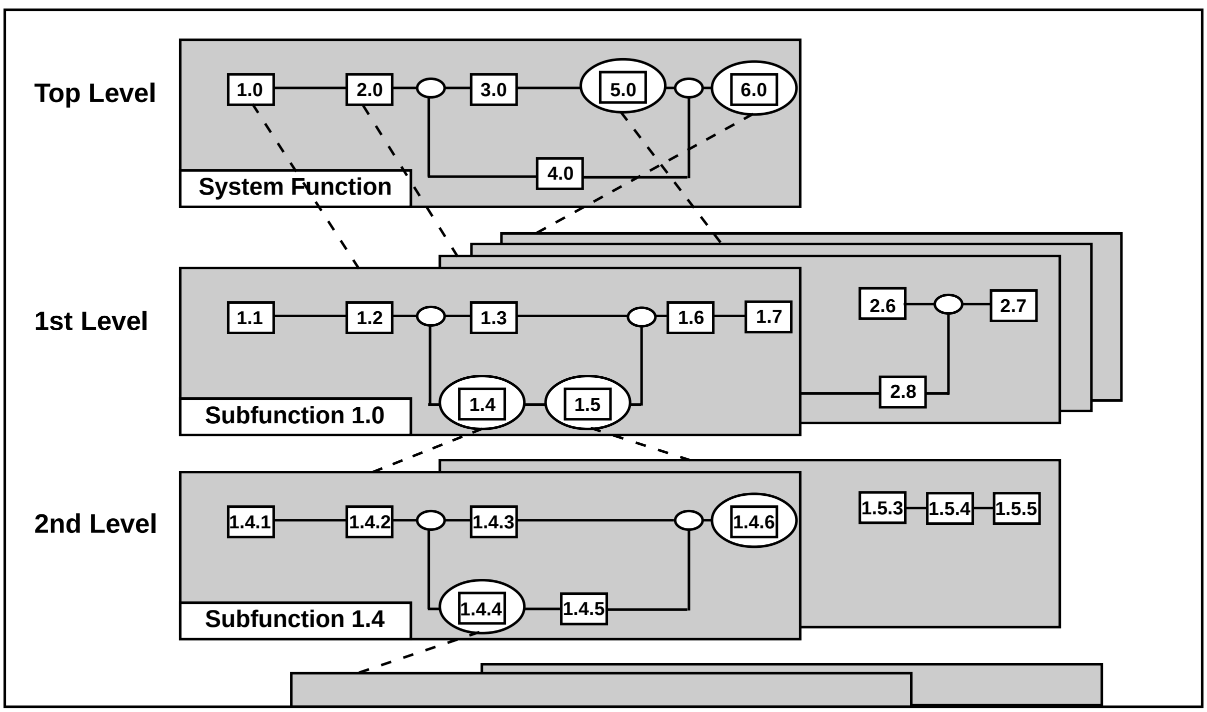

Figure 5-3. FFBD Traceability and Indenture

Key FFBD Attributes

Function block: Each function on an FFBD should be separate and be represented by single box (solid line). Each function needs to stand for definite, finite, discrete action to be accomplished by system elements.

Function numbering: Each level should have a consistent number scheme and provide information concerning function origin. (E.g., top level—1.0, 2.0, 3.0, etc; first indenture (level 2)—1.1, 1.2, 1.3, etc; second indenture (level 3)—1.1.1, 1.1.2, 1.1.3, etc.) These numbers establish identification and relationships that will carry through all Functional Analysis and Allocation activities and facilitate traceability from lower to top levels.

Functional reference: Each diagram should contain a reference to other functional diagrams by using a functional reference (box in brackets).

Flow connection: Lines connecting functions should only indicate function flow and not a lapse in time or intermediate activity.

Flow direction: Diagrams should be laid out so that the flow direction is generally from left to right. Arrows are often used to indicate functional flows.

Summing gates: A circle is used to denote a summing gate and is used when AND/OR is present. AND is used to indicate parallel functions and all conditions must be satisfied to proceed. OR is used to indicate that alternative paths can be satisfied to proceed.

GO and NO-GO paths: "G" and "bar G" are used to denote "go" and "no-go" conditions. These symbols are placed adjacent to lines leaving a particular function to indicate alternative paths.

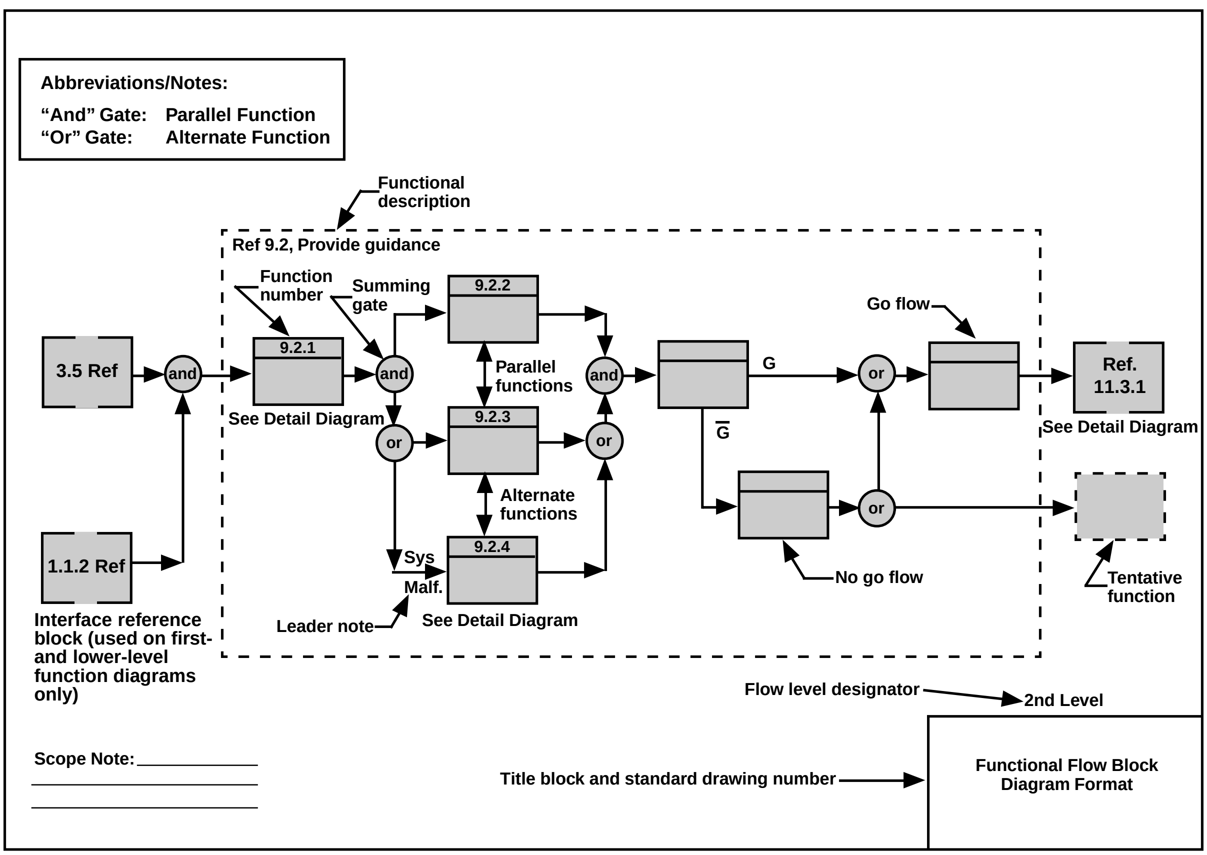

Figure 5-4. Functional Flow Block Diagrams (FFBD) Format

SUPPLEMENT 5-B

IDEF0

Integration Definition for Function Modeling (IDEF0) is a common modeling technique for the analysis, development, re-engineering, and integration of information systems; business processes; or software engineering analysis. Where the FFBD is used to show the functional flow of a product, IDEF0 is used to show data flow, system control, and the functional flow of life cycle processes.

IDEF0 is capable of graphically representing a wide variety of business, manufacturing and other types of enterprise operations to any level of detail. It provides rigorous and precise description, and promotes consistency of usage and interpretation. It is well-tested and proven through many years of use by government and private industry. It can be generated by a variety of computer graphics tools. Numerous commercial products specifically support development and analysis of IDEF0 diagrams and models.

IDEF0 is a model that consists of a hierarchical series of diagrams, text, and glossary cross-referenced to each other. The two primary modeling components are: functions (represented on a diagram by boxes), and data and objects that interrelate those functions (represented by arrows). As shown by Figure 5-5 the position at which the arrow attaches to a box conveys the specific role of the interface. The controls enter the top of the box. The inputs, the data or objects acted upon by the operation, enter the box from the left. The outputs of the operation leave the right-hand side of the box. Mechanism arrows that provide supporting means for performing the function join (point up to) the bottom of the box.

The IDEF0 process starts with the identification of the prime function to be decomposed. This function is identified on a "Top Level Context Diagram," that defines the scope of the particular IDEF0 analysis. An example of a Top Level Context Diagram for an information system management process is shown in Figure 5-6. From this diagram lower-level diagrams are generated. An example of a derived diagram, called a "child" in

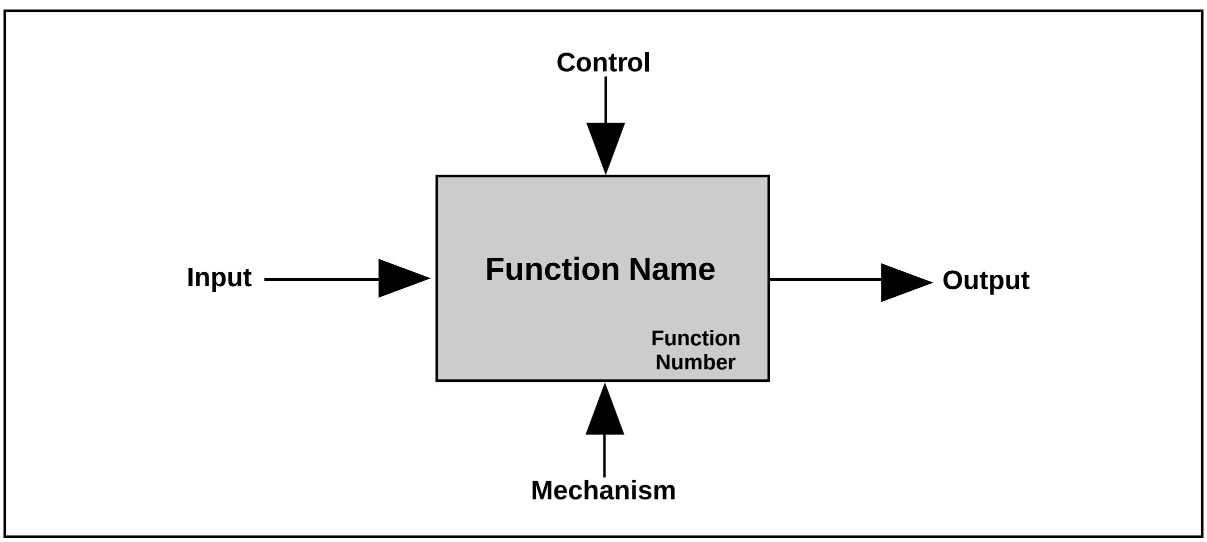

Figure 5-5. Integration Definition for Function Modeling (IDEF0) Box Format

IDEF0 terminology, for a life cycle function is shown in Figure 5-7.

An associated technique, Integration Definition for Information Modeling (IDEF1x), is used to supplement IDEF0 for data intensive systems. The IDEF0 standard, Federal Information Processing Standards Publication 183 (FIPS 183), and the IDEF1x standard (FIPS 184) are maintained by the National Institute of Standards and Technology (NIST).

