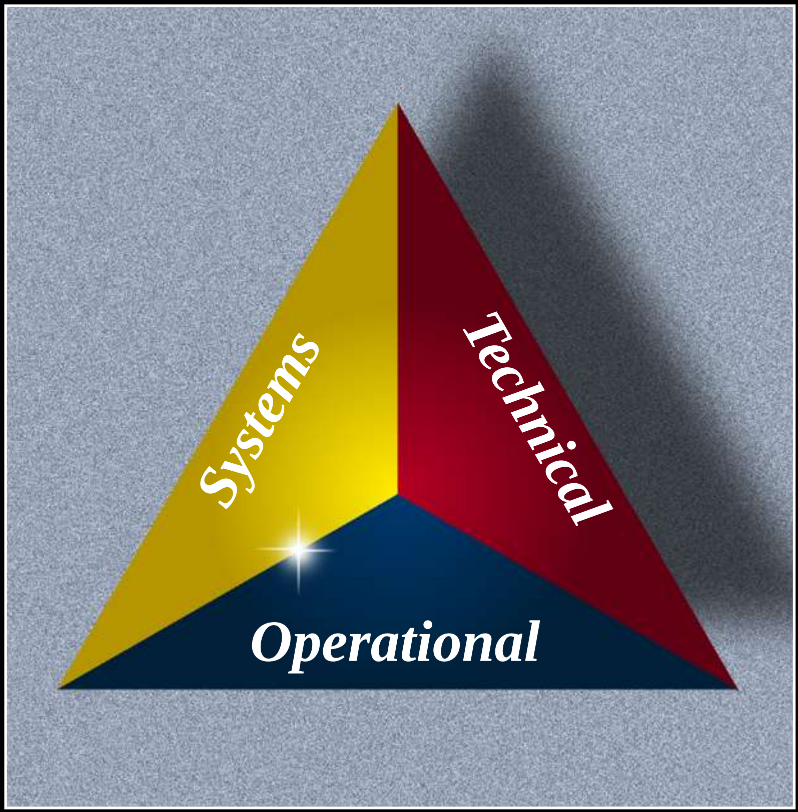
C4ISR Architecture Framework Version 2.0



C4ISR
AWG
Architectures Working Group

18 December 1997

PREFACE

The principal objective of this effort is to define a coordinated approach, i.e., a framework, for Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) architecture development, presentation, and integration. Framework development is an evolutionary process. The first release of a defined framework, the *C4ISR Architecture Framework, Version 1.0*, was developed by the Integrated Architectures Panel of the C4ISR Integration Task Force, and was published 7 June 1996. This report presents Version 2.0 of the Framework. Version 2.0 is an expansion and maturing of concepts presented in Version 1.0, and is based on recent community experience and inputs.

The *C4ISR Architecture Framework* is intended to ensure that the architectures developed by the geographic and functional unified Commands, military Services, and defense Agencies are interrelatable between and among the organizations' operational, systems, and technical architecture views, and are comparable and integratable across Joint and multi-national organizational boundaries.

The *C4ISR Architecture Framework, Version 2.0* was developed under the auspices of the C4ISR Architecture Working Group (AWG), Framework Panel, whose members included representatives from the Joint Staff, the Services, the Office of the Secretary of Defense, and Defense agencies. The Framework Panel was co-chaired by the Space & Naval Warfare Systems Command, Chief Engineer, Architecture and Engineering Directorate (SPAWAR 051), and by the Air Force, Deputy Chief of Staff Communications & Information (AF/SC), Directorate of Architectures and Technology. The Framework Products Work Team was led by the Army, Director of Information Systems for Command, Control, Communications and Computers, Director of Architectures. The Architectures Directorate of the C4I Integration Support Activity (CISA), Office of the Assistant Secretary of Defense for Command, Control, Communications, and Intelligence (OASD[C3I]), with the technical support provided by MITRE, facilitated the coordinated development and evolution of Version 2.0 of the Framework from Version 1.0.

The *C4ISR Architecture Framework, Version 2.0* is a final product of the AWG. The intent is that this product will be accepted by the community and that a memorandum will be promulgated by the Office of the Secretary of Defense designating the *C4ISR Architecture Framework, Version 2.0* as the strategic direction for a DoD Architecture Framework.

Recent government legislation is placing more emphasis on the need to pursue interoperable, integrated, and cost-effective business practices and capabilities within each organization and across DoD, particularly with respect to information technology. Two legislative acts that impact DoD architecture analysis and integration activities are the Information Technology Management Reform Act (ITMRA), also known as the Clinger-Cohen Act of 1996, and the Government Performance and Results Act of 1993 (GPRA). Together, the ITMRA and GPRA serve to codify the efficiency, interoperability, and leveraging goals being pursued by the Commands, Services, and Agencies of DoD.

The ITMRA and the GPRA require DoD organizations to measure the performance of existing and planned information systems and to report performance measures on an annual basis. The *C4ISR Architecture Framework* provides uniform methods for describing information systems and their performance in context with mission and functional effectiveness.

Intentionally Blank

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Intentionally Blank

SECTION 1

INTRODUCTION

"The Defense Science Board and other major studies have concluded that one of the key means for ensuring interoperable and cost effective military systems is to establish comprehensive architectural guidance for all of DoD."

- USD (A&T), ASD (C3I), JS/J6 Memorandum,
Subject: DoD Architecture Coordination Council (ACC), 14 January 1997

1.1 PURPOSE

This report presents Version 2.0 of the Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) Architecture Framework for the development and presentation of architectures. The Framework provides the rules, guidance, and product descriptions for developing and presenting architecture descriptions that ensure a common denominator for understanding, comparing, and integrating architectures. The application of the Framework will enable architectures to contribute most effectively to building interoperable and cost-effective military systems.

Architectures provide a mechanism for understanding and managing complexity. The purpose of C4ISR architectures is to improve capabilities by enabling the quick synthesis of "go-to-war" requirements with sound investments leading to the rapid employment of improved operational capabilities, and enabling the efficient engineering of warrior systems. The ability to compare, analyze, and integrate architectures developed by the geographical and functional, unified Commands, Military Services, and Defense Agencies (hereinafter also referred to as Commands, Services, and Agencies, or C/S/As) from a cross-organizational perspective is critical to achieving these objectives.

The *C4ISR Architecture Framework* is intended to ensure that the architecture descriptions developed by the Commands, Services, and Agencies are interrelatable between and among each organization's operational, systems, and technical architecture views, and are comparable and integratable across Joint and combined organizational boundaries.

This version of the Framework builds on Version 1.0 by specifying an enriched set of products with comparable information content, a data model for representing that information content, and the consistent use of terminology.

1.2 OBJECTIVE AND SCOPE

As implied by the report title, the Framework is currently directed at C4ISR architectures with the focus on C4ISR support to the warfighter. The objective was to develop a common unifying approach for the Commands, military Services, and Defense Agencies to follow in developing their various architectures. While the specific focus has been C4ISR, the approach defined in the

Framework is readily extendible to other DoD functional areas as personnel management, systems acquisition, and finance.

The Framework provides direction on how to *describe* architectures; the Framework does not provide guidance in how to *design or implement* a specific architecture or how to *develop and acquire* systems-of-systems. The distinction between architecture description and architecture implementation is important to understand and is discussed in section 2.

Although the Framework provides a "product-focused" method for standardizing architecture descriptions, the products are intended to represent consistent architectural information. The goal is to eventually reach an "information-focused" method for consistent and integratable architectures. [See section 3.2, section 4.3.1, and appendix B for information on the C4ISR Core Architecture Data Model (CADM), which is intended as a starting point for organizing and portraying the structure of common architecture information.] For Version 2.0 of the Framework, standardizing on architecture products is the only practical approach.

1.3 BACKGROUND

Until recently, there has been no common approach for architecture development and use within the Department of Defense. The individual Commands, Services, and Agencies in DoD traditionally developed their C4ISR architectures using techniques, vocabularies, and presentation schemes that suited their unique needs and purposes. In recent years, National Military Strategy has placed a clearly increasing focus on Joint and multi-national military operations. Moreover, resource reductions and government-wide streamlining and downsizing initiatives have placed a premium on finding opportunities for cross-organization leveraging, increased collaboration, and redefined ways of doing business. Architectures provide a framework for finding these opportunities.

In October 1995, the Deputy Secretary of Defense directed that a DoD-wide effort be undertaken "... to define and develop better means and processes for ensuring that C4I capabilities meet the needs of warfighters." To accomplish this goal, the C4ISR Integration Task Force (ITF) was established under the direction of the Assistant Secretary of Defense for Command, Control, Communications, and Intelligence (ASD [C3I]). This task force, consisting of representatives from the Joint Chiefs of Staff, the military Services, and DoD Agencies, organized itself into sets of panels and subpanels, each charged with tackling a different aspect of the problem.

The Integrated Architectures Panel (IAP) of the ITF provided the foundation for the first version of the Framework by defining three related architecture types: operational, systems, and technical. The *C4ISR Architecture Framework, Version 1.0*, dated 7 June 1996, was developed as a product of the IAP, and was endorsed by the ITF. This initial development of a common approach built upon other architecture efforts within the DoD, as shown in figure 1-1, capitalizing on many of their concepts and ideas. Version 1.0 was intended to provide a basis from which the community could work collectively to evolve and mature architecture development concepts and promulgate them as DoD direction via appropriate DoD policy directives and guidance instructions.

In October 1996, PDASD (C3I) and Joint Staff/J6 established the C4ISR Architecture Working Group (AWG) to continue the effort begun by the IAP. The AWG was charged specifically to review the recommendations of the IAP (which included the Framework) and to develop a DoD-wide implementation strategy. As stated by PDASD (C3I) and Joint Staff/J6...

"We believe that most of the IAP recommendations warrant the eventual mandate of the Deputy Secretary of Defense. However, we think that it is prudent to establish a process in which we assess those recommendations and refine them, if it is necessary, prior to their implementation..."