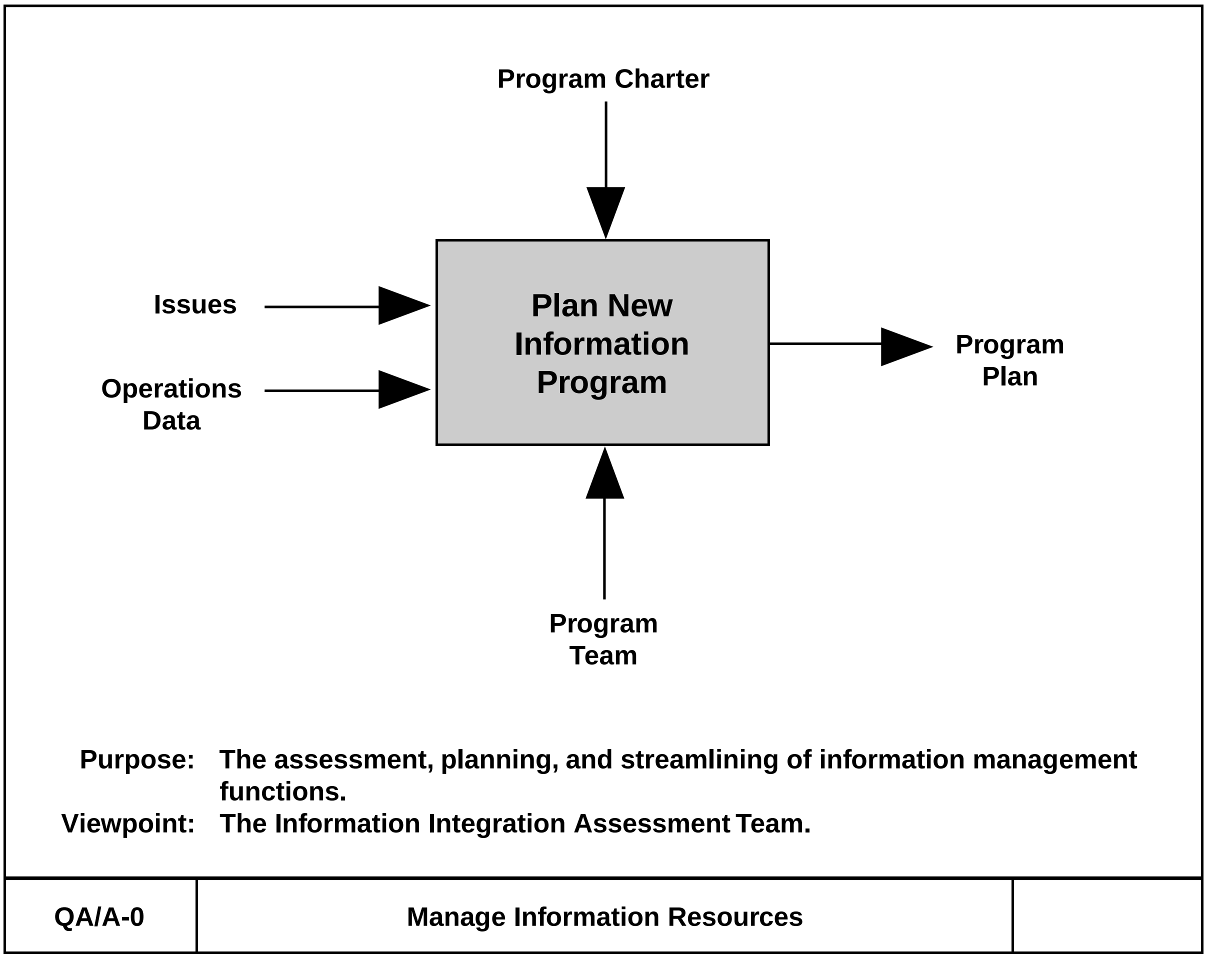

Figure 5-6. Top-Level Context Diagram

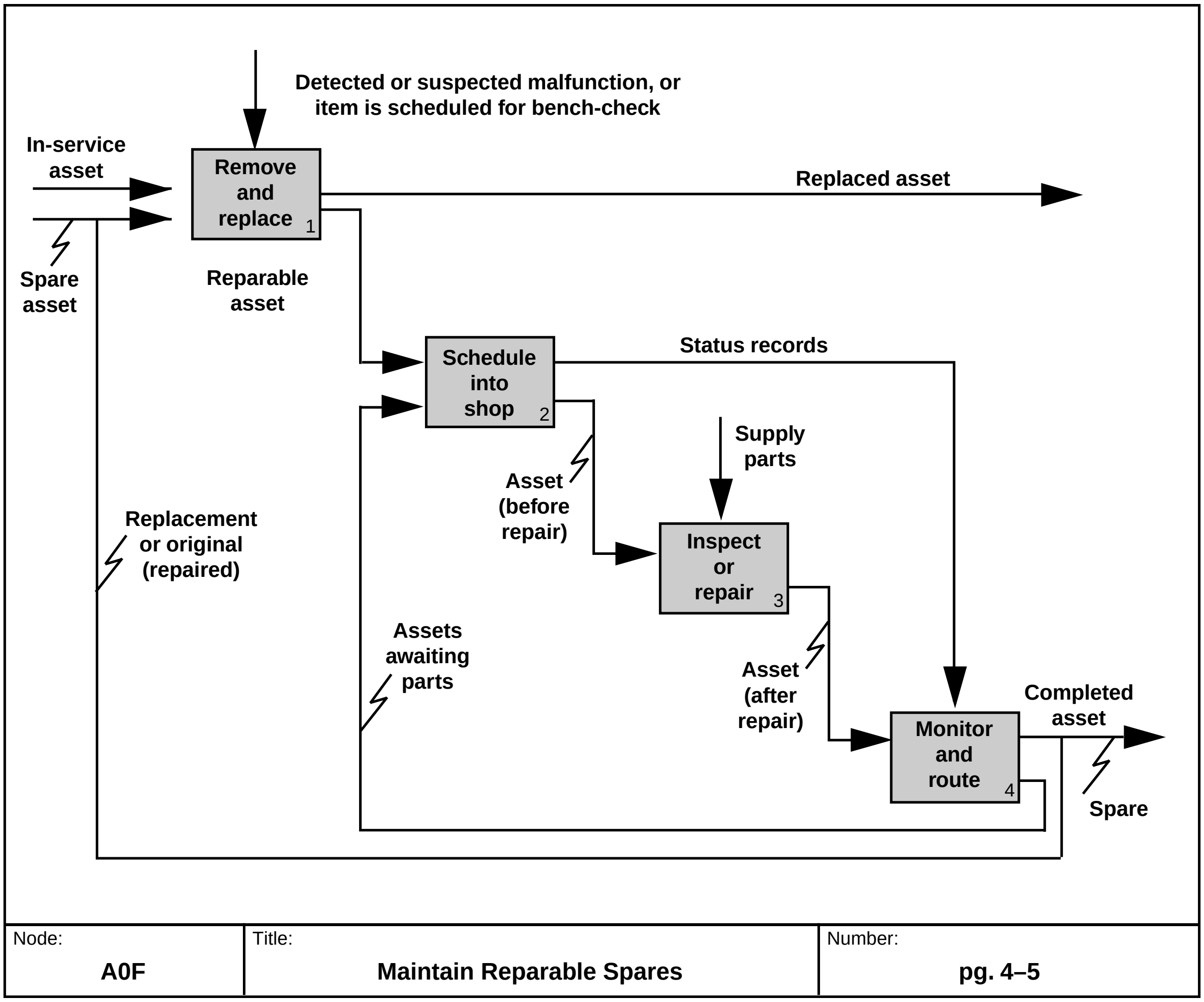

Figure 5-7. IDEF0 Diagram Example

SUPPLEMENT 5-C

TIMELINE ANALYSIS SHEETS

The timeline analysis sheet (TLS) adds detail to defining durations of various functions. It defines concurrency, overlapping, and sequential relationships of functions and tasks. It identifies time critical functions that directly affect system availability, operating time, and maintenance downtime. It is used to identify specific time-related design requirements.

The TLS includes purpose of function and the detailed performance characteristics, criticality of function, and design constraints. It identifies both quantitative and qualitative performance requirements. Initial resource requirements are identified.

Figure 5-8 shows an example of a TLS. The time required to perform function 3.1 and its subfunctions are presented on a bar chart showing how the timelines relate. (Function numbers match the FFBD.)

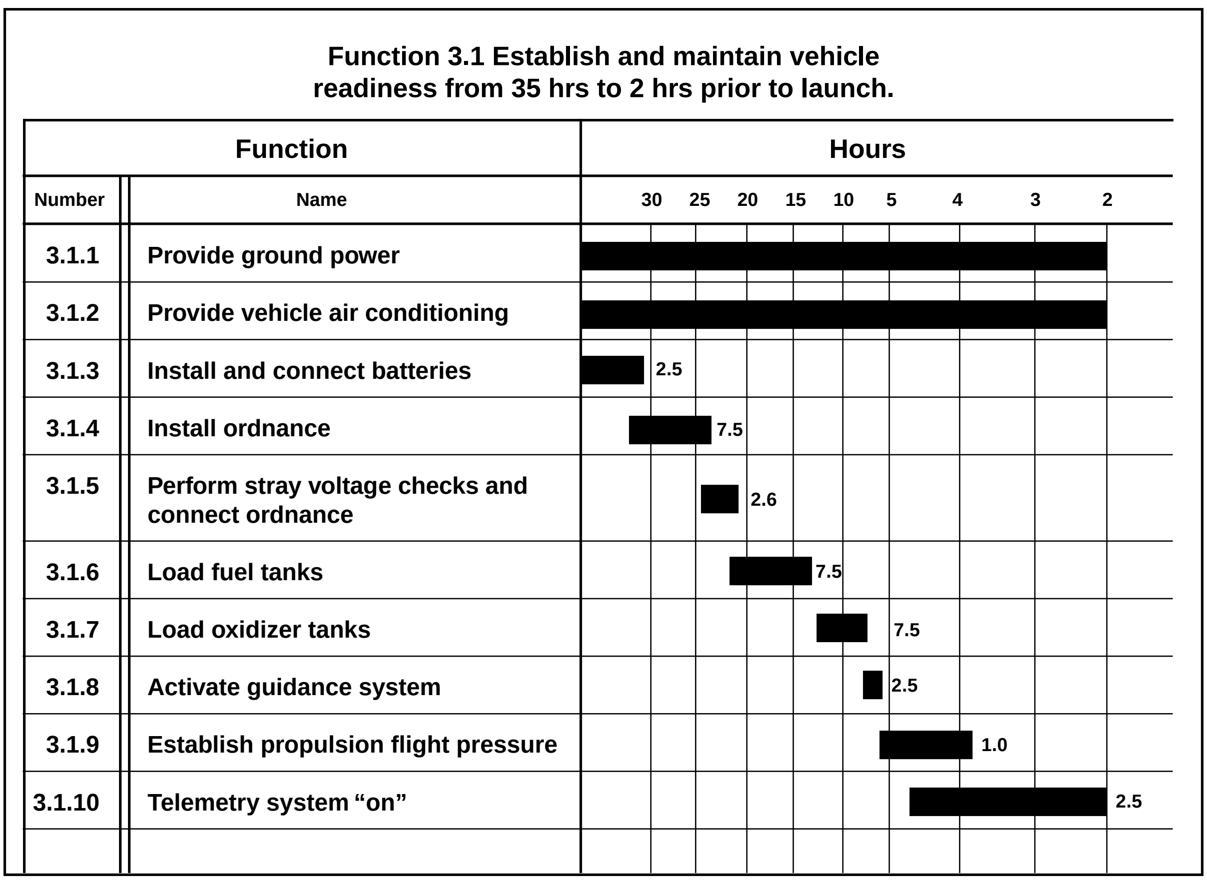

Figure 5-8. Time Analysis Sheet

SUPPLEMENT 5-D

REQUIREMENTS ALLOCATION SHEET

The Requirements Allocation Sheet documents the connection between allocated functions, allocated performance and the physical system. It provides traceability between Functional Analysis and Allocation and Design Synthesis, and shows any disconnects. It is a major tool in maintaining consistency between functional architectures and designs that are based on them. (Function numbers match the FFBD.)

Requirements Allocation Sheet	Functional Flow Diagram Title and No. 2.58.4 Provide Guidance Compartment Cooling	Equipment Identification		
Function Name and No.	**Functional Performance and Design Requirements**	**Facility Rqmnts**	**Nomen-clature**	**CI or Detail Spec No.**
2.58.4 Provide Guidance Compartment Cooling	The temperature in the guidance compartment must be maintained at the initial calibration temperature of +0.2 Deg F. The initial calibration temperature of the compartment will be between 66.5 and 68.5 Deg F.			
2.58.4.1 Provide Chilled Coolant (Primary)	A storage capacity for 65 gal of chilled liquid coolant (deionized water) is required. The temperature of the stored coolant must be monitored continuously. The stored coolant must be maintained within a temperature range of 40–50 Deg F. for an indefinite period of time. The coolant supplied must be free of obstructive particles 0.5 micron at all times.			

Figure 5-9. Requirements Allocation Sheet (Example)

CHAPTER 6

DESIGN SYNTHESIS

6.1 DESIGN DEVELOPMENT

Design Synthesis is the process by which concepts or designs are developed based on the functional descriptions that are the products of Functional Analysis and Allocation. Design synthesis is a creative activity that develops a physical architecture (a set of product, system, and/or software elements) capable of performing the required functions within the limits of the performance parameters prescribed. Since there may be several hardware and/or software architectures developed to satisfy a given set of functional and performance requirements, synthesis sets the stage for trade studies to select the best among the candidate architectures. The objective of design synthesis is to combine and restructure hardware and software components in such a way as to achieve a design solution capable of satisfying the stated requirements. During concept development, synthesis produces system concepts and establishes basic relationships among the subsystems. During preliminary and detailed design, subsystem and component descriptions are elaborated, and detailed interfaces between all system components are defined.

The physical architecture forms the basis for design definition documentation, such as, specifications, baselines, and work breakdown structures (WBS). Figure 6-1 gives an overview of the basic parameters of the synthesis process.

- **Outputs:**
  - Physical Architecture (Product Elements and Software Code)
  - Decision Database
- **Inputs:**
  - Functional Architecture
- **Enablers:**
  - IPTs, Decision Database, Automated Tools, Models
- **Controls:**
  - Constraints; GFE, COTS, & Reusable S/W; System concept & subsystem choices; organizational procedures
- **Activities:**
  - Allocate functions and constraints to system elements
  - Synthesize system element alternatives
  - Assess technology alternatives
  - Define physical interfaces
  - Define system product WBS
  - Develop life cycle techniques and procedures
  - Integrate system elements
  - Select preferred concept/design

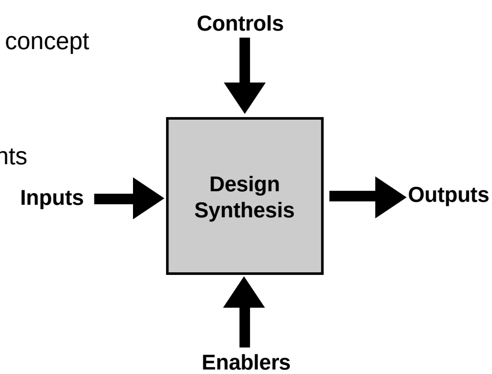

Figure 6-1. Design Synthesis

Characteristics

Physical architecture is a traditional term. Despite the name, it includes software elements as well as hardware elements. Among the characteristics of the physical architecture (the primary output of Design Synthesis) are the following:

- The correlation with functional analysis requires that each physical or software component meets at least one (or part of one) functional requirement, though any component can meet more than one requirement,
- The architecture is justified by trade studies and effectiveness analyses,
- A product WBS is developed from the physical architecture,
- Metrics are developed to track progress among KPPs, and
- All supporting information is documented in a database.

Modular Designs

Modular designs are formed by grouping components that perform a single independent function or single logical task; have single entry and exit points; and are separately testable. Grouping related functions facilitates the search for modular design solutions and furthermore increases the possibility that open-systems approaches can be used in the product architecture.

Desirable attributes of the modular units include low coupling, high cohesion, and low connectivity. Coupling between modules is a measure of their interdependence, or the amount of information shared between two modules. Decoupling modules eases development risks and makes later modifications easier to implement. Cohesion (also called binding) is the similarity of tasks performed within the module. High cohesion is desirable because it allows for use of identical or like (family or series) components, or for use of a single component to perform multiple functions. Connectivity refers to the relationship of internal elements within one module to internal elements within another module. High connectivity is undesirable in that it creates complex interfaces that may impede design, development, and testing.

Design Loop

The design loop involves revisiting the functional architecture to verify that the physical architecture developed is consistent with the functional and performance requirements. It is a mapping between the functional and physical architectures. Figure 6-2 shows an example of a simple physical architecture and how it relates to the functional architecture. During design synthesis, re-evaluation of the functional analysis may be caused by the discovery of design issues that require re-examination of the initial decomposition, performance allocation, or even the higher-level requirements. These issues might include identification of a promising physical solution or open-system opportunities that have different functional characteristics than those foreseen by the initial functional architecture requirements.

6.2 SYNTHESIS TOOLS

During synthesis, various analytical, engineering, and modeling tools are used to support and document the design effort. Analytical devices such as trade studies support decisions to optimize physical solutions. Requirements Allocation Sheets (RAS) provide traceability to the functional and performance requirements. Simple descriptions like the Concept Decription Sheet (CDS) help visualize and communicate the system concept. Logic models, such as the Schematic Block Diagram (SBD), establish the design and the interrelationships within the system.

Automated engineering management tools such as Computer-Aided Design (CAD), Computer-Aided-Systems Engineering (CASE), and the Computer-Aided-Engineering (CAE) can help organize, coordinate and document the design effort. CAD generates detailed documentation describing the product design including SBDs, detailed

PHYSICAL ARCHITECTURE

Aircraft

FUNCTIONAL ARCHITECTURE

Function Performed	Air Frame	Engine	Communi-cations	Nav System	Fire Control
Preflight check	X	X	X	X	X
Fly					
Load	X				
Taxi	X	X	X		
Take-off	X	X			
Cruise	X	X	X	X	
Recon	X	X	X	X	
Communicate			X		
–					
–					
Surveillance					
–					
–					

Figure 6–2. Functional/Physical Matrix

drawings, three dimensional and solid drawings, and it tracks some technical performance measurements. CAD can provide significant input for virtual modeling and simulations. It also provides a common design database for integrated design developments. Computer-Aided Engineering can provide system requirements and performance analysis in support of trade studies, analysis related to the eight primary functions, and cost analyses. Computer-Aided Systems Engineering can provide automation of technical management analyses and documentation.

Modeling

Modeling techniques allow the physical product to be visualized and evaluated prior to design decisions. Models allow optimization of hardware and software parameters, permit performance predictions to be made, allow operational sequences to be derived, and permit optimum allocation of functional and performance requirements among the system elements. The traditional logical prototyping used in Design Synthesis is the Schematic Block Diagram.

6.3 SUMMARY POINTS

- Synthesis begins with the output of Functional Analysis and Allocation (the functional architecture). The functional architecture is transformed into a physical architecture by defining physical components needed to perform the functions identified in Functional Analysis and Allocation.

- Many tools are available to support the development of a physical architecture:
  - Define and depict the system concept (CDS),
  - Define and depict components and their relationships (SBD), and
  - Establish traceability of performance requirements to components (RAS).
- Specifications and the product WBS are derived from the physical architecture.

SUPPLEMENT 6-A

CONCEPT DESCRIPTION SHEET

The Concept Description Sheet describes (in textual or graphical form) the technical approach or the design concept, and shows how the system will be integrated to meet the performance and functional requirements. It is generally used in early concept design to show system concepts.

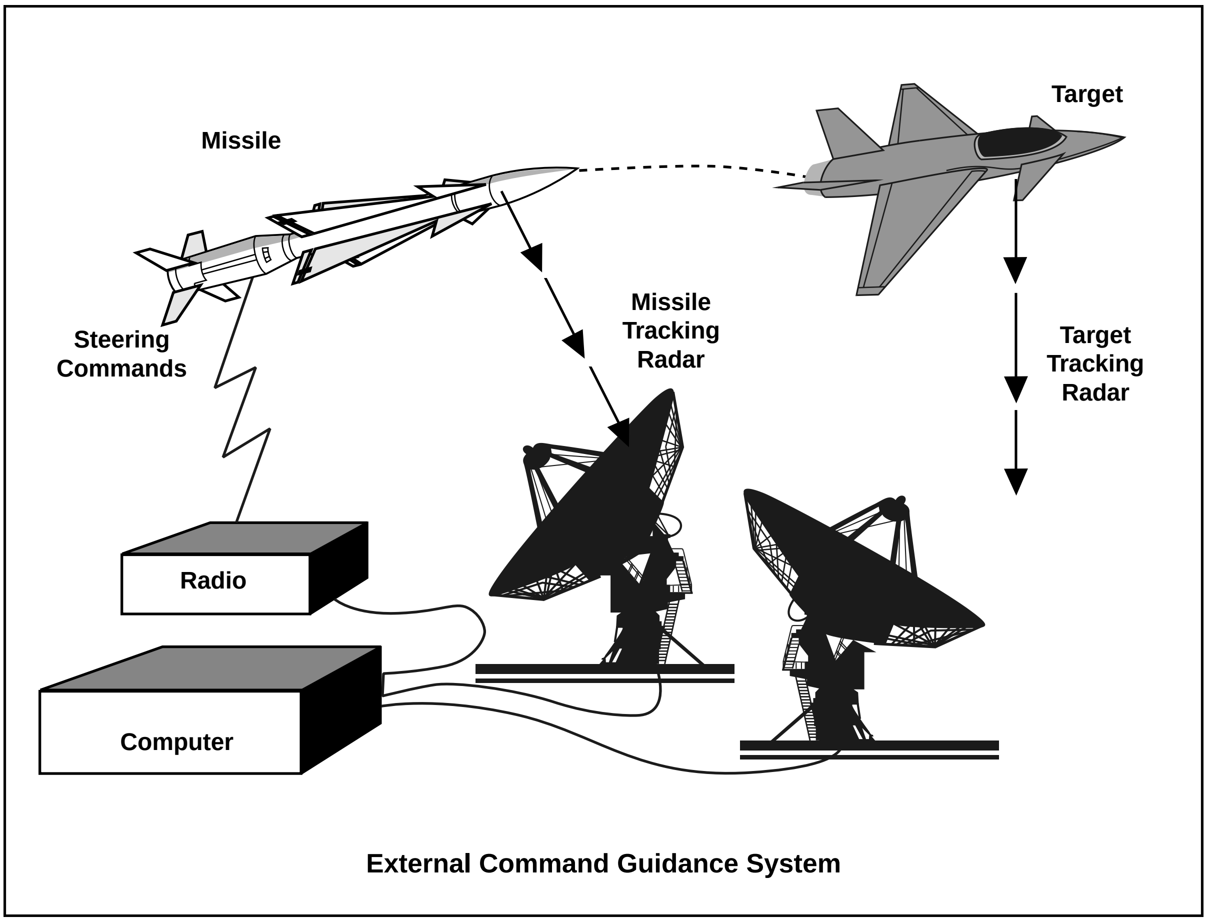

Figure 6-3. Concept Description Sheet Example

SUPPLEMENT 6-B

SCHEMATIC BLOCK DIAGRAMS

The Schematic Block Diagram (SBD) depicts hardware and software components and their interrelationships. They are developed at successively lower levels as analysis proceeds to define lower-level functions within higher-level requirements. These requirements are further subdivided and allocated using the Requirements Allocation Sheet (RAS). SBDs provide visibility of related system elements, and traceability to the RAS, FFBD, and other system engineering documentation. They describe a solution to the functional and performance requirements established by the functional architecture; show interfaces between the system components and between the system components and other systems or subsystems; support traceability between components and their functional origin; and provide a valuable tool to enhance configuration control. The SBD is also used to develop Interface Control Documents (ICDs) and provides an overall understanding of system operations.

A simplified SBD, Figure 6-4, shows how components and the connection between them are presented on the diagram. An expanded version is usually developed which displays the detailed functions performed within each component and a detailed depiction of their interrelationships. Expanded SBDs will also identify the WBS numbers associated with the components.