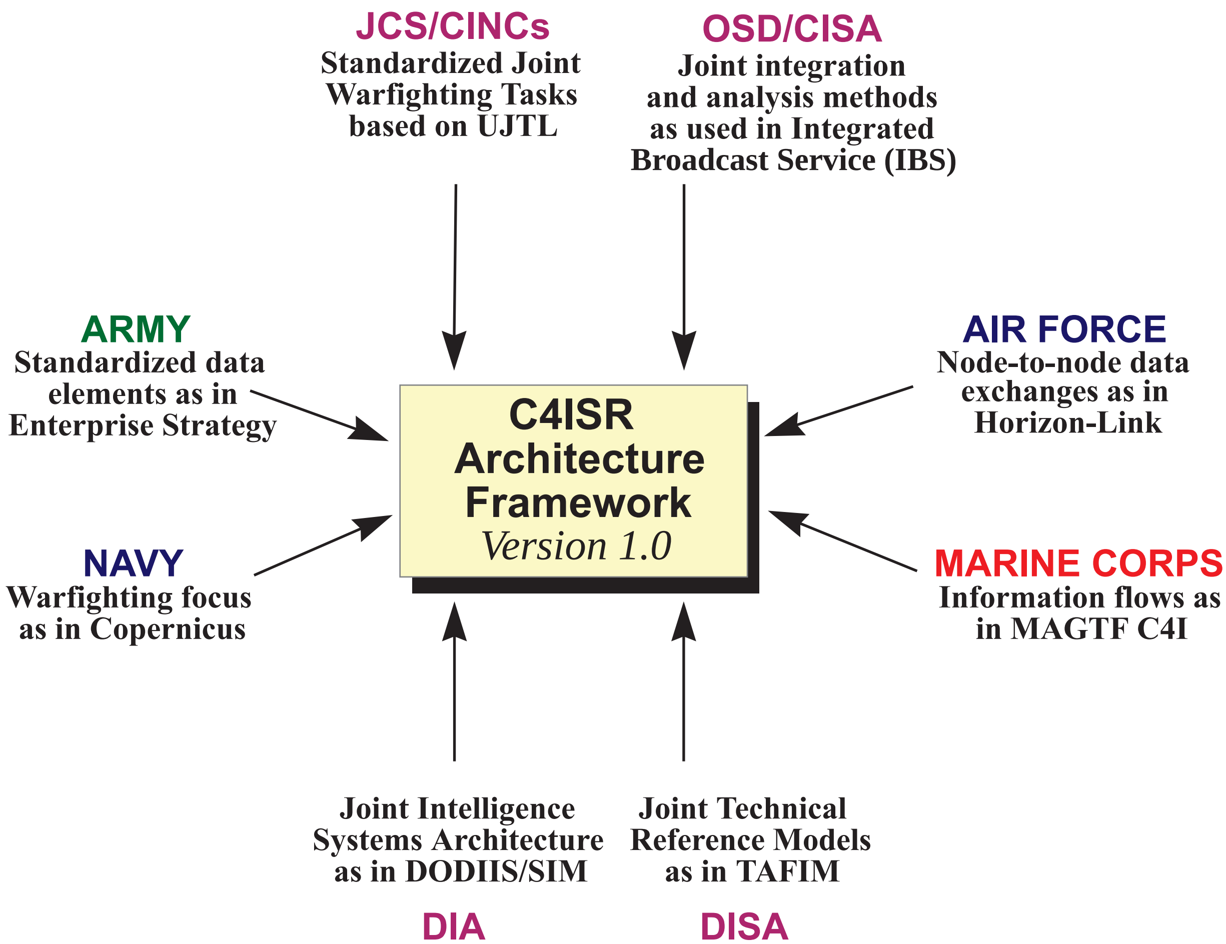


Figure 1-1. Leveraging Prior Efforts

In response, the AWG created and tasked its Framework Panel to develop Version 2.0 of the *C4ISR Architecture Framework*. The Framework Panel was co-chaired by Air Force and Navy representatives and included a Products Work Team led by an Army representative. In addition to the four Services and Command representation, participants included OASD (C3I), OUSD (A&T), DISA, CISA, Joint Staff, JBC, DIA, NIMA, DARO, DoD Space Architect, JTAMDO, BMDO, DMSO, and DoD SIMO.

1.4 ORGANIZATION OF THIS DOCUMENT

The remainder of Version 2.0 is organized as described below.:

Section 2 provides the fundamental definitions, roles, and interrelationships of the operational, systems, and technical architecture views.

Section 3 provides architecture description guidelines. Included are a set of guiding principles, Framework-compliant guidance for building architecture descriptions (including the specific product types required for all architecture descriptions), and a procedure for using the Framework.

Section 4 provides detailed descriptions of the product types that must be used to describe operational, systems, and technical architecture views. Section 4 also provides descriptions of supporting product types , i.e., products that should be used on an as-needed basis.

Five appendices follow the references and glossary. All of the appendices provide additional detail on subjects that are treated at a higher level in the body of the document.

- Appendix A provides detailed tables of the product attributes (information to capture in each product).
- Appendix B provides a mapping of the C4ISR Core Architecture Data Model to a Framework product.
- Appendix C provides a high-level example of a categorization scheme for warfighter information, i.e., instantiations of the information types that are referenced in the information-exchange related products of the Framework.
- Appendix D provides a description of the Levels of Information System Interoperability (LISI) Reference Model.
- Appendix E provides an extract of the relevant portions of the DoD Technical Reference Model (TRM), currently contained within the Technical Architecture Framework for Information Management (TAFIM).

1.5 VERSION 1.0 FEEDBACK AND RESULTING CHANGES IN VERSION 2.0

The overall reaction to the guidance contained within the *C4ISR Architecture Framework, Version 1.0* was quite positive. Most organizations supported the requirement for such guidance, and the consensus was that, if executed properly, it can provide a valuable vehicle for streamlining the architecture process as well as related processes. However, there were a number of suggestions that several organizations submitted with respect to Framework enhancements. Some of the more significant suggestions are described in table 1-1. For a more complete treatment of community lessons-learned, see *C4ISR Architecture Framework, V1.0, Lessons-Learned and Issues for Consideration* (see Sources).

Table 1-1. Some Version 2.0 Major Changes Resulting from Community Feedback *

Community Feedback on Version 1.0	**Resulting Changes Incorporated into Version 2.0**
• **Additional products are needed to describe the systems architecture view**	• **Several additional products are now included (sections 4.2.1 and 4.2.2)**
• **Products should be added that describe behavioral aspects of an architecture (e.g., timing and sequencing of actions)**	• **Accommodated via Rules Model, State Transition Diagrams, & Event Trace Diagrams (section 4.2.2)**
• **Compliance criteria regarding the *Framework* guidance needs to be articulated (i.e., mandatory vs. discretionary)**	• **Distinctions are now made (i.e., essential vs. supporting products) (sections 4.1 and 4.2); in addition, compliance-facilitating principles are also provided (section 3.1.2)**
• **There is some confusion regarding the degree of latitude that can be exercised in interpreting product guidelines**	• **More product examples are now provided to illustrate an acceptable range of product interpretations (sections 4.2.1 and 4.2.2)**
• **There is some confusion regarding products one creates vs. products one consults**	• **Products one consults are now clearly identified as "Universal Reference Resources" (section 4.3)**
• **A few users requested more guidance in "how to build" an architecture description**	• **For these users, more guidance and a flow chart have been included (section 3.2.1)**

* This table attempts to capture the major concerns or suggestions provided by users of Version 1.0. Many other constructive comments were received but not identified here.

Intentionally Blank

SECTION 2

ARCHITECTURE VIEWS — DEFINITIONS, ROLES, AND LINKAGES

The IEEE STD 610.12, as extended slightly by the IAP of the ITF, defines *"architecture"* as "*the structure of components, their relationships, and the principles and guidelines governing their design and evolution over time.*"

The *C4ISR Architecture Framework* provides guidance on describing architectures. An architecture description is a representation, as of a current or future point in time, of a defined "domain" in terms of its component parts, what those parts do, how the parts relate to each other, and the rules and constraints under which the parts function.

What constitutes each of the elements of the above definitions depends on the degree of detail of interest. For example, "domains" can be at any level, from DoD as a whole down to individual functional areas or groups of functional areas. "Component parts" can be anything from "U.S. Air Force" as a component of DoD, down to a satellite ground station as a component of a communications network, or "workstation *A*" as a component of system *x*. "What those parts do" can be as general as their high-level operational concept or as specific as the lowest-level action they perform. "How they relate to each other" can mean something as general as how organizations fit into a very high-level command structure or as specific as what frequency one unit uses in communicating with another. "The rules and constraints under which they work" can mean something as general as high-level doctrine or as specific as the e-mail standard they must use.

It is important to note the difference between an architecture *description* and an architecture *implementation*. As stated above, an architecture description is a representation or "blueprint" of a current or postulated "real-world" configuration of resources, rules, and relationships. Once the blueprint enters the design, development, and acquisition process, the architecture description is then transformed into a real *implementation* of capabilities and assets in the field. The Framework does not address this blueprint-to-implementation transformation process.

Hereinafter in this document, the term "architecture" will be used, in most cases, as a shorthand reference to "architecture description."

2.1 DEFINITIONS OF THE ARCHITECTURE VIEWS

There are three major perspectives, i.e., views, that logically combine to describe an architecture. These three architecture views are the operational, systems, and technical views.

Each of the three architecture views has implications on which architecture characteristics are to be considered and/or displayed, though there is often some degree of redundancy in displaying certain characteristics from one view to another.

Because the views provide different perspectives on the same architecture, it is expected that, in most cases, the most useful architecture description will be an "integrated" one, i.e., one that consists of multiple views. Compared to a single-view architecture description, an integrated architecture description often can provide closer linkage to the planning, programming, and budgeting process and to the acquisition process, and can provide more useful information to those processes.

The definitions and tenets that follow are based on those provided in Version 1.0 of the Framework, modified somewhat to reflect current community thinking.