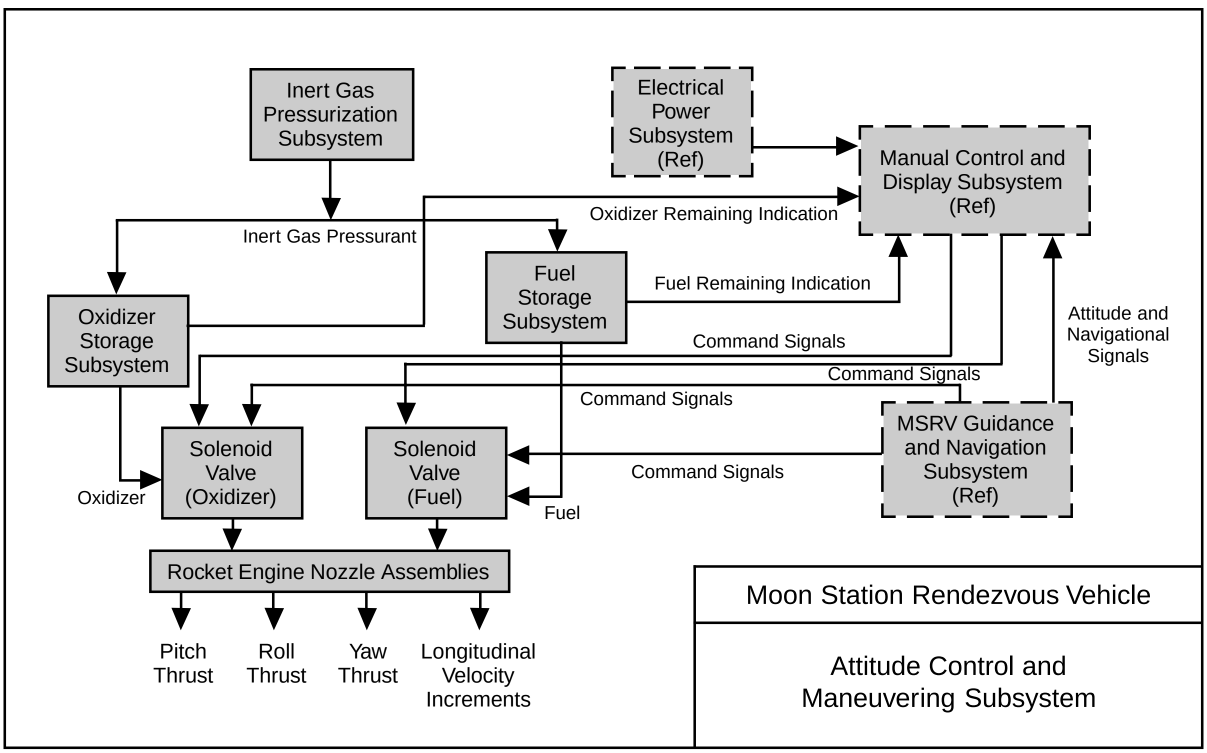

Figure 6-4. Schematic Block Diagram Example

SUPPLEMENT 6-C

REQUIREMENTS ALLOCATION SHEET

The RAS initiated in Functional Analysis and Allocation is expanded in Design Synthesis to document the connection between functional requirements and the physical system. It provides traceability between the Functional Analysis and Allocation and Synthesis activities. It is a major tool in maintaining consistency between functional architectures and the designs that are based on them. (Configuration Item (CI) numbers match the WBS.)

Requirements Allocation Sheet	Functional Flow Diagram Title and No. 2.58.4 Provide Guidance Compartment Cooling		Equipment Identification	
Function Name and No.	**Functional Performance and Design Requirements**	**Facility Rqmnts**	**Nomenclature**	**CI or Detail Spec No.**
2.58.4 Provide Guidance Compartment Cooling	The temperature in the guidance compartment must be maintained at the initial calibration temperature of +0.2 Deg F. The initial calibration temperature of the compartment will be between 66.5 and 68.5 Deg F.		Guidance Compartment Cooling System	3.54.5
2.58.4.1 Provide Chilled Coolant (Primary)	A storage capacity for 65 gal of chilled liquid coolant (deionized water) is required. The temperature of the stored coolant must be monitored continuously. The stored coolant must be maintained within a temperature range of 40–50 Deg F. for an indefinite period of time. The coolant supplied must be free of obstructive particles 0.5 micron at all times.		Guidance Compartment Coolant Storage Subsystem	3.54.5.1

Figure 6-5. Requirements Allocation Sheet (Example)

CHAPTER 7

VERIFICATION

7.1 GENERAL

The Verification process confirms that Design Synthesis has resulted in a physical architecture that satisfies the system requirements. Verification represents the intersection of systems engineering and test and evaluation.

Verification Objectives

The objectives of the Verification process include using established criteria to conduct verification of the physical architecture (including software and interfaces) from the lowest level up to the total system to ensure that cost, schedule, and performance requirements are satisfied with acceptable levels of risk. Further objectives include generating data (to confirm that system, subsystem, and lower level items meet their specification requirements) and validating technologies that will be used in system design solutions. A method to verify each requirement must be established and recorded during requirements analysis and functional allocation activities. (If it can not be verified it can not be a legitimate requirement.) The verification list should have a direct relationship to the requirements allocation sheet and be continually updated to correspond to it.

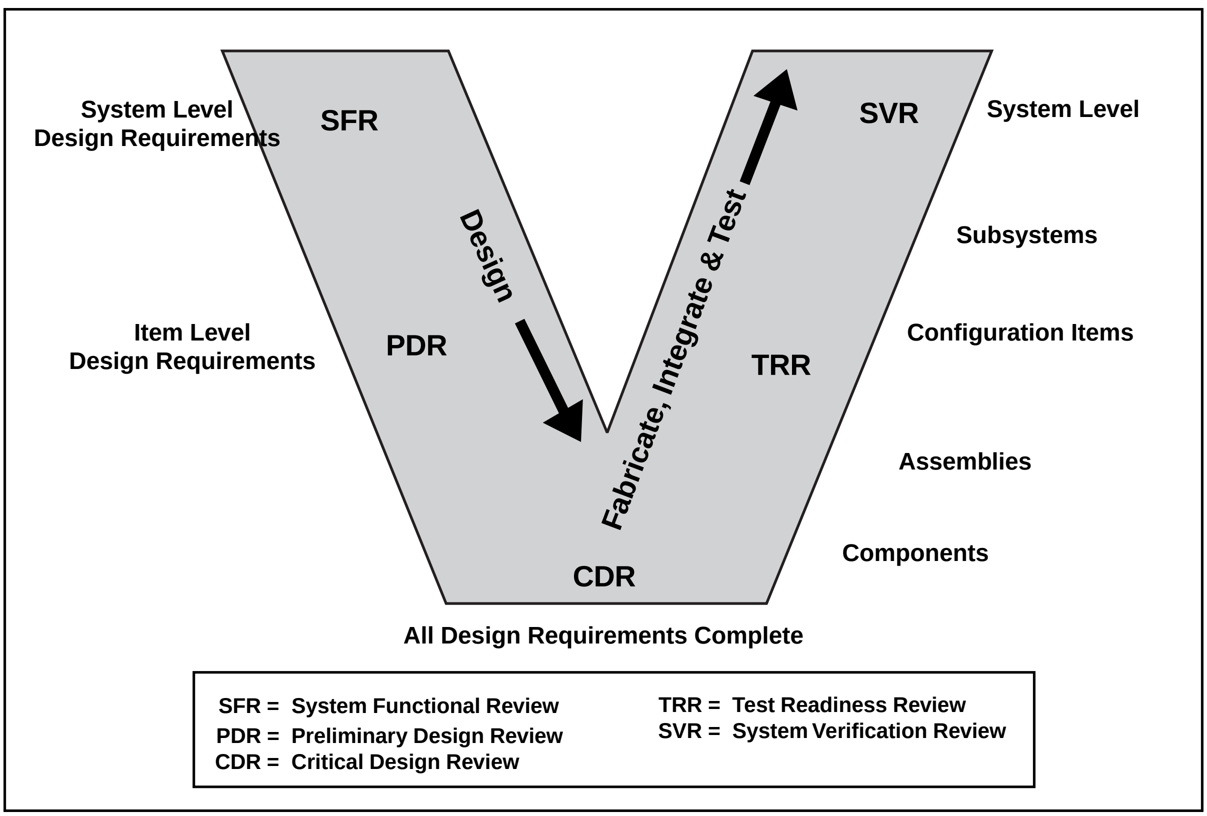

Figure 7-1. Systems Engineering and Verification

Verification Activities

System design solutions are verified by the following types of activities:

1. Analysis – the use of mathematical modeling and analytical techniques to predict the compliance of a design to its requirements based on calculated data or data derived from lower level component or subsystem testing. It is generally used when a physical prototype or product is not available or not cost effective. Analysis includes the use of both modeling and simulation which is covered in some detail in chapter 13,

2. Inspection – the visual examination of the system, component, or subsystem. It is generally used to verify physical design features or specific manufacturer identification,

3. Demonstration – the use of system, subsystem, or component operation to show that a requirement can be achieved by the system. It is generally used for a basic confirmation of performance capability and is differentiated from testing by the lack of detailed data gathering, or

4. Test – the use of system, subsystem, or component operation to obtain detailed data to verify performance or to provide sufficient information to verify performance through further analysis. Testing is the detailed quantifying method of verification, and as described later in this chapter, it is ultimately required in order to verify the system design.

Choice of verification methods must be considered an area of potential risk. Use of inappropriate methods can lead to inaccurate verification. Required defining characteristics, such as key performance parameters (KPPs) are verified by demonstration and/or test. Where total verification by test is not feasible, testing is used to verify key characteristics and assumptions used in design analysis or simulation. Validated models and simulation tools are included as analytical verification methods that complement other methods. The focus and nature of verification activities change as designs progress from concept to detailed designs to physical products.

During earlier design stages, verification focuses on proof of concept for system, subsystem and component levels. During later stages, as the product definition effort proceeds, the focus turns to verifying that the system meets the customer requirements. As shown by Figure 7-1, design is a top-down process while the Verification activity is a bottom-up process. Components will be fabricated and tested prior to the subsystems. Subsystems will be fabricated and tested prior to the completed system.

Performance Verification

Performance requirements must be objectively verifiable, i.e., the requirement must be measurable. Where appropriate, Technical Performance Measurements (TPM) and other management metrics are used to provide insight on progress toward meeting performance goals and requirements. IEEE Standard P1220 provides a structure for Verification activity. As shown in Figure 7-2 the structure is comprehensive and provides a good starting point for Verification planning.

7.2 DOD TEST AND EVALUATION

DoD Test and Evaluation (T&E) policies and procedures directly support the system engineering process of Verification. Testing is the means by which objective judgments are made regarding the extent to which the system meets, exceeds, or fails to meet stated objectives. The purpose of evaluation is to review, analyze, and assess data obtained from testing and other means to aid in making systematic decisions. The purpose of DoD T&E is to verify technical performance, operational effectiveness, operational suitability; and it provides essential information in support of decision making.