2.1.1 Definition of the Operational Architecture View

The operational architecture view is a **description of the tasks and activities, operational elements, and information flows required to accomplish or support a military operation**.

It contains descriptions (often graphical) of the operational elements, assigned tasks and activities, and information flows required to support the warfighter. It defines the types of information exchanged, the frequency of exchange, which tasks and activities are supported by the information exchanges, and the nature of information exchanges in detail sufficient to ascertain specific interoperability requirements.

Tenets that apply to the operational architecture view include the following:

- The primary purpose of an operational architecture is to define operational elements, activities and tasks, and information exchange requirements
- Operational architectures incorporate doctrine and assigned tasks and activities
- Activities and information-exchange requirements may cross organizational boundaries
- Operational architectures are not generally systems-dependent
- Generic activity descriptions are not based on an organizational model or force structure
- Operational architectures should clearly identify the time phase(s) covered (e.g., specific years; “as-is” or “to-be;” “baseline,” “planned,” and/or “transitional”).

2.1.2 Definition of the Systems Architecture View

The systems architecture view is a **description, including graphics, of systems and interconnections providing for, or supporting, warfighting functions**.

For a domain, the systems architecture view shows how multiple systems link and interoperate, and may describe the internal construction and operations of particular systems within the architecture. For the individual system, the systems architecture view includes the physical connection, location, and identification of key nodes (including materiel item nodes), circuits, networks, warfighting platforms, etc., and specifies system and component performance parameters (e.g., mean time between failure, maintainability, availability). The systems architecture view associates physical resources and their performance attributes to the operational view and its requirements per standards defined in the technical architecture.

Tenets that apply to the systems architecture include the following:

- The primary purpose of a systems architecture is to enable or facilitate operational tasks and activities through the application of physical resources
- Systems architectures map systems with their associated platforms, functions, and characteristics back to the operational architecture
- Systems architectures identify system interfaces and define the connectivities between systems
- Systems architectures define system constraints and bounds of system performance behavior
- Systems architectures are technology-dependent, show how multiple systems within a subject area link and interoperate, and may describe the internals of particular systems
- Systems architectures can support multiple organizations and missions
- Systems architectures should clearly identify the time phase(s) covered
- Systems architectures are based upon and constrained by technical architectures

2.1.3 Definition of the Technical Architecture View

The technical architecture view is **the minimal set of rules governing the arrangement, interaction, and interdependence of system parts or elements, whose purpose is to ensure that a conformant system satisfies a specified set of requirements.**

The technical architecture view provides the technical systems-implementation guidelines upon which engineering specifications are based, common building blocks are established, and product lines are developed. The technical architecture view includes a collection of the technical standards, conventions, rules and criteria organized into profile(s) that govern system services, interfaces, and relationships for particular systems architecture views and that relate to particular operational views.

Tenets that apply to the technical architecture view include the following:

- Technical architecture views are based on associations between operational requirements and their supporting systems, enabling technologies, and appropriate interoperability criteria
- The primary purpose of a technical architecture is to define the set of standards and rules that govern system implementation and system operation
- A technical architecture profile is constructed from an enterprise-wide set of standards and design rules for specific standards contained in the Joint Technical Architecture and other applicable standards documents
- The technical architecture standards and criteria should reflect multiple information system implementation paradigms
- Technical architecture profiles account for the requirements of multiplatform and network interconnections among all systems that produce, use, or exchange information electronically for a specifically bounded architecture configuration
- Technical architectures must accommodate new technology, evolving standards, and the phasing out of old technology
- Technical architectures should be driven by commercial standards and direction

2.2 REPRESENTATIVE ROLES OF THE OPERATIONAL, SYSTEMS, AND TECHNICAL ARCHITECTURE VIEWS

Warfighter information capabilities must be able to "plug and play" in a Joint, global environment. To achieve this ability, there must be a mechanism for incorporating information technology consistently, controlling the configuration of technical components, and ensuring compliance with technical "building codes." Architectures provide this mechanism.

Architectures are developed according to a defined scope and within a specific context. The scope includes the architecture type, subject area, and time frame for which the architecture is applicable.

In general, the subject area for operational architecture views is based upon mission areas such as Joint Maritime Operations, Mine Warfare, and Theater Air Defense or is based upon operational processes such as Joint planning, air task planning, call for fire, and situational awareness. The interrelated conditions that compose the setting in which the architecture exists constitute the context for the architecture. The context includes such things as doctrine; tactics, techniques, and procedures; relevant goals and vision statements; and concepts of operations, scenarios, and environmental conditions. High-level, broad-scope architectures embrace the range of potential physical, military, and civil environmental conditions so that the resulting architectures are highly stable and are relatively insensitive to moderate changes in environmental conditions. Specific environmental conditions (e.g., threats, weather, geographical features, and scenario) are reflected in operation plans and may also be more directly reflected in lower-level, issue-focused architectures. These specific conditions can be used to enhance operation planning and execution through more concrete planning support and less reactionary operation execution.

In the context of C4ISR architectures, system architecture views are expected to address the full range of systems from sensors that collect information and pass it on, through processing and information systems, communications systems, and shooters that require information to accomplish their objectives. System architecture views depict the functional and physical automated systems, nodes, platforms, communications paths, and other critical elements that provide for supporting information-exchange requirements and warfighter tasks described in the operational architecture views. Various attributes of the systems, nodes, and required information exchanges are included according to the purpose of the specific architecture effort.

Well-planned and comprehensive technical architecture views facilitate integration and promote interoperability across systems and compatibility among related architectures. As part of a disciplined process to build systems, technical architecture views reduce information technology costs across an organization by highlighting risks, identifying technical or programmatic issues, and driving technology reuse. Adherence to a technical architecture streamlines and accelerates systems definition, approval, and implementation.

2.2.1 Role of the Operational Architecture View

The operational architecture view describes the tasks and activities of concern and the information exchanges required. These kinds of descriptions are useful for facilitating a number of actions and assessments across DoD such as examining business processes for reengineering or technology insertion, training personnel, examining doctrinal and policy implications, coordinating Joint and multi-national relationships, and defining the operational requirements to be supported by physical resources and systems, e.g., communications throughput, specific node-to-node interoperability levels, information transaction time windows, and security protection needed.

Operational architecture views are generally independent of organization or force structures. However, for some specific purposes, it may be necessary to document how business processes are performed under current structures in order to examine possible changes to those business processes under a different structure.

Operational architecture views are generally driven by doctrine. However, in some cases, external forces compel an organization to operate in a way that is not compliant with doctrine. In those cases, it may be useful to build an architecture description that shows how the organization really does operate, so its operations can be analyzed and a way can be found either to bring the operations into compliance with doctrine or to present a case to change doctrine. In some cases, actual ("as-is") operations cannot be conducted strictly in conformance with current policy because of inefficiencies induced, for example, by lack of supporting infrastructure or node and information-exchange degradation resulting from threats or acts of nature.

Operational architectures are generally independent of technology. Sometimes, however, operations and their relationships may be influenced, or "pushed," by new capabilities such as collaboration technology, where process "improvements" are in practice before policy can reflect the new procedures. There may be some cases, as well, in which it is necessary to document the way processes are performed given the restrictions of current systems, in order to examine ways in which new systems could facilitate streamlining the processes.

Operational architecture views can describe activities and information exchange requirements at any level of detail and to any breadth of scope that is appropriate for the use or purpose at hand. It may be necessary to show only broad functional areas, in which case the information exchanges would be depicted at a commensurately high level. At a lower level of detail, for a different purpose, it may be necessary to show specific node-to-node information exchanges and the details of the exchanges if articulating interoperability-level distinctions and requirements is the focus. At an even lower level of detail, for still another purpose, it may be necessary to show how specific information supports a specific unit during particular circumstances, such as how specific information supports the Theater Joint Intelligence Center (JIC) during a type-three contingency in the Southwest Asian Theater.

2.2.2 Role of the Systems Architecture View

JCS Pub 1-02, 23 March 1994, defines "system" as "any organized assembly of resources and procedures united and regulated by interaction or interdependence to accomplish a set of specific functions." In the context of the Framework, a "system" may be a partially or fully automated system, or may be a non-automated system, such as some weapon systems.

The systems architecture view describes the systems of concern and the connections among those systems in context with the operational architecture view. The systems architecture view may be used for many purposes, including, for example, systems baselining, making investment decisions concerning cost-effective ways to satisfy operational requirements, and evaluating interoperability improvements. A systems architecture view addresses specific technologies and "systems." These technologies can be existing, emerging, planned, or conceptual, depending on the purpose that the architecture effort is trying to facilitate (e.g., reflection of the "as-is" state, transition to a "to-be" state, or analysis of future investment strategies).

For many purposes, a systems architecture view will need to take the information exchanges described in the operational view down a level in order to translate node-to-node exchanges into system-to-system transactions, communications capacity requirements, security protection needs, et cetera. For other purposes, it may be necessary to go further and to break these system-to-system information exchanges down into the applications that support the production and transmission of specific data elements of those exchanges. For the latter case, an information model at a corresponding level of detail would be useful, specifically, one that includes the applications and their attributes and relationships.

An important point to make here is that, oftentimes, the degree of granularity of the operational architecture view should be driven by the type of systems analysis or assessments that are of interest. Since examination of current and postulated system characteristics must be performed in context with operational missions and requirements in order to have real meaning, then the nature of the systems investigation dictates which operational requirements attributes need to be articulated. Figure 2-1 illustrates this point.