Common Types of T&E in DoD

T&E policy requires developmental tests. They confirm that technical requirements have been

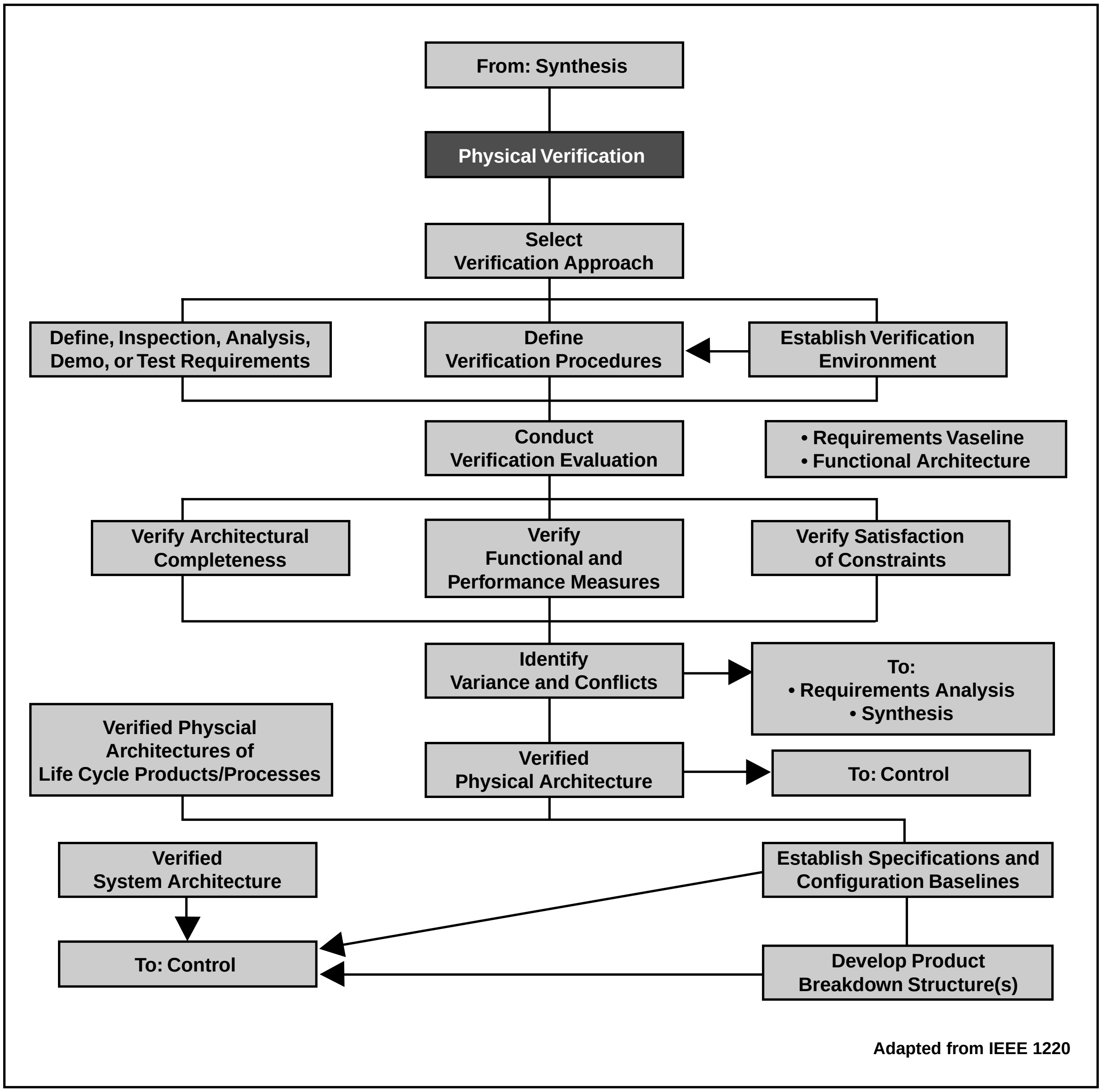

Figure 7-2. Verification Tasks

satisfied, and independent analysis and tests verify the system's operational effectiveness and suitability. DoD T&E traditionally and by directive is categorized as:

- Developmental T&E which focuses primarily on technical achievement,
- Operational T&E which focuses on operational effectiveness and suitability and includes Early Operational Assessments (EOA), Operational Assessment (OA), Initial Operational Test and Evaluation (IOT&E), and Follow-On Operational Test and Evaluation (FOT&E), and
- Live Fire T&E which provides assessment of the vulnerability and lethality of a system by subjecting it to real conditions comparable to the required mission.

T&E

The program office plans and manages the test effort to ensure testing is timely, efficient, comprehensive and complete—and that test results are converted into system improvements. Test planning will determine the effectiveness of the verification process. Like all systems engineering planning activities, careful attention to test planning can reduce program risk. The key test planning document is the Test and Evaluation Master Plan (TEMP). This document lays out the objectives, schedule, and resources reflecting program office and operational test organization planning decisions. To ensure integration of this effort, the program office organizes a Test Planning Work Group (TPWG) or Test Working Level IPT (WIPT) to coordinate the test planning effort.

Test Planning Work Group/Test WIPT

The TPWG/Test WIPT is intended to facilitate the integration of test requirements and activities through close coordination between the members who represent the material developer, designer community, logistic community, user, operational tester, and other stakeholders in the system development. The team outlines test needs based on system requirements, directs test design, determines needed analyses for each test, identifies potential users of test results, and provides rapid dissemination of test and evaluation results.

Test and Evaluation Master Plan (TEMP)

The Test and Evaluation Master Plan is a mandatory document prepared by the program office. The operational test organization reviews it and provides the operational test planning for inclusion. The TEMP is then negotiated between the program office and operational test organization. After differences are resolved, it is approved at appropriate high levels in the stakeholder organizations. After approval it becomes binding on managers and designers (similar to the binding nature of the Operational Requirements Document (ORD)).

The TEMP is a valuable Verification tool that provides an excellent template for technology, system, and major subsystem-level Verification planning. The TEMP includes a reaffirmation of the user requirements, and to an extent, an interpretation of what those requirements mean in various operational scenarios. Part I of the required TEMP format is *System Introduction*, which provides the mission description, threat assessment, MOEs/MOSs, a system description, and an identification of critical technical parameters. Part II, *Integrated Test Program Summary*, provides an integrated test program schedule and a description of the overall test management process. Part III, *Developmental Test & Evaluation (DT&E) Outline*, lays out an overview of DT&E efforts and a description of future DT&E. Part IV, *Operational Test & Evaluation (OT&E) Outline*, is provided by the operational test organization and includes an OT&E overview, critical operational issues, future OT&E description, and LFT&E description. Part V, *Test & Evaluation Resource Summary*, identifies the necessary physical resources and activity responsibilities. This last part includes such items as test articles, test sites, test instrumentation, test support equipment, threat representation, test targets and other expendables, operational force test support, simulations, models, test-beds, special requirements, funding, and training.

Key Performance Parameters

Every system will have a set of KPPs that are the performance characteristics that *must* be achieved by the design solution. They flow from the operational requirements and the resulting derived MOEs. They can be identified by the user, the decision authority, or the operational tester. They are documented in the TEMP.

Developmental Test and Evaluation

The DT&E verifies that the design solution meets the system technical requirements and the system is prepared for successful OT&E. DT&E activities assess progress toward resolving critical operational issues, the validity of cost-performance tradeoff decisions, the mitigation of acquisition technical risk, and the achievement of system maturity.

DT&E efforts:

- Identify potential operational and technological capabilities and limitations of the alternative concepts and design options being pursued;

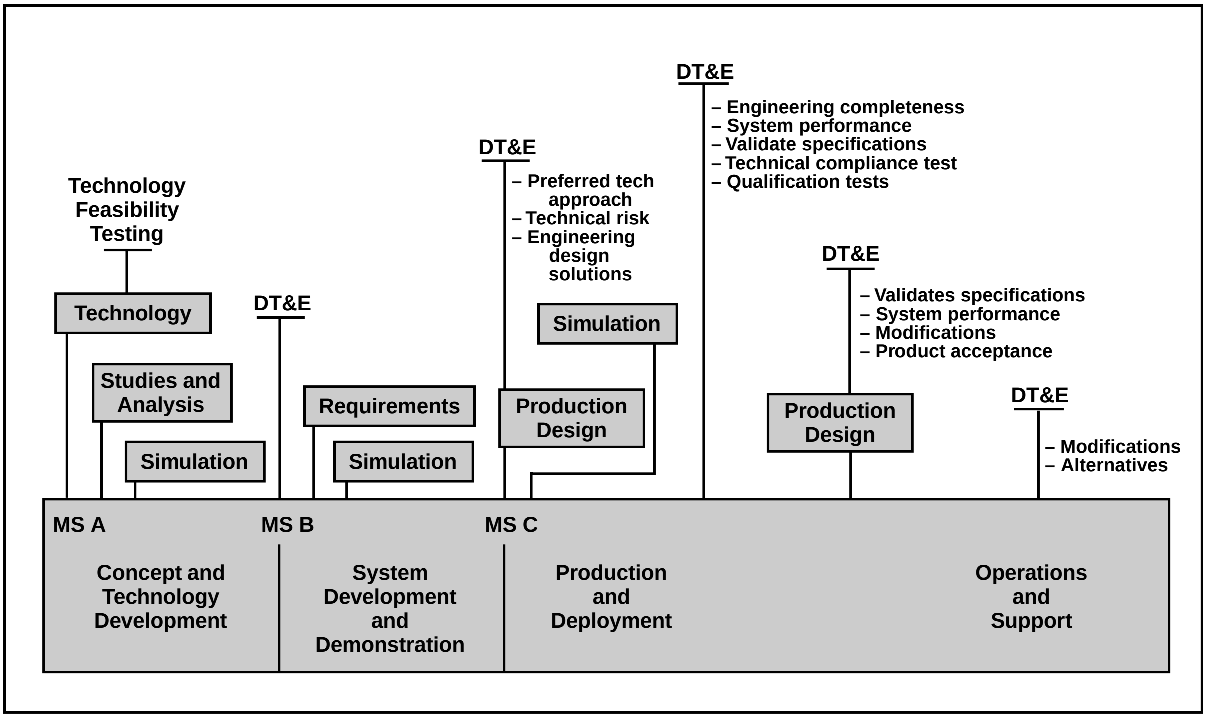

Figure 7-3. DT&E During System Acquisition

- Support the identification of cost-performance tradeoffs by providing analyses of the capabilities and limitations of alternatives;
- Support the identification and description of design technical risks;
- Assess progress toward resolving Critical Operational Issues, mitigating acquisition technical risk, achieving manufacturing process requirements and system maturity;
- Assess validity of assumptions and analysis conclusions; and
- Provide data and analysis to certify the system ready for OT&E, live-fire testing and other required certifications.

Figure 7-3 highlights some of the more significant DT&E focus areas and where they fit in the acquisition life cycle.

Live Fire Test and Evaluation

LFT&E is performed on any Acquisition Category (ACAT) I or II level weapon system that includes features designed to provide protection to the system or its users in combat. It is conducted on a production configured article to provide information concerning potential user casualties, vulnerabilities, and lethality. It provides data that can establish the system's susceptibility to attack and performance under realistic combat conditions.

Operational Test and Evaluation

OT&E programs are structured to determine the operational effectiveness and suitability of a system under realistic conditions, and to determine if the minimum acceptable operational performance requirements as specified in the ORD and reflected by the KPPs have been satisfied. OT&E uses threat-representative forces whenever possible, and employs typical users to operate and maintain the system or item under conditions simulating both combat stress and peacetime conditions. Operational tests will use production or production-

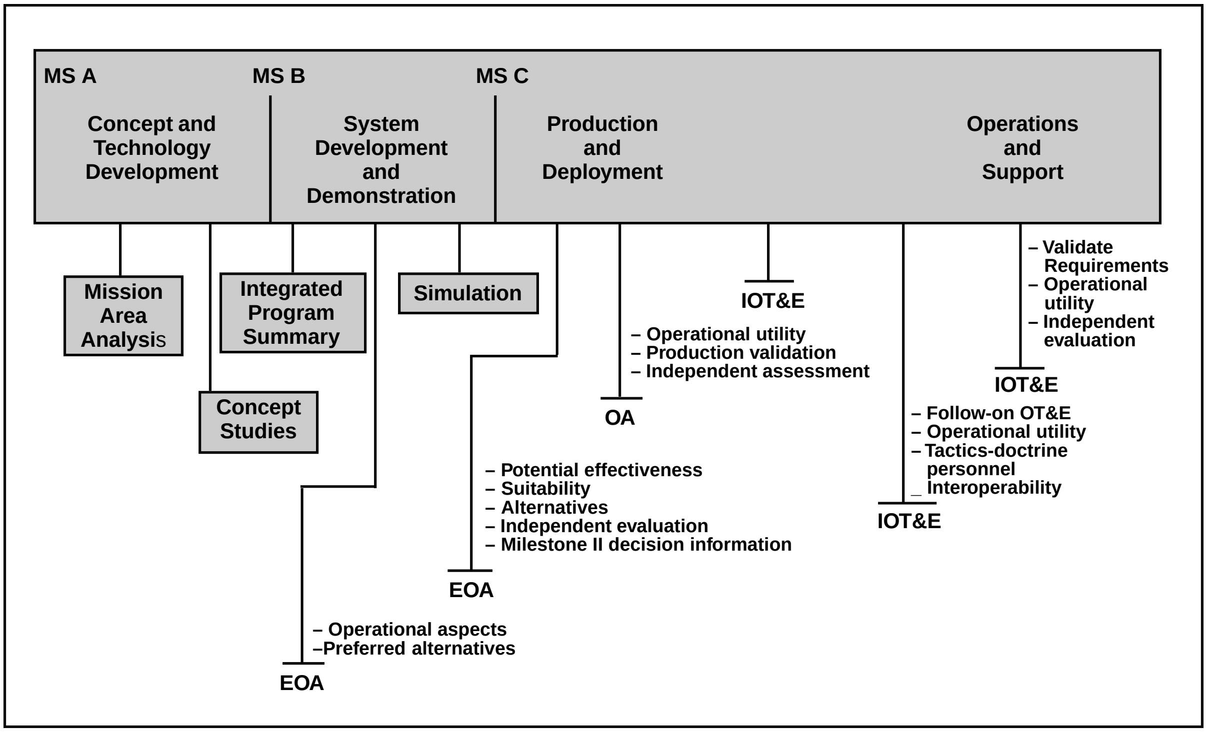

Figure 7-4. OT&E During System Acquisition

representative articles for the operational tests that support the full-rate production decision. Live Fire Tests are usually performed during the operational testing period. Figure 7-4 shows the major activities associated with operational testing and where they fit in the DoD acquisition life cycle.