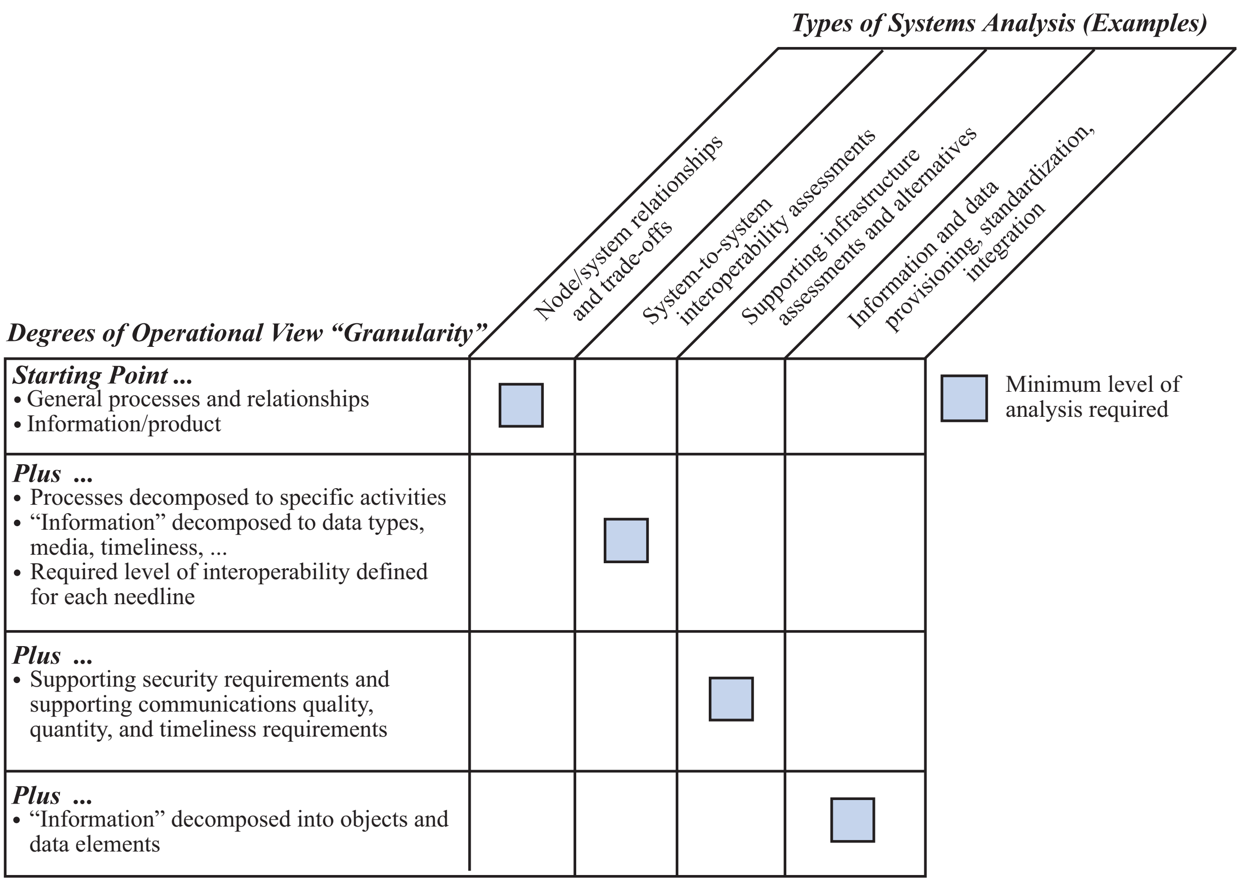


Figure 2-1. Operational Architecture Granularity Required for Systems Analyses

2.2.3 Role of the Technical Architecture View

The technical architecture view describes a profile of a minimal set of time-phased standards and rules governing the implementation, arrangement, interaction, and interdependence of system elements. The appropriate use of the technical architecture view is to promote efficiency and interoperability, and to ensure that developers can adequately plan for evolution.

There are a number of existing technical references such as the Joint Technical Architecture (JTA), the Levels of Information Systems Interoperability (LISI), and numerous policies, directives, and conventions, in addition to Service-level and Agency-level technical architectures. In many cases, an effort to develop a technical architecture view consists of extracting the portions of these sources that are applicable to the scope of the architecture description being developed, and tailoring their guidance to the purpose at hand.

With respect to system-to-system interoperability, the technical architecture view delineates the technical implementation criteria or "rules" with which the system(s) should comply as reflected in the systems architecture view.

2.3 LINKAGES AMONG THE ARCHITECTURE VIEWS

To be consistent and integrated, an architecture description must provide explicit linkages among its various views. Such linkages are also needed to provide a cohesive audit trail from integrated mission operational requirements and measures of effectiveness (MOEs) to the supporting systems and their characteristics, and to the specific technical criteria governing the acquisition/development of the supporting systems.

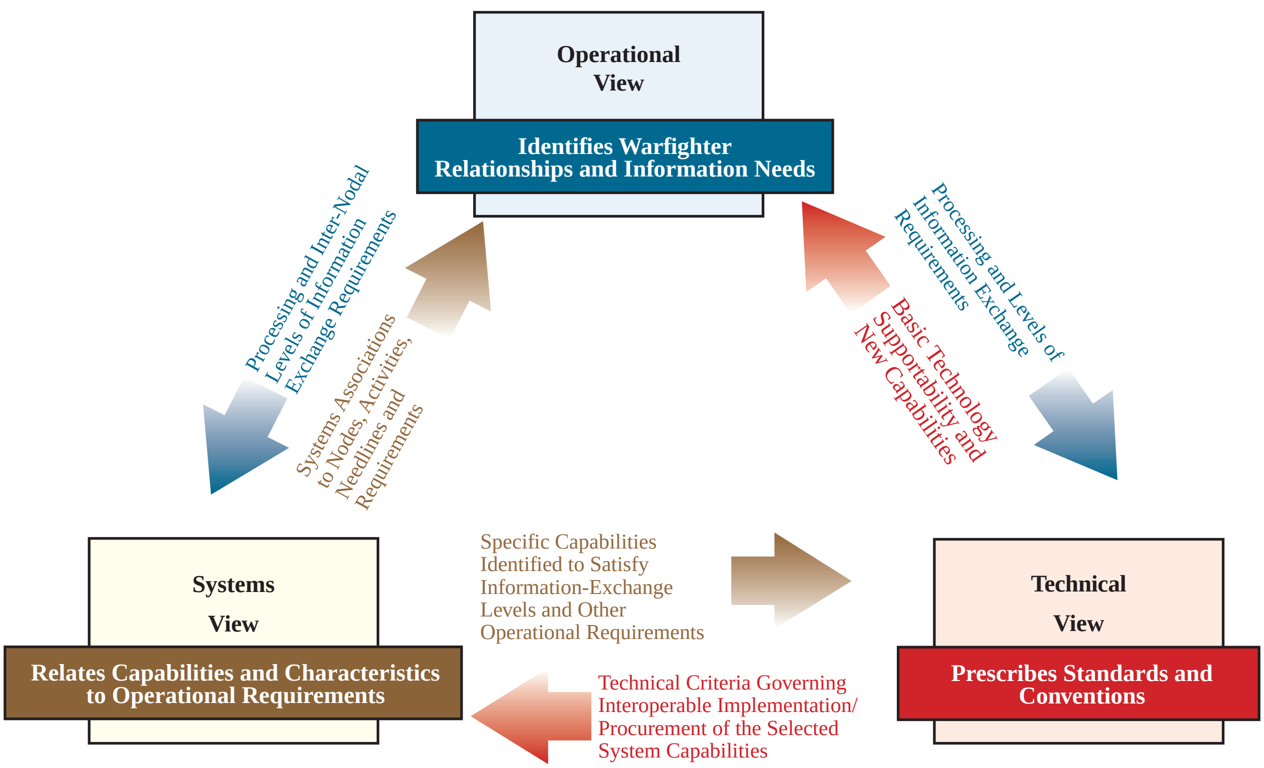


Figure 2-2. Fundamental Linkages Among the Views

Figure 2-2 illustrates some of the linkages that serve to describe the interrelationships among the three architecture views. "Interoperability" is a typical architecture focus that demonstrates the criticality of developing these inter-view relationships.

The operational view describes the nature of each needline's information exchange in detail sufficient to determine what specific degree of information-exchange interoperability is required. The systems view identifies which systems support the requirement, translates the required degree of interoperability into a set of system capabilities needed, and compares current/postulated implementations with the needed capabilities. The technical view articulates the criteria that should govern the compliant implementation of each required system capability.

The ITMRA requires organizations to define measures of performance for evaluating the impact and progress of their information systems. Integrated architecture descriptions (those that consist of more than one view) are essential to meet this requirement. For example, systems and/or system attributes (identified in the systems architecture view) and their "measures of performance" must be assessed with respect to the utility they provide to the missions (identified in the operational architecture view in terms of "measures of effectiveness"). Similarly, systems must be assessed with respect to the standards and conventions that apply (identified in the technical architecture view).

As the reader will see in section 4, the operational architecture description provides detail regarding the information-exchange, interoperability, and performance parameters required to support a particular mission and task. The systems architecture description defines system attributes, and provides the basis for comparing system performance against operational requirements. The technical architecture description defines the specific implementation criteria that will result in the fielding of an interoperable system. Thus, the descriptions of the three architecture views and their interrelationships provide the basis for deriving measures such as interoperability or performance and also provide the basis for measuring the <u>impact</u> of these metrics on operational mission and task effectiveness.

SECTION 3

ARCHITECTURE GUIDELINES, PROCESS, AND FACILITATORS

3.1 ARCHITECTURE GUIDELINES

The C4ISR Architecture Framework contains four main types of guidance for the architecture development process: (1) guidelines, which include a set of guiding principles and guidance for building architectures that are compliant with the Framework, (2) a process for using the Framework to build and integrate architectures, (3) a discussion of architecture data and tools that can serve as facilitators of the architecture-description process, and (4) a detailed description of the product types. This section discusses the first three of these aspects of Framework guidance; section 4 describes the product types.

3.1.1 Guiding Principles

The following set of guiding principles for building architectures is critical to the objectives of the guidance.