OT&E Differences

Though the overall objective of both DT&E and OT&E is to verify the effectiveness and suitability of the system, there are distinct differences in their specific objects and focus. DT&E primarily focuses on verifying system technical requirements, while OT&E focuses on verifying operational requirements. DT&E is a program office responsibility that is used to develop the design. OT&E is an independent evaluation of design maturity that is used to determine if the program should proceed to full-rate production. Figure 7-5 lists the major differences between the two.

7.3 SUMMARY POINTS

The Verification activities of the Systems Engineering Process are performed to verify that physical design meets the system requirements.

- DoD T&E policy supports the verification process through a sequence of Developmental, Operational, and Live-Fire tests, analyses, and assessments. The primary management tools for planning and implementing the T&E effort are the TEMP and the integrated planning team.

Development Tests	Operational Tests
• Controlled by program manager • One-on-one tests • Controlled environment • Contractor environment • Trained, experienced operators • Precise performance objectives and threshold measurements • Test to specification • Developmental, engineering, or production representative test article	• Controlled by independent agency • Many-on-many tests • Realistic/tactical environment with operational scenario • No system contractor involvement • User troops recently trained • Performance measures of operational effectiveness and suitability • Test to operational requirements • Production representative test article

Figure 7-5. DT/OT Comparison

CHAPTER 8

SYSTEMS ENGINEERING PROCESS OUTPUTS

8.1 DOCUMENTING REQUIREMENTS AND DESIGNS

Outputs of the systems engineering process consist of the documents that define the system requirements and design solution. The physical architecture developed through the synthesis process is expanded to include enabling products and services to complete the system architecture. *This system level architecture then becomes the reference model for further development of system requirements and documents.* System engineering process outputs include the system and configuration item architectures, specifications, and baselines, and the decision database.

Outputs are dependent on the level of development. They become increasingly technically detailed as system definition proceeds from concept to detailed design. As each stage of system definition is achieved, the information developed forms the input for succeeding applications of the system engineering process.

Architectures: System/Configuration Item

The System Architecture describes the entire system. It includes the physical architecture produced through design synthesis and adds the enabling products and services required for life cycle employment, support, and management. Military Handbook (MIL-HDBK)-881, *Work Breakdown Structures,* provides reference models for weapon systems architectures. As shown by Figure 8-1, MIL-HDBK-881 illustrates the first three levels of typical system architectures. Program Offices can use MIL-HDBK-881 templates during system definition to help develop a top-level architecture tailored to the needs of the specific system considered. The design contractor will normally develop the levels below these first three. Chapter 9 of this text describes the WBS in more detail.

Specifications

A specification is a document that clearly and accurately describes the essential technical requirements for items, materials, or services including the procedures by which it can be determined that the requirements have been met. Specifications help avoid duplication and inconsistencies, allow for accurate estimates of necessary work and resources, act as a negotiation and reference document for engineering changes, provide documentation of configuration, and allow for consistent communication among those responsible for the eight primary functions of Systems Engineering. They provide IPTs a precise idea of the problem to be solved so that they can efficiently design the system and estimate the cost of design alternatives. They provide guidance to testers for verification (qualification) of each technical requirement.

Program-Unique Specifications

During system development a series of specifications are generated to describe the system at different levels of detail. These program unique specifications form the core of the configuration baselines. As shown by Figure 8-2, in addition to referring to different levels within the system hierarchy, these baselines are defined at different phases of the design process.

Initially the system is described in terms of the top-level (system) functions, performance, and interfaces. These technical requirements are derived from the operational requirements established by

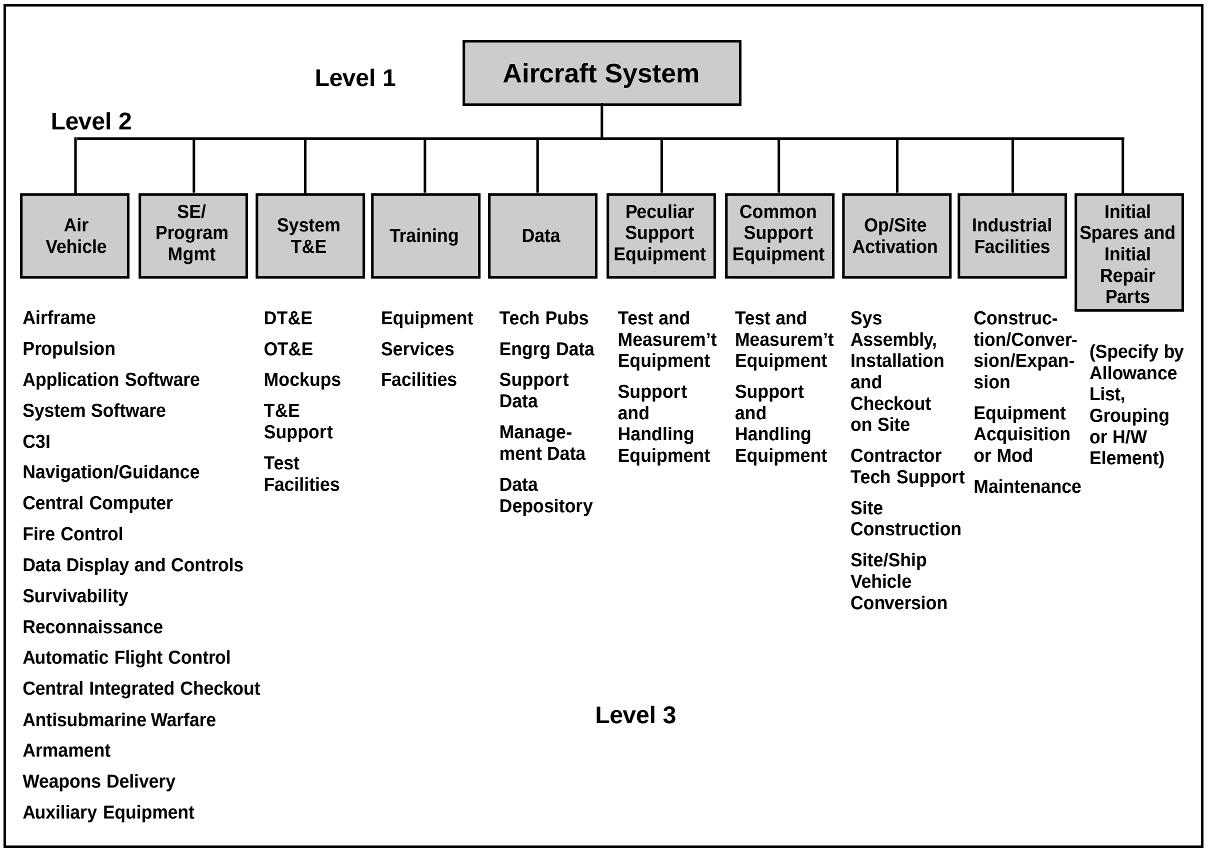

Figure 8-1. Example from MIL-HDBK-881

the user. This system-level technical description is documented in the System Specification, which is the primary documentation of the system-level Functional Baseline. The system requirements are then flowed down (allocated) to the items below the system level, such that a set of design criteria are established for each of those items. These item descriptions are captured in a set of Item Performance Specifications, which together with other interface definitions, process descriptions, and drawings, document the Allocated Baseline (sometimes referred to as the "Design To" baseline). Having baselined the design requirements for the individual items, detailed design follows. Detailed design involves defining the system from top to bottom in terms of the physical entities that will be employed to satisfy the design requirements. When detailed design is complete, a final baseline is defined. This is generally referred to as the Product Baseline, and, depending on the stage of development, may reflect a "Build to" or "As built" description. The Product Baseline is documented by the Technical Data Package, which will include not only Item Detail Specifications, but also, Process and Material Specifications, as well as drawings, parts lists, and other information that describes the final system in full physical detail. Figure 8-3 shows how these specifications relate to their associated baselines.

Role of Specifications

Requirements documents express why the development is needed. Specification documents are an intermediate expression of what the needed system has to do in terms of technical requirements (function, performance, and interface). Design documents (drawings, associated lists, etc.) describe the means by which the design requirements are to be satisfied. Figure 8-4 illustrates how requirements flow down from top-level specifications to design documentation. Preparation of specifications are part of the system engineering process, but also involve techniques that relate to

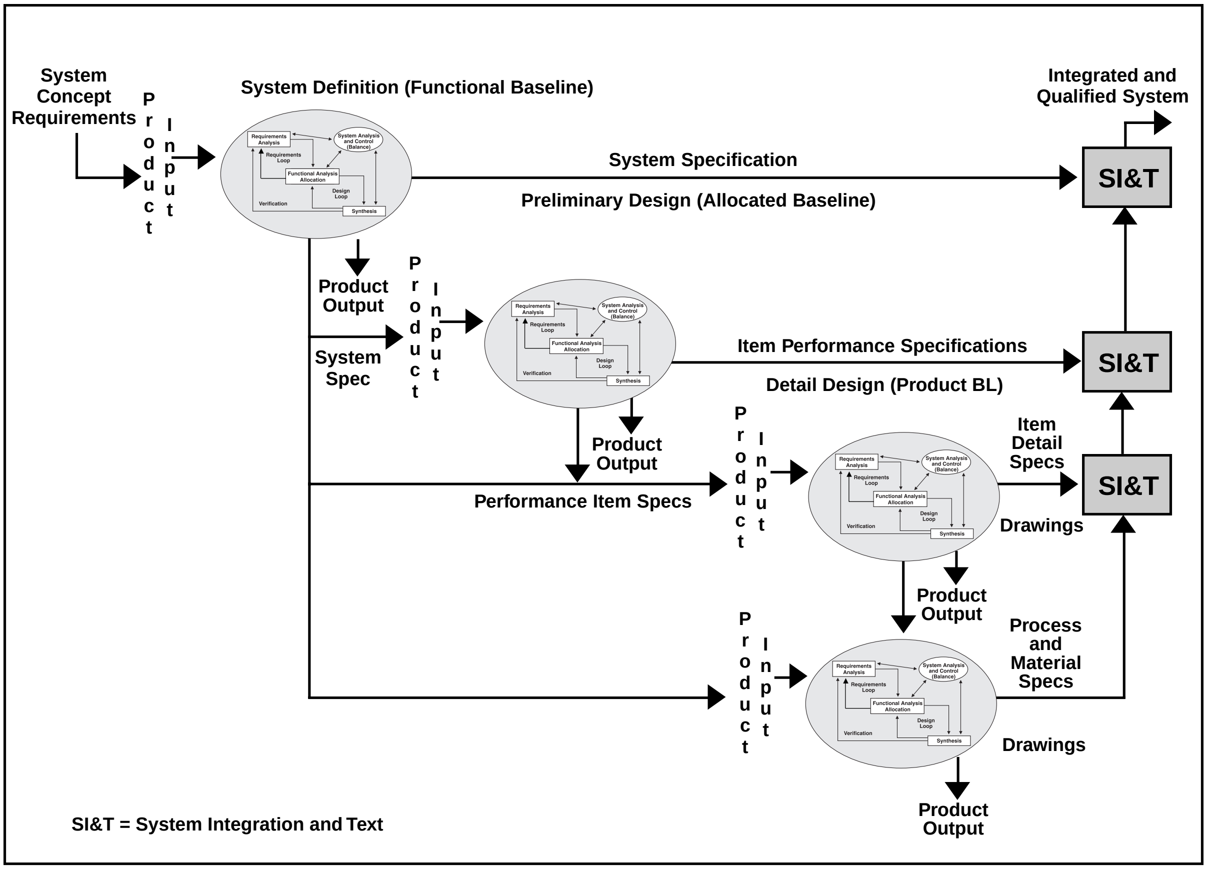

Figure 8-2. Specifications and Levels of Development

Specification	Content	Baseline
System Spec	Defines mission/technical performance requirements. Allocates requirements to functional areas and defines interfaces.	Functional
Item Performance Spec	Defines performance characteristics of CIs and CSCIs. Details design requirements and with drawings and other documents form the Allocated Baseline.	Allocated "Design To"
Item Detail Spec	Defines form, fit, function, performance, and test requirements for acceptance. (Item, process, and material specs start the Product Baseline effort, but the final audited baseline includes all the items in the TDP.)	Product "Build To" or "As Built"
Process Spec	Defines process performed during fabrication.	
Material Spec	Defines production of raw materials or semi-fabricated material used in fabrication.	