- ***Architectures should be built with a purpose in mind.*** Having a specific and commonly understood purpose before starting to build an architecture greatly increases the efficiency of the effort and the utility of the resulting architecture. The purpose determines how wide the scope needs to be, which characteristics need to be captured, and what timeframes need to be considered. This principle applies equally to the development of an architecture as a whole and to the development of any portion or view of an architecture. This principle can also be said to apply to groups of architectures. If groups of architectures built by various organizations are to be compared, it is important that they all be built from the start with the purpose of comparison in mind.

- ***Architectures should facilitate, not impede, communication among humans.*** Architectures must be structured in a way that allows humans to understand them quickly and that guides the human thinking process in discovering, analyzing, and resolving issues. This means that extraneous information must be excluded and common terms and definitions must be used. Often, graphical formats are best for rapid human understanding, but the appropriate format for a given purpose must be used, whatever that format may be.

- ***Architectures should be relatable, comparable, and integratable across DoD.*** Like the principle above, this principle requires the use of common terms and definitions. This principle also requires that a common set of architectural "building blocks" is used as the basis for architecture descriptions. For example, a likely candidate as a starting point for warfighter and warfighter-support tasks (from which activities can be derived) is the Universal Joint Task List (UJTL). The universal reference resources identified in section 4.3 provide documentation concerning common terms, pick-lists, and structures. This principle also dictates that products of a given type that are developed for different architectures must display similar types of information about their respective domains, in similar formats. (This is discussed further in section 4.)

- ***Architectures should be modular, reusable, and decomposable***. Architecture representations should consist of separate but related pieces that can be recombined with a minimum amount of tailoring, so that they can be used for multiple purposes.

The set of products to be built, the characteristics to capture in those products, and high-level steps for using the Framework have been designed to ensure that the above principles are followed.

3.1.2 Framework Compliance Guidance

The paragraphs below provide guidance concerning how to be compliant with Version 2.0 of the Framework. As was mentioned earlier, the *future* direction of DoD architecture descriptions is toward an information-focused approach rather than a product-focused approach. However, given that sufficient commonality in information does not yet exist, a logical interim step is to facilitate human understanding of architectures by providing common representational formats (products) and by specifying the information to be captured in each product. The following paragraphs describe compliance with Version 2.0 of the Framework.

In order to comply with the Framework, architectures must:

- Provide the specified, minimum set of essential products
- Use specified standardized supporting products when needed
- Use the common terms and definitions as specified in this document
- Describe Joint and multi-national relationships in a standard way
- Describe interoperability requirements in a standard way

3.1.2.1 Build the Essential Products

All architectures, whatever their purpose, should include all <u>essential</u> products (defined in section 4) that are pertinent to each and all views (operational, systems, and technical) that need to be developed. The essential products portray the basic information and relationships that constitute an architecture and that are necessary for the integration of multiple architectures from a cross-organizational perspective. Architectures should identify each product by the name specified in section 4, and capture the information specified in section 4 and appendix A.

An architecture that requires only an operational view for its specified purpose may not be required to include system-specific products. Similarly, an architecture that requires operational and high-level system views for its particular purpose may not require standards-specific (i.e., technical) products.

3.1.2.2 Use Standardized Supporting Products When Needed or Helpful

In addition to the essential products, architectures should include products selected from the set of <u>supporting</u> products, described in section 4, as needed to achieve the specific objectives of the architecture. As with the essential products, supporting products should be identified by the name specified in section 4, and capture the information specified in section 4 and appendix A.

The decision of which products to build, beyond the essential set, must be made based on the issues the architecture is intended to explore and the resulting characteristics that the architecture must capture. A given architecture may contain all of the supporting products, a selected subset, or none of the supporting products. For example, an architecture that is to be used in business process reengineering should include an Activity Model; an architecture that is to be used in examining technology insertion and achievable states of "to-be" capabilities should include a System Technology Forecast.

The combined set of essential and supporting products defined in section 4 captures the products most commonly used in architectures. However, the products presented in this document are not an exhaustive set of products that may be used. Architectures may include other products, in addition to the essential product set and relevant supporting products, as necessary to meet their specific objectives. Additional products, as recommended by architecture developers, will be considered for inclusion in future versions of the Framework.

3.1.2.3 Use Common Terms and Definitions

Architectures should use common and/or standardized terms and definitions.

The use of common terms with universally understood definitions continues to be a goal for architecture descriptions. This version of the Framework does not attempt to provide the definitive set of terms that must be used in all architectures. However, the Framework does provide a limited set of critical definitions. More extensive lists and definitions of common terms are more appropriately contained in approved Joint dictionaries and data models such as those referenced in section 4.3, *Universal Reference Resources*. One such model currently being developed is the C4ISR Core Architecture Data Model discussed in section 3.3. Because one of the aims of the Framework is to promote common understanding of architectures and their descriptions, the Framework does require that every architecture contain an *Integrated Dictionary*, which defines terms used in the other products. The *Integrated Dictionary* is described in section 4.2.

3.1.2.4 Describe Joint and Multi-National Relationships in a Standard Way

Architectures should clearly describe external interfaces with Joint and multi-national components in a manner consistent with the method used to describe internal relationships.

One of the Framework's guiding principles states that architectures should be relatable, comparable, and integratable across DoD. Much of the Framework's guidance serves that principle, e.g., common descriptive products, common characteristics to be shown in each product type, the use of common terms and definitions where possible, and the use of a common functional basis for architectures. However, one more critical piece of information needs to be captured in all architectures so that they will be integratable from a Joint perspective, namely, clear *descriptions of each individual node's Joint and multi-national relationships.*

Another of the Framework's guiding principles states that architectures should be built with a purpose in mind, and that information gathering and representation should be limited to what is needed for that purpose. However, *every* architecture, at whatever level of organizational hierarchy, has an implicit purpose in addition to its organization-specific purpose. That implicit purpose is to contribute to the analysis of DoD interoperability and the potential leveraging or sharing of capabilities across Joint boundaries. Some issues that continually confront cross-organizational architecture analyses include aligning and interrelating architecture segments, assuring correct and commonly understood interfaces across the boundaries, and identifying opportunities for integration.

Descriptions of Joint and multi-national relationships may not be needed to satisfy a specific organization's purpose, but they will always be needed to support Joint and/or multi-national integration analyses.

3.1.2.5 Describe Interoperability Requirements in a Standard Way

Architectures should capture specific interoperability requirements in a standard way (content and form). Architects must also ensure that these requirements and the system and technical "responses" are clearly related to each other across the three architecture views and their related products.

These standard characteristics are included in section 4 and appendix A in the specification of the kinds of information to be captured in each product. The *Levels of Information System Interoperability (LISI),* described in the Interoperability Panel Report of the Architecture Working Group (referenced in section 4), represents one approach to satisfy this compliance guideline.

3.2 ARCHITECTURE DESCRIPTION PROCESS

This section discusses ways to apply the Framework in building and integrating architectures.

3.2.1 The Six-Step Architecture Description Process

The fundamental steps to building an architecture in accordance with the Framework are briefly described below, in the general sequence in which they often will be performed, along with some discussion of the significance of each step. Figure 3-1 depicts the six-step process for developing architectures.

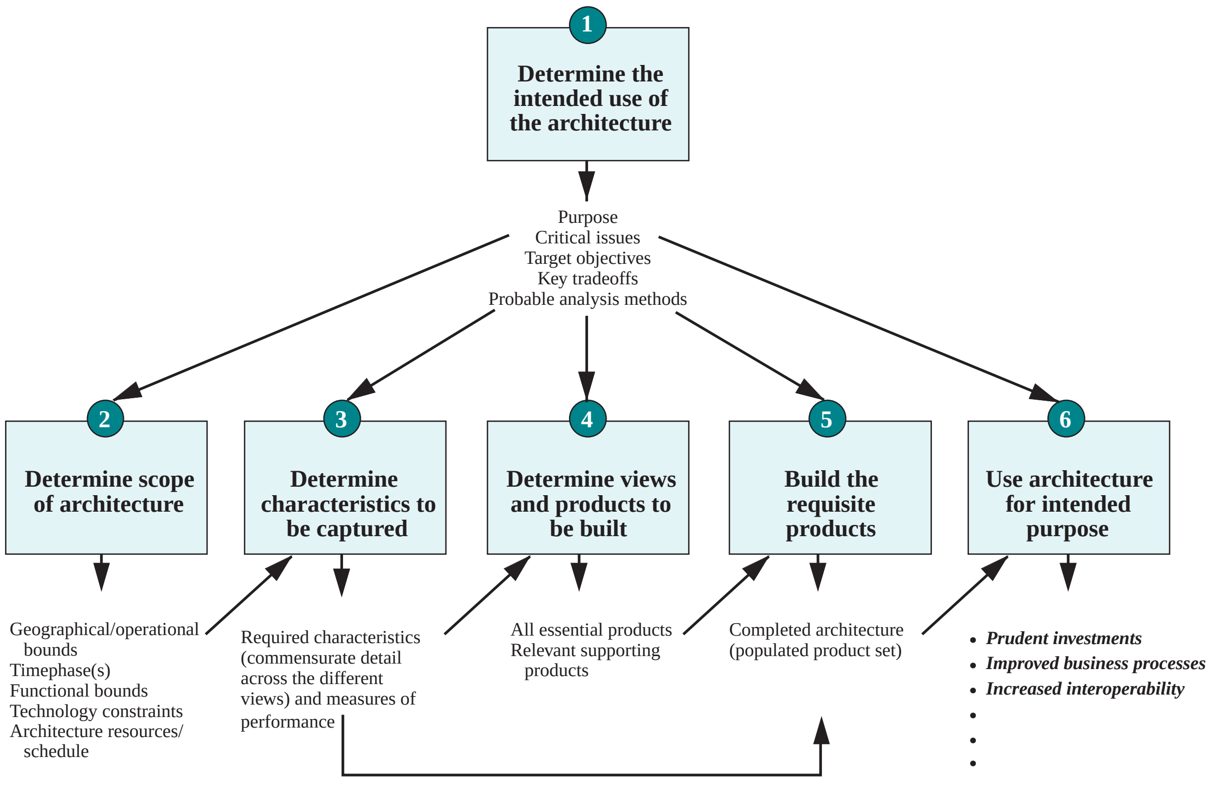


Figure 3-1. The Six-Step Process of Building an Architecture

Step 1: Determine the intended use of the architecture. In most cases, there will not be enough time, money, or resources to build top-down, all-inclusive architectures. Architectures should be built with a specific purpose, whether the intent is business process reengineering, system acquisition, system-of-systems migration or integration, user training, interoperability evaluation, or any other intent. Before beginning to describe an architecture, an organization must determine as specifically as possible the issue(s) the architecture is intended to explore, the questions the architecture is expected to help answer, and the interests and perspectives of the audience and users. In addition, the types of analysis that are expected to be performed must be considered; for example, knowing that the architecture may be used as input to specific models or simulations can affect what is included and how the products are structured. This focusing will make the architecture-development effort more efficient and the resulting architecture more appropriately balanced and useful.