Figure 8-3. Specification Types

communication skills, both legal and editorial. Figure 8-5 provides some rules of thumb that illustrate this.

In summary, specifications document what the system has to do, how well it has to do it, and how to verify it can do it.

Baselines

Baselines formally document a product at some given level of design definition. They are references for the subsequent development to follow. Most DoD systems are developed using the three classic baselines described above: functional, allocated, and product. Though the program unique specifications are the dominant baseline documentation, they alone do not constitute a baseline.

Additional documents include both end and enabling product descriptions. End product baseline documents normally include those describing system requirements, functional architecture, physical architecture, technical drawing package, and requirements traceability. Enabling product baseline documents include a wide range of documents that could include manufacturing plans and processes, supportability planning, supply documentation, manuals, training plans and programs, test planning, deployment planning, and others. All enabling products should be reviewed for their susceptibility to impact from system configuration changes. If a document is one that describes a part of a system and could require change if the configuration changes, then most likely it should be included as a baseline document.

Acquisition Program Baselines

Acquisition Program Baselines and Configuration Baselines are related. To be accurate the Program baseline must reflect the realities of the Configuration Baseline, but the two should not be confused. Acquisition Program Baselines are high level assessments of program maturity and viability. Configuration Baselines are system descriptions. Figure 8-6 provides additional clarification.

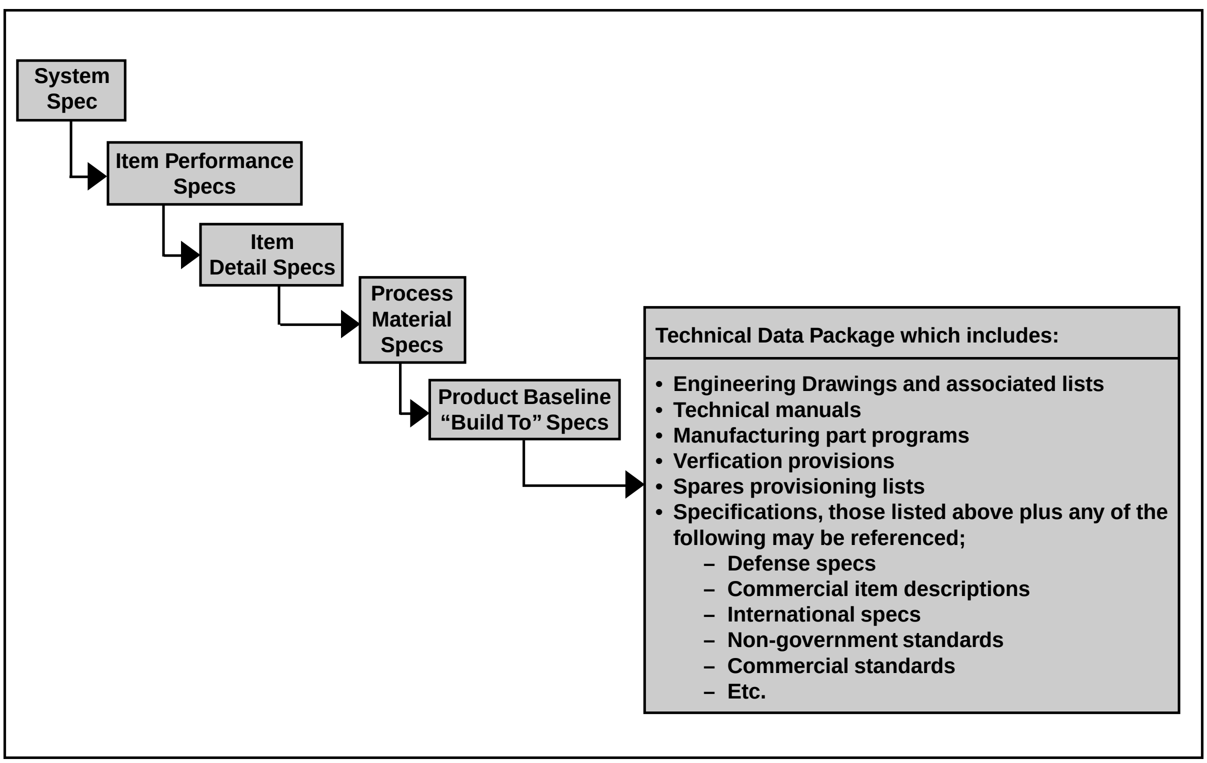

Figure 8-4. How Specifications Lead to Design Documents

- **Use a table of contents and define all abbreviations and acronyms.**
- **Use active voice.**
- **Use "shall" to denote mandatory requirement and "may" or "should" to denote guidance provisions.**
- **Avoid ambiguous provisions, such as "as necessary," "contractor's best practice," "smooth finish," and similar terms.**
- **Use the System Engineering Process to identify requirements. Do not over-specify.**
- **Avoid "tiering." Any mandatory requirement in a document below the first tier, should be stated in the specification.**
- **Only requirement sections of the MIL-STD-491D formats are binding. Do not put requirements in non-binding sections, such as *Scope*, *Documents*, or *Notes*.**
- **Data documentation requirements are specified in a Contract Data Requirements List.**

Figure 8–5. Rules of Thumb for Specification Preparation

Decision Database

The decision database is the documentation that supports and explains the configuration solution decisions. It includes trade studies, cost effectiveness analyses, Quality Function Deployment (QFD) analysis, models, simulations, and other data generated to understand a requirement, develop alternative solutions, or make a choice between them. These items are retained and controlled as part of the Data Management process described in Chapter 10.

8.2 DOD POLICY AND PRACTICE— SPECIFICATIONS AND STANDARDS

DoD uses specifications to communicate product requirements and standards to provide guidance concerning proven methods and practices.

Specifications

DoD uses three basic classifications of specifications: materiel specifications (developed by DoD components), Program-Unique Specifications, and non-DoD specifications.

- **Program Baselines**
  - **Embody only the most important cost, schedule, and performance objectives and thresholds**
  - **Threshold breach results in re-evaluation of program at MDA level**
  - **Selected key performance parameters**
  - **Specifically evolves over the development cycle and is updated at each major milestone review or program restructure**
- **Required on ALL programs for measuring and reporting status**

- **Configuration Baselines**

  Identify and define an item's functional and physical characteristics
  - ***Functional Baseline* – Describes system level requirements**
  - ***Allocated Baseline* – Describes design requirements for items below system level**
  - ***Product Baseline* – Describes product physical detail**
- **Documents outputs of Systems Engineering Process**

Figure 8–6. Acquisition Program Baselines and Configuration Baselines

DoD developed specifications describe essential technical requirements for purchase of materiel. Program-Unique Specifications are an integral part of the system development process. Standard practice for preparation of DoD and Program-Unique Specifications is guided by MIL-STD-961D. This standard provides guidance for the development of performance and detail specifications. MIL-STD-961D, Appendix A provides further guidance for the development of Program-Unique Specifications.

Non-DoD specifications and standards approved for DoD use are listed in the *DoD Index of Specifications and Standards* (DoDISS).

DoD Policy (Specifications)

DoD policy is to develop *performance* specifications for procurement and acquisition. In general, detail specifications are left for contractor development and use. Use of a detail specification in DoD procurement or acquisition should be considered only where absolutely necessary, and then only with supporting trade studies and acquisition authority approval.

DoD policy gives preference to the use of commercial solutions to government requirements, rather than development of unique designs. Therefore, the use of commercial item specifications and descriptions should be a priority in system architecture development. Only when no commercial solution is available should government detail specifications be employed.

In the case of re-procurement, where detail specifications and drawings are government owned, standardization or interface requirements may present a need for use of detailed specifications. Trade studies that reflect total ownership costs and the concerns related to all eight primary functions should govern decisions concerning the type of specification used for re-procurement of systems, subsystems, and configuration items. Such trade studies and cost analysis should be preformed prior to the use of detail specifications or the decision to develop and use performance specifications in a reprocurement.

Performance Specifications

Performance Specifications state requirements in terms of the required results with criteria for verifying compliance, but without stating the methods for achieving the required results. In general, performance specifications define products in terms of functions, performance, and interface requirements. They define the functional requirements for the item, the environment in which it must operate, and interface and interchangeability characteristics. The contractor is provided the flexibility to decide how the requirements are best achieved, subject to the constraints imposed by the government, typically through interface requirements. System Specifications and Item Performance Specifications are examples of performance specifications.

Detail Specifications

Detail Specifications, such as Item Detail, Material and Process Specifications, provide design requirements. This can include materials to be used, how a requirement is to be achieved, or how an item is to be fabricated or constructed. If a specification contains both performance and detail requirements, it is considered a Detail Specification, with the following exception: Interface and interchangeability requirements in Performance Specifications may be expressed in detailed terms. For example, a Performance Specification for shoes would specify size requirements in detailed terms, but material or method of construction would be stated in performance terms.

Software Documentation – IEEE/EIA 12207

IEEE/EIA 12207, *Software Life Cycle Processes*, describes the U.S. implementation of the ISO standard on software processes. This standard describes the development of software specifications as one aspect of the software development process.

The process described in IEEE/EIA 12207 for allocating requirements in a top-down fashion and documenting the requirements at all levels parallels the systems engineering process described in this text. The standard requires first that system-level requirements be allocated to software items (or configuration items) and that the software requirements then be documented in terms of functionality, performance, and interfaces, and that qualification requirements be specified. Software item requirements must be traceable to system-level, and be consistent and verifiable.

The developer is then required to decompose each software item into software components and then into software units that can be coded. Requirements are allocated from item level, to component, and finally to unit level. This is the detailed design activity and IEEE/EIA 12207 requires that these allocations of requirements be documented in documents that are referred to as “descriptions,” or, if the item is a “stand alone” item, as “specifications.” The content of these documents is defined in the IEEE/EIA standard; however, the level of detail required will vary by project. Each project must therefore ensure that a common level of expectation is established among all stakeholders in the software development activity.

Standard Practice for Defense Specifications – MIL-STD-961D

The purpose of MIL-STD-961D is to establish uniform practices for specification preparation, to ensure inclusion of essential requirements, to ensure Verification (qualification) methods are established for each requirement, and to aid in the use and analysis of specification content. MIL-STD-961D establishes the format and content of system, configuration item, software, process and material specifications. These Program-Unique Specifications are developed through application of the systems engineering process and represent a hierarchy as shown in Figure 8-7.