Step 2: Determine the architecture's scope, context, environment, and any other assumptions to be considered. Once the purpose or use has been decided, the prospective content of the architecture can be determined. Items to be considered include, but are not limited to, the scope of the architecture (activities, functions, organizations, timeframes, etc.); the appropriate level of detail to be captured; the architecture effort's context within the "bigger picture;" operational scenarios, situations and geographical areas to be considered; the projected economic situation; and the projected availability and capabilities of specific technologies during the timeframe to be depicted. Project-management factors that contribute to the above determinations include the resources available for building the architecture as well as the resources and level of expertise available for analyzing the architecture.

Step 3: Based on the intended use and the scope, determine which characteristics the architecture needs to capture. Care should be taken to determine which architecture characteristics will need to be described to satisfy the purpose of the architecture. If pertinent characteristics are omitted, the architecture may not be useful; if unnecessary characteristics are included, the architecture effort may prove infeasible given the time and resources available, or the architecture may be confusing and/or cluttered with details that are superfluous to the issues at hand. Care should be taken as well to predict the future uses of the architecture so that, within resource limitations, the architecture can be structured to accommodate future tailoring, extension, or reuse.

Step 4: Based on the characteristics to be displayed, determine which architecture views and supporting products should be built. Depending on steps one through three, it may not be necessary to build the complete set of architecture views and supporting products. Beyond the essential products that must be built for all architectures, only those supporting products that portray the required characteristics should be built.

Step 5: Build the requisite products. The obvious next step is to build the required set of architecture products, which consists of the essential products, the needed supporting products, and individually-defined products driven by architecture specific needs.. To facilitate integration with other architectures, it is critical to include all depictions of relationships with applicable Joint and multi-national components. If the architecture needs some re-tailoring to serve its purpose, that tailoring should be done as efficiently as possible. In this regard, it may be useful, resources permitting, to conduct some proof-of-principle analysis of the architecture at various stages of its development, i.e., make trial runs of step six using carefully selected subsets of the areas to be analyzed. Care should be taken to ensure that the products built are consistent and properly interrelated.

Step 6: Use the architecture for its intended purpose. The architecture will have been built with a particular purpose in mind. As stated in the discussion of step one, the ultimate purpose may be to redesign operational processes, to consolidate and streamline systems, to provide documentation for training personnel, to support the need for proposed systems, or some other purpose. It must be emphasized that the architecture facilitates and enables these purposes but does not itself provide conclusions or answers. For that, human and possibly automated analysis must be applied. The Framework does not attempt to dictate how this analysis should be performed; rather, the Framework intends to promote architectures that are sufficiently complete, understandable, and integratable to serve as one basis for such analysis.

3.2.2 Considerations for Integrating Architectures

Enabling the integration of multiple architectures is an important role of the Framework. Many organizations are already using the Framework to integrate architectures within their individual domains. The basic Framework principles and guidelines have been used in recent years by CISA and the Intelligence Systems Secretariat (ISS) to focus on selected Joint issues and consolidation opportunities. The Joint Staff has recently undertaken an effort to use the Framework for constructing the Joint Operational Architecture (JOA).

3.2.2.1 Degrees of Integration

To say that architectures must be "integratable" implies varying degrees of cross-architecture integration. At the low end, integration means that various architectures (whether prepared by one organization or many organizations) have a "look, touch, and feel" that is sufficiently similar to enable critical relationships to be identified, thereby at least setting the stage for further investigation. At the high end, integration means that various architectures can be intertwined, or plugged together, into a single logical and physical representation.

Today, and in the near future, architecture integration will probably be accomplished toward the lower end of the integration continuum. This level of integration is often satisfactory, depending on the focus of the architecture integration initiative. As universal data models and standard data structures and elements emerge, integration toward the high end of the continuum will be facilitated. However, it is debatable whether "plug-and-play" integration will ever be achievable without the need for some level of manual coordination and "deconfliction," simply because different organizations tend to have unique perspectives on how they interact with each other. In addition, unless all organizations are focused on the same missions, activities, scenarios, timeframes, etc., there will be a lack of a "common denominator" for easily reconciling conflicts among the various architectures.

3.2.2.2 Integration Dimensions

There are four dimensions of architecture integration that represent varying degrees of integration scope. Figure 3-2 illustrates these four dimensions in context with a global, hierarchical view of warfighter operations and support. Note that the need to integrate multiple architecture views and descriptions is certainly not limited to Joint or cross-organizational considerations. The Framework is intended to facilitate all four integration dimensions.

A first dimension involves a single organization and its operations within a single "echelon." In the example shown, the focus is on Army operations at the tactical level (echelon). In addition to the obvious need to interrelate the three views (and associated products) of an Army tactical architecture, in this case there may be multiple architectures — at the same echelon — that cover different functional areas or viewpoints that need to be interrelated, depending on the purpose and scope of the initiative. For example, the Army may be investigating more cost-effective means of providing logistics support to troops in the field. This may involve integrating the architecture views that reflect a warfighting perspective with the views reflecting a logistics-support perspective to assess tradeoffs between C4ISR and logistics investment options.

A second dimension illustrated in figure 3-2 still involves a single organization (Army), but the integration scope expands vertically to include Army operations across multiple echelons. In this particular case, the organization may be examining opportunities to streamline its operations or investments from top to bottom.

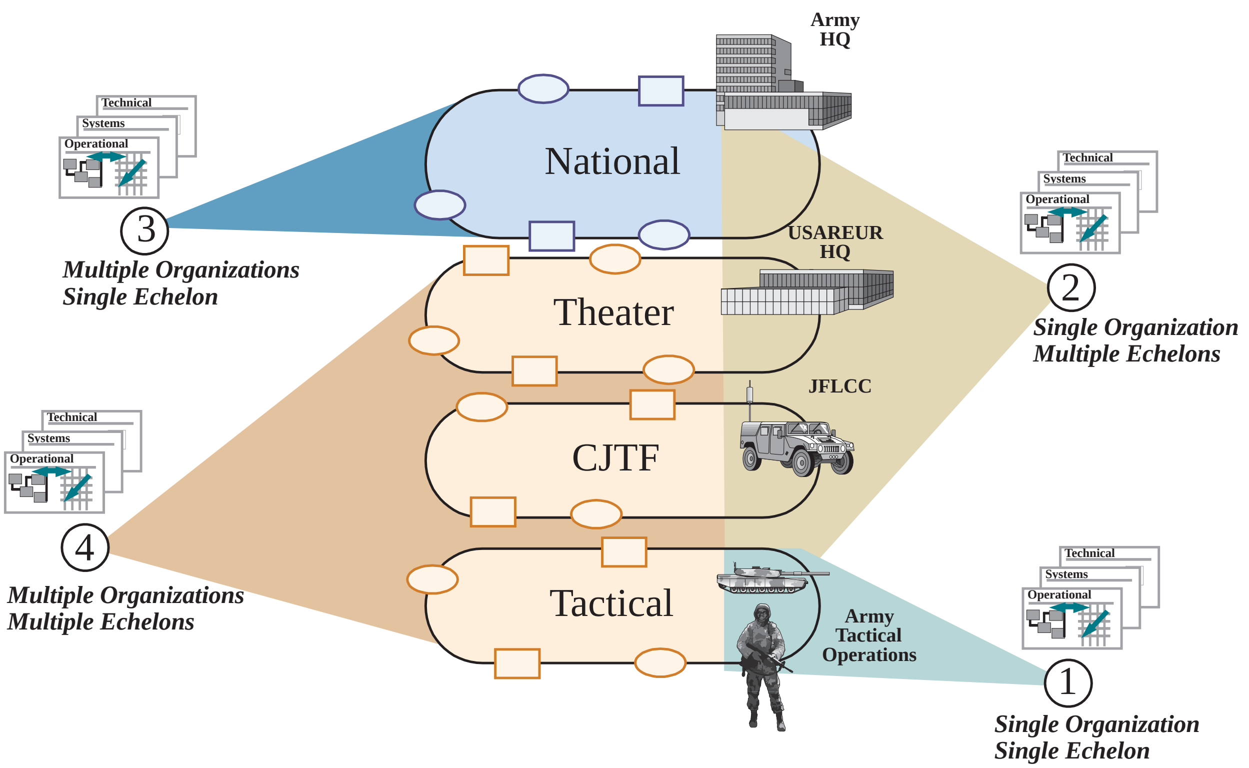


Figure 3-2. Four Dimensions of Architecture Integration

A third integration dimension involves architecture initiatives that cross-cut multiple organizations (U.S. and/or multi-national) horizontally, within a single echelon. An example of this dimension is an architecture whose objective is to investigate opportunities for the various components of DoD to exploit or leverage National information infrastructure capabilities.

A fourth dimension of integration involves multiple organizations and multiple echelons, where vertical and horizontal Joint relationships need to be articulated and examined. An example of this dimension is an architecture whose focus is on assessing the effectiveness of intelligence information support to the warfighter. This could involve examining tradeoffs between hierarchical support policies and practices, e.g., theater-based Joint Intelligence Center dissemination to lower-echelon users and direct dissemination from collectors to forces.

3.2.2.3 The Value of Integrating Mechanisms

One of the guiding principles (section 3.1.1) emphasized the importance of using a common set of architectural building blocks as the basis for architecture development. These common building blocks include common terms and definitions, common task listings, common activity sets, common operating environments, and others. Acceptance and use of such integrating mechanisms can promote architecture commonality and comparability, and can facilitate architecture development.

The Universal Joint Task List (UJTL) is an example of a task listing that could provide a common basis for deriving activities. Figure 3-3 illustrates the contribution that a universal task set can bring to the integration process. The value of such a mechanism in enabling integration increases as the scope of the architecture integration initiatives broadens. In the example shown, a common UJTL task, such as "Conduct Joint Force Targeting," provides a common-ground functional basis for comparing seemingly disparate architectures. Thus, the various architectural views described by different organizations can be more easily compared with respect to common activities and tasks.

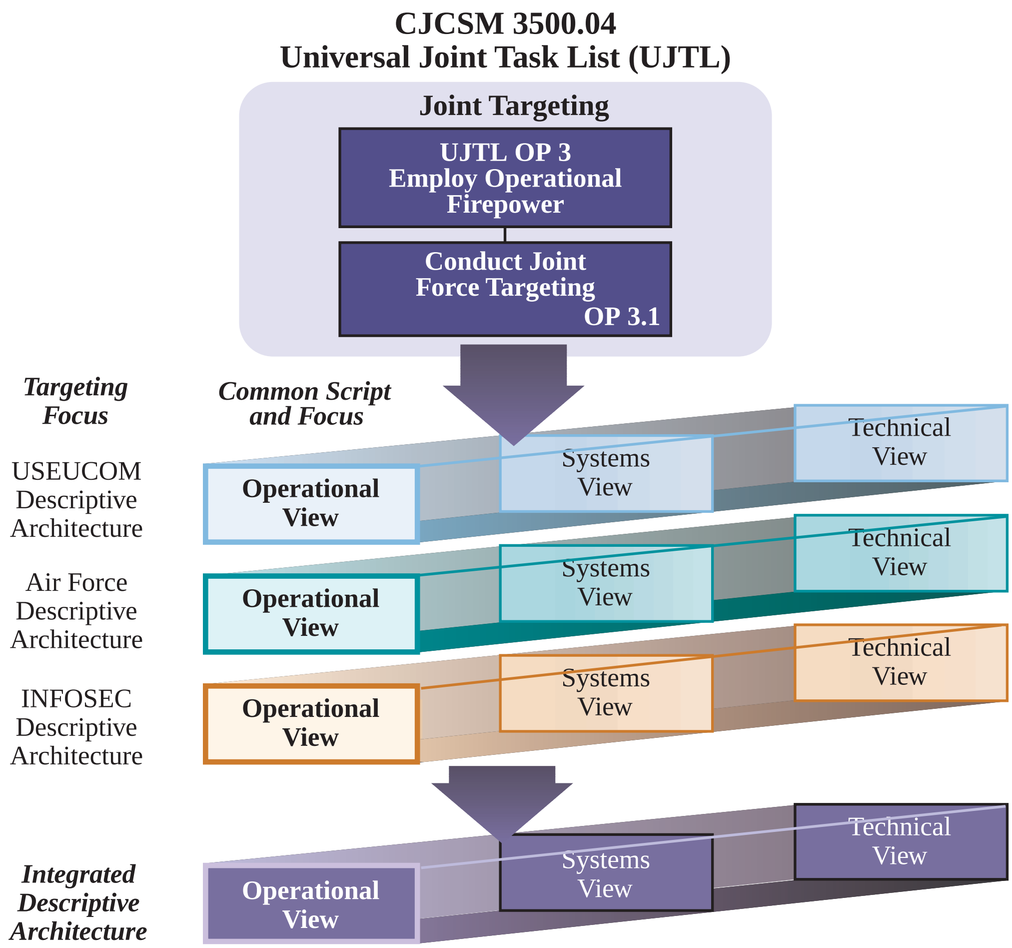


Figure 3-3. Illustration of the UJTL Serving as an Integrating Mechanism

3.3 FACILITATORS — ARCHITECTURE DATA AND TOOLS

3.3.1 Evolution of Architecture Data

Prior to adopting the Framework as guidance for creating future DoD architectures, each Military Service, major command, and Defense Agency used its own methodologies for developing and describing C4ISR architectures. Architecture databases were usually among the tools used to support these methodologies. Unfortunately, each database was developed around a different data model. That made it difficult for architects and system developers to exchange information and ensure joint interoperability. They first had to familiarize themselves with several different approaches for structuring similar information. They then had to translate and correlate the data from disparate sources before they could perform any meaningful comparison or analysis. Now, with the growing emphasis on Joint operations and interoperability of C4ISR systems, a common, DoD-wide approach is needed for organizing and portraying the structure of architecture information.

3.3.2 C4ISR Core Architecture Data Model (CADM)

The C4ISR Core Architecture Data Model (CADM), discussed in detail in section 4.3.1, is a formal model of architecture products, structures, and their relationships. The CADM is aimed at providing a common meta-model, or (logical) schema, for repositories of architecture information.

A repository based on the CADM would be able to store architecture products from multiple Framework-based architecture projects in a common way, so that products from different projects could be jointly analyzed and compared. The CADM is useful in guiding the evolution of Framework product types because the CADM can provide a check on the completeness and consistency of the information called for in the products. The CADM will also be useful to toolbuilders who will provide tools for building Framework-compliant products and repositories to store those products, and to toolsmiths who will be tailoring those tools and repositories for specific architecture projects. However, the CADM itself is not a Framework architecture product, and most users of the Framework (with the exception of toolbuilders and toolsmiths) will usually not deal directly with the CADM.

The CADM and the Architecture Framework's products are complementary, not alternatives. Thus, both the CADM and the Framework's products will remain important to DoD architecture processes. In essence, the CADM defines a common approach for organizing and sharing the information that is contained in the Framework products. The CADM offers flexible and automated queries while the Framework offers standardized views to facilitate comparison and integration. A database implementing the CADM can store information used to produce Framework products. It can also store the Framework products themselves.

Intentionally Blank

SECTION 4

C4ISR ARCHITECTURE PRODUCTS

Architecture products are those graphical, textual, and tabular items that are developed in the course of building a given architecture description and that describe characteristics pertinent to its purpose. When completed, this set of products constitutes the architecture description. These architecture products are differentiated from the world of pre-existing information sources that may be used in building architectures, such as existing architectural models, lexicons, pick-lists, and technical reference models. Applicable extracts from these sources may be used in the architecture description itself as portions of products, and the completed architecture becomes an information source for other efforts.

4.1 PRODUCT CATEGORIES

This document presents principles and techniques that can be used by organizations at all levels for building architectures. However, an important objective is to enable the construction of architectures that can be used in Joint and multi-national analysis. The two main types of analysis of concern here are: (1) analysis that supports the rapid synthesis of "go-to-war" capabilities suites; and (2) analysis that supports DoD investment decisions. These kinds of analyses require architectures to be comparable and integratable. For *every* architecture to have the potential for use in such analysis, it is necessary for *every* architecture to contain a common subset of the standard products.

The architecture products that will be developed by DoD organizations in support of their specific architecture scopes and purposes are classified into two categories, namely:

- **Essential Products**: These products constitute the minimal set of products required to develop architectures that can be commonly understood and integrated within and across DoD organizational boundaries and between DoD and multi-national elements. These products must be developed for all architectures.

- **Supporting Products**: These products provide data that will be needed depending on the purpose and objectives of a specific architecture effort. Appropriate products from the supporting product set will be developed depending on the purpose and objectives of the architecture.

The essential and supporting product types, built in accordance with the guidance and examples provided herein, will capture the characteristics needed for particular purposes, as well as satisfying Joint and multi-national analysis needs.

In the course of developing the essential and supporting products, one or more DoD references, e.g., the Joint Technical Architecture, may be required to ensure that specific architectures are complete and in conformance with current policy and formal direction. These references are addressed in section 4.3, in a special product category called *universal reference resources*.

The product set actually built for each architecture depends on the architecture's purpose and intended uses. In general, for broad scope and high-level analyses, the essential product set will suffice. As the purpose and scope narrow, and the uses involve more detailed analysis and/or modeling, supporting products are brought to bear as well. Figure 4-1 illustrates this concept.

The rows in figure 4-1 represent various purposes for which architectures are commonly built, ranging from broad, "community-wide" interests such as cross-DoD or cross-CINC strategies for leveraging common solutions, to focused initiatives, e.g., interoperability assessments or system design tradeoffs and analysis.

The columns in the figure notionally depict products that would be brought to bear. Note that all of the essential products are used in all cases. Supporting products are used selectively, depending on the value they contribute to the specific architecture purpose. The figure illustrates that, in general, those supporting products that add "breadth" of scope (e.g., decomposition of activities, command relationships, systems integration perspectives, etc.) may be selected to augment the essential product set to support the broader types of architecture purposes. On the other hand, those supporting product types that augment the essential product set by providing a more concentrated focus and treatment of minute details (e.g., detailed system components and functions) would likely be selected to support more concentrated architecture analysis purposes or detailed system design.