Standards

Standards establish engineering and technical limitations and applications for items, materials, processes, methods, designs, and engineering practices. They are “corporate knowledge” documents describing how to do some process or a

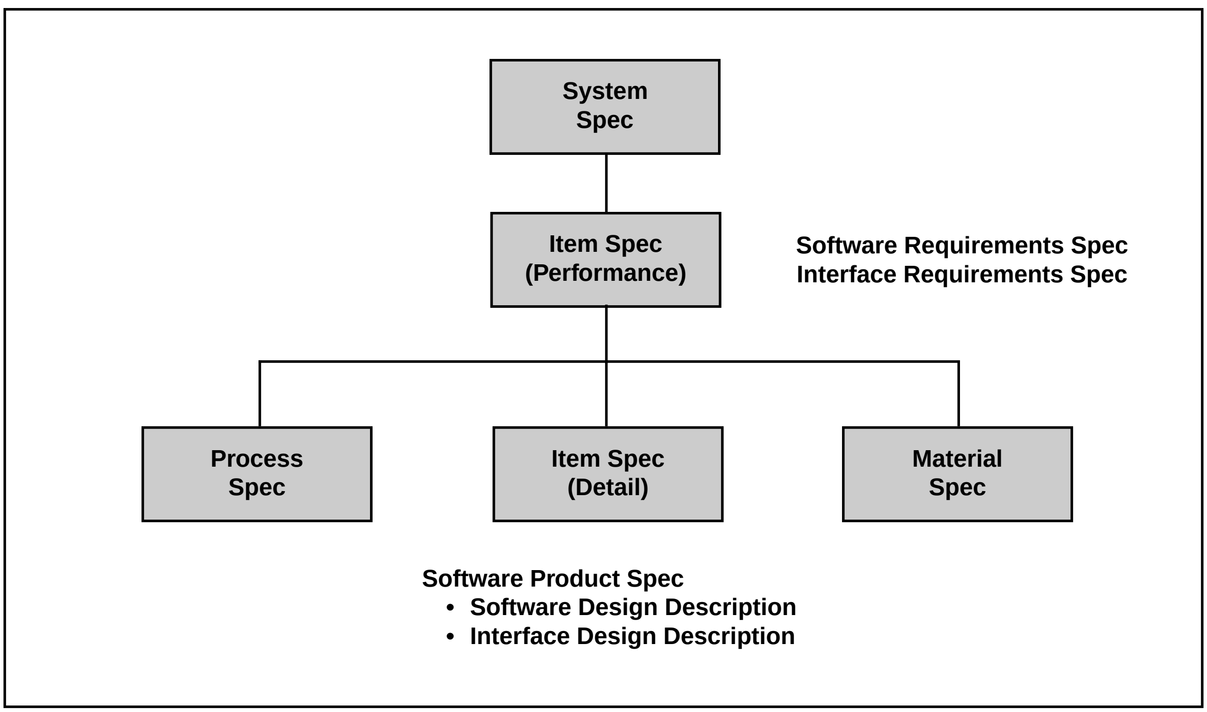

Figure 8–7. Specification Hierarchy

description of a body of knowledge. Standards come from many sources, reflecting the practices or knowledge base of the source. Format and content of Defense Standards, including Handbooks, are governed by MIL-STD-962. Other types of standards in use in DoD include Commercial Standards, Corporate Standards, International Standards, Federal Standards, and Federal Information Processing Standards.

DoD Policy (Standards)

DoD policy does not require standard management approaches or manufacturing processes on contracts. This policy applies to the imposition of both Military Specifications and Standards and, in addition, to the imposition of Commercial and Industry Standards. In general, the preferred approach is to allow contractors to use industry, government, corporate, or company standards they have determined to be appropriate to meet government's needs. The government reviews and accepts the contractor's approach through a contract selection process or a contractual review process.

The government should impose a process or standard only as a last resort, and only with the support of an appropriate trade study analysis. If a specific standard is imposed in a solicitation or contract, a waiver will be required from an appropriate Service authority.

However, there is need on occasion to direct the use of some standards for reasons of standardization, interfaces, and development of open systems. A case in point is the mandated use of the Joint Technical Architecture (JTA) for defining interoperability standards. The JTA sets forth the set of interface standards that are expected to be employed in DoD systems. The JTA is justifiably mandatory because it promotes needed interoperability standardization, establishes supportable interface standards, and promotes the development of open systems.

DoD technical managers should be alert to situations when directed standards are appropriate to their program. Decisions concerning use of directed standards should be confirmed by trade studies and requirements traceability.

DoD Index of Specifications and Standards

The DoDISS lists all international, adopted industry standardization documents authorized for use by the military departments, federal and military specifications and standards. Published in three volumes, it contains over 30,000 documents in 103 Federal Supply Groups broken down into 850 Federal Supply Classes. It covers the total DoD use of specifications and standards, ranging from fuel specifications to international quality standards.

8.3 SUMMARY POINTS

- System Engineering Process Outputs include the system/configuration item architecture, specifications and baselines, and the decision database.
- System/Configuration Item Architectures include the physical architecture and the associated products and services.
- Program-Unique specifications are a primary output of the System Engineering Process. Program-Unique specifications describe what the system or configuration item must accomplish and how it will be verified. Program-Unique specifications include the System, Item Performance, and Item Detail Specifications. The System Specification describes the system requirements, while Item Performance and Item Detail Specifications describe configuration item requirements.
- Configuration baselines are used to manage and control the technical development. Program baselines are used for measuring and supporting program status.
- The Decision Database includes those documents or software that support understanding and decision making during formulation of the configuration baselines.

- DoD policy is to develop *performance* specifications for procurement and acquisition. Use of other than performance specifications in a contract must be justified and approved.

- It is DoD policy not to require standard management approaches or manufacturing processes on contracts.

- Mandatory use of some standard practices are necessary, but must be justified through analysis. A case in point is the mandatory use of the standards listed in the Joint Technical Architecture.

GLOSSARY

SYSTEMS ENGINEERING FUNDAMENTALS

AAAV	Advanced Amphibious Assault Vehicle
ACAT	Acquisition Category
ACR	Alternative Concept Review
AMSDL	Acquisition Management Systems Data List
ASR	Alternative Systems Review
AUPP	Average Unit Procurement Price
AWP	Awaiting Parts
BL	Baseline
BLRIP	Beyond Low Rate Initial Production
C4ISR	Command, ontrol, Communications, Computers, Intelligence, and Reconnaissance
CAD	Computer-Aided Design
CAE	Computer-Aided Engineering
CAIV	Cost As an Independent Variable
CALS	Continuous Acquisition and Life Cycle Support
CAM	Computer-Aided Manufacturing
CASE	Computer-Aided Systems Engineering
CATIA	Computer-Aided Three-Dimensional Interactive Application
CCB	Configuration Control Board
CCR	Contract Change Request
CDR	Critical Design Review
CDRL	Contract Data Requirement List
CDS	Concept Design Sheet
CE	Concept Exploration

CEO	Chief Executive Officer
CI	Configuration Item
Circular A-109	Major Systems Acquisitions
CM	Configuration Management
CM	Control Manager
COTS	Commercial Off-The-Shelf
CSCI	Computer Software Configuration Item
CWI	Continuous Wave Illumination
DAU	Defense Acquisition University
DCMC	Defense Contract Management Command
DDR	Detail Design Review
DFARS	Defense Supplement to the Federal Acquisition Regulation
DID	Data Item Description
DoD	Department of Defense
DoD 5000.2-R	Mandatory Procedures for Major Defense Acquisition Programs (MDAPs), and Major Automated Information System Acquisition Programs (MAIS)
DoDISS	DoD Index of Specifications and Standards
DSMC	Defense Systems Management College
DT	Developmental Testing
DTC	Design To Cost
DT&E	Developmental Test and Evaluation
EC	Engineering Change
ECP	Engineering Change Proposal
EDI	Electronic Data Interchange
EIA	Electronic Industries Alliance
EIA IS 632	Electronic Industries Association Interim Standard 632, on Systems Engineering
EIA IS-649	Electronic Industries Association Interim Standard 649, on Configuration Management
EOA	Early Operational Assessments

FAR Federal Acquisition Regulation

FCA Functional Configuration Audit

FEO Field Engineering Order

FFBD Functional Flow Block Diagram

FIPS Federal Information Processing Standard

FMECA Failure Modes, Effects, and Criticality Analysis

FOT&E Follow-On Operational Test and Evaluation

FQR Formal Qualification Review

GFE Government Furnished Equipment

GFM Government Furnished Material

ICD Interface Control Documentation

ICWG Interface Control Working Group

IDE Integrated Digital Environment

IDEF Integration Definition Function

IDEF0 Integrated Definition for Function Modeling

IDEF1x Integration Definition for Information Modeling

IEEE Institute of Electrical and Electronics Engineers

IEEE/EIA 12207 IEEE/EIA Standard 12207, Software Life Cycle Processes

IEEE P1220 IEEE Draft Standard 1220, Application and Management of the Systems Engineering Process

IFB Invitation for Bid

IIPT Integrating Integrated Product Teams

IMS Integrated Master Schedule

IOC Initial Operational Capability

IOT&E Initial Operational Test and Evaluation

IPPD Integrated Product and Process Development

IPR In-Progress/Process Review

IPT Integrated Product Teams

JASSM	Joint Air-to-Surface Standoff Missile
JROC	Joint Requirements Oversight Council
JTA	Joint Technical Architecture
KPPs	Key Performance Parameters
LFT&E	Live Fire Test and Evaluation
LRU	Line-Replaceable Unit
LRIP	Low Rate Initial Production
M&S	Modeling and Stimulation
MAIS	Major Automated Information System
MAISRC	Major Automated Information Systems Review Council
MBTF	Mean Time Between Failure
MDA	Milestone Decision Authority
MDAP	Major Defense Acquisition Program
MIL-HDBK-61	Military Handbook 61, on Configuration Management
MIL-HDBK-881	Military Handbook 881, on Work Breakdown Structure
MIL-STD 499A	Military Standard 499A, on Engineering Management
MIL-STD-961D	Military Standard 961D, on Standard Practice for Defense Specifications
MIL-STD 962	Military Standard 962, on Format and Content of Defense Standards
MIL-STD-973	Military Standard 973, on Configuration Management
MNS	Mission Need Statement
MOE	Measure of Effectiveness
MOP	Measure of Performance
MOS	Measure of Suitability
MRP II	Manufacturing Resource Planning II
MS	Milestone
MTTR	Mean Time To Repair
NDI	Non-Developmental Item
NIST	National Institute of Standards and Technology

NRTS Not Repairable This Station

OA Operational Assessment
OIPT Overarching Integrated Product Teams
OMB Office of Management and Budget
OPS Operations
ORD Operational Requirements Document
OSD Office of the Secretary of Defense
OT&E Operational Test and Evaluation

P3I Preplanned Product Improvement
PAR Production Approval Reviews
PCA Physical Configuration Audit
PDR Preliminary Design Review
PDRR Program Definition and Risk Reduction
PEO Program Executive Office
PM Program Manager
PME Program/Project Manager – Electronics
PMO Program Management Office
PMT Program Management Team
PPBS Planning, Programming and Budgeting System
PRR Production Readiness Review

QA Quality Assurance
QFD Quality Function Deployment

R&D Research and Development
RAS Requirements Allocation Sheets
RCS Radar Cross Section
RDT&E Research, Development, Test and Evaluation
RFP Request for Proposal

S&T Science and Technology

SBA Simulation Based Acquisition

SBD Schematic Block Diagram

SD&E System Development and Demonstration

SDefR System Definition Review (as referred to in IEEE P1220)

SDR System Design Review

SE Systems Engineering

Section L Instructions to Offerors (Portion of Uniform Contract Format)

Section M Evaluation Criteria (Portion of Uniform Contract Format)

SEDS Systems Engineering Detail Schedule

SEMS Systems Engineering Master Schedule

SEP Systems Engineering Process

SFR System Functional Review

SI Software Item

SI&T System Integration and Test

SOO Statement of Objectives

SOW Statement of Work

SPEC Specification

SSA Source Selection Authority

SSAC Source Selection Advisory Council

SSEB Source Selection Evaluation Board

SSP Source Selection Plan

SSR Software Specification Review

SRR System Requirements Review

SRU Shop-Replaceable Unit

STD Standard

SVR System Verification Review

S/W Software

T&E Test and Evaluation

TDP Technical Data Package

TEMP Test and Evaluation Master Plan

TLS Timeline Analysis Sheet

TOC Team Operating Contract

TPM Technical Performance Measurement

TPWG Test Planning Work Group

TRR Test Readiness Review

VV&A Verfication, Validation, and Accreditation

WIPT Working-Level Integrated Product Team