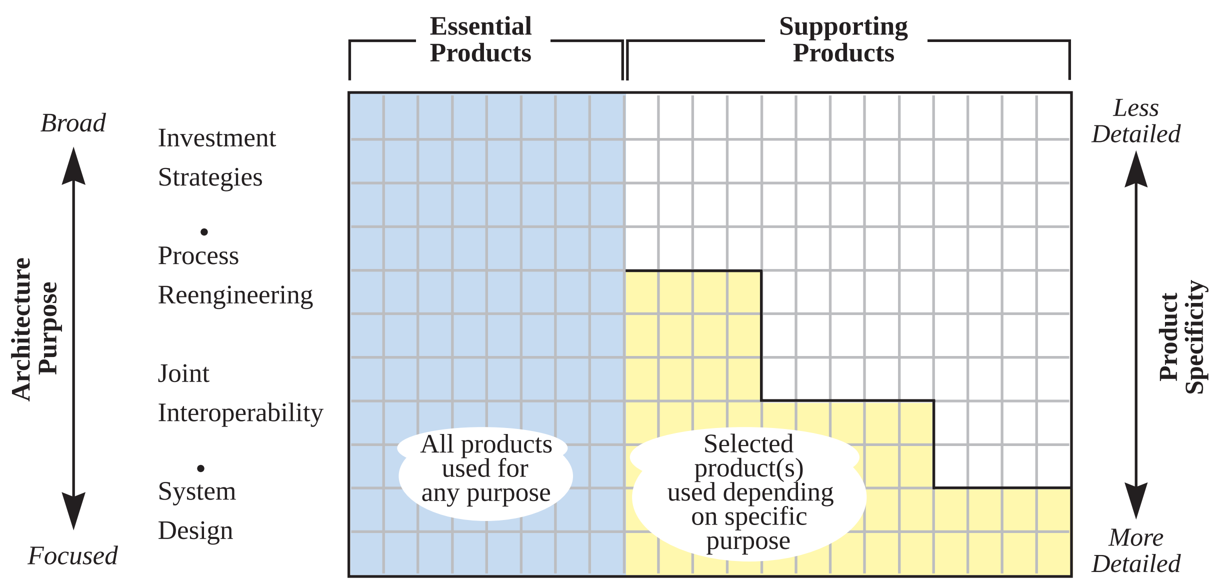


Figure 4-1. Conceptual Relationship Between Architecture Purposes and Products Used

The essential product set was selected so that, taken as a whole, it facilitates the ability to:

- Compare, analyze, and integrate operational, systems, and technical views of one architecture to another to determine overlaps and gaps

- Identify Joint interfaces and reveal potential new Joint interfaces

- Have at least one essential product for each of the architectural views (operational, systems, technical)

- Describe the relationships among an architecture's operational, systems, and technical views.

- Provide the essential information flows

The supporting product set, as notionally represented in figure 4-1, provides the architect with choices for extending the description to suit the specific purpose at hand.

4.2 ESSENTIAL AND SUPPORTING PRODUCTS

In the paragraphs that follow, the essential products (section 4.2.1) and the supporting products (section 4.2.2), both types identified in table 4-1, are described. For most of the products, a generic template is shown that illustrates the basic format of the product. In many cases, existing, real-world examples are provided.

Table 4-1 provides a summary of the essential and supporting products. The first column indicates the architecture view or views generally supported by each product. The second column provides an alphanumeric reference "identifier" for each product, where AV = all views, OV = operational view, SV = systems view, and TV = technical view. The third column provides the formal name of the product. The fourth column indicates whether the product is *essential* or *supporting*. *Essential* products are also highlighted by green shading. The fifth column captures the general nature of the product's content, followed by the number of the section where the product is described.

More details regarding the descriptive attributes associated with the essential and supporting products are provided in tables in appendix A.

Table 4-1. Essential and Supporting Framework Products

Applicable Architecture View	Product Reference	Architecture Product	Essential or Supporting	General Nature
All Views (Context)	AV-1	***Overview and Summary Information***	**Essential**	Scope, purpose, intended users, environment depicted,analytical findings, if applicable *(4.2.1.1)*
All Views (Terms)	AV-2	***Integrated Dictionary***	**Essential**	Definitions of all terms used in all products *(4.2.1.2)*
Operational	OV-1	***High-level Operational Concept Graphic***	**Essential**	High-level graphical description of operational concept (high-level organizations, missions, geographic configuration, connectivity, etc.) *(4.2.1.3)*
Operational	OV-2	***Operational Node Connectivity Description***	**Essential**	Operational nodes, activities performed at each node, connectivities & information flow between nodes *(4.2.1.4)*
Operational	OV-3	***Operational Information Exchange Matrix***	**Essential**	Information exchanged between nodes and the relevant attributes of that exchange such as media, quality, quantity, and the level of interoperability required. *(4.2.1.5)*
Operational	OV-4	***Command Relationships Chart***	Supporting	Command, control, coordination relationships among organizations *(4.2.2.1)*
Operational	OV-5	***Activity Model***	Supporting	Activities, relationships among activities, I/Os, constraints (e.g., policy, guidance), and mechanisms that perform those activities. In addition to showing mechanisms, overlays can show other pertinent information. *(4.2.2.2)*
Operational	OV-6a	***Operational Rules Model***	Supporting	One of the three products used to describe operational activity sequence and timing that identifies the business rules that constrain the operation *(4.2.2.3.1)*
Operational	OV-6b	***Operational State Transition Description***	Supporting	One of the three products used to describe operational activity sequence and timing that identifies responses of a business process to events *(4.2.2.3.2)*
Operational	OV-6c	***Operational Event/Trace Description***	Supporting	One of the three products used to describe operational activity sequence and timing that traces the actions in a scenario or critical sequence of events *(4.2.2.3.3)*
Operational	OV-7	***Logical Data Model***	Supporting	Documentation of the data requirements and structural business process rules of the Operational View. *(4.2.2.4)*
Systems	SV-1	***System Interface Description***	**Essential**	Identification of systems and system components and their interfaces, within and between nodes *(4.2.1.6)*
Systems	SV-2	***Systems Communications Description***	Supporting	Physical nodes and their related communications laydowns *(4.2.2.5)*
Systems	SV-3	***Systems² Matrix***	Supporting	Relationships among systems in a given architecture; can be designed to show relationships of interest, e.g., system-type interfaces, planned vs. existing interfaces, etc. *(4.2.2.6)*
Systems	SV-4	***Systems Functionality Description***	Supporting	Functions performed by systems and the information flow among system functions *(4.2.2.7)*
Systems	SV-5	***Operational Activity to System Function Traceability Matrix***	Supporting	Mapping of system functions back to operational activities *(4.2.2.8)*
Systems	SV-6	***System Information Exchange Matrix***	Supporting	Detailing of information exchanges among system elements, applications and H/W allocated to system elements *(4.2.2.9)*
Systems	SV-7	***System Performance Parameters Matrix***	Supporting	Performance characteristics of each system(s) hardware and software elements, for the appropriate timeframe(s) *(4.2.2.10)*
Systems	SV-8	***System Evolution Description***	Supporting	Planned incremental steps toward migrating a suite of systems to a more efficient suite, or toward evolving a current system to a future implementation *(4.2.2.11)*
Systems	SV-9	***System Technology Forecast***	Supporting	Emerging technologies and software/hardware products that are expected to be available in a given set of timeframes, and that will affect future development of the architecture *(4.2.2.12)*
Systems	SV-10a	***Systems Rules Model***	Supporting	One of three products used to describe systems activity sequence and timing -- Constraints that are imposed on systems functionality due to some aspect of systems design or implementation *(4.2.2.13.1)*
Systems	SV- 10b	***Systems State Transition Description***	Supporting	One of three products used to describe systems activity sequence and timing -- Responses of a system to events *(4.2.2.13.2)*
Systems	SV -10c	***Systems Event/Trace Description***	Supporting	One of three products used to describe systems activity sequence and timing -- System-specific refinements of critical sequences of events described in the operational view *(4.2.2.13.3)*
Systems	SV-11	***Physical Data Model***	Supporting	Physical implementation of the information of the Logical Data Model, e.g., message formats, file structures, physical schema *(4.2.2.14)*
Technical	TV-1	***Technical Architecture Profile***	**Essential**	Extraction of standards that apply to the given architecture *(4.2.1.7)*
Technical	TV-2	***Standards Technology Forecast***	Supporting	Description of emerging standards that are expected to apply to the given architecture, within an appropriate set of timeframes *(4.2.2.15)*

4.2.1 Essential Framework Products

As stated earlier, the essential products are the minimal set required to develop architectures that can be commonly understood and integrated within and across DoD organizational boundaries and between DoD and multi-national elements. These products must be developed for all architecture descriptions that contain the applicable views; i.e., all architecture descriptions that contain an operational view must include the "OV (Operational View)" essential products, all architecture descriptions that contain a systems view must include the "SV (Systems View)" essential products, and all architecture descriptions that contain a technical view must include the "TV (Technical View)" essential product.

4.2.1.1 Overview and Summary Information (AV-1)

All Views	Essential Product

The Overview and Summary Information product serves two purposes. In the initial phases of architecture development it serves as a planning guide. Upon completion of an architecture project this product provides summary textual information concerning "who, what, when, why, and how."

Overview and summary information should be provided in a consistent form that will allow quick reference and comparison among architectures. The following directions apply when providing the Overview and Summary Information:

- ***Identification.*** Provide a unique descriptive name for the architecture, identify the architect (i.e., name and organization), identify involved organizations, and indicate when the architecture was developed.

- ***Purpose***. Explain why the architecture is needed, what it is intended to demonstrate, the types of analysis expected to be applied to it, who is expected to perform the analysis, what decisions are expected to be made on the basis of that analysis, who is expected to make those decisions, and what actions are expected to result from the architecture.

- ***Scope.*** Identify the architecture views and products that have been developed (operational, systems, and/or technical) and the temporal nature of the architecture, such as the time frame covered, whether by specific years or by designations such as "as-is," "to-be," "transitional," "objective," et cetera.

- ***Context.*** Describe the interrelated conditions that compose the setting in which the architecture exists. Include such things as doctrine, relevant goals and vision statements, concepts of operation, scenarios, and environmental conditions. Identify the tasking that led to the architecture's development, and known or anticipated linkages to other architectures. Document specific assumptions and constraints regarding the architecture development effort, and identify authoritative sources for the rules, criteria, and conventions that were followed in developing the architecture.

- ***Findings.*** State the findings and recommendations that have been developed based on the architecture. Examples of findings include identification of shortfalls, recommended systems implementations, and opportunities for technology insertion.

- ***Tools and file formats.*** Identify the tool suite used to develop the architecture data and products. Identify the file names, file format, and location of the data for each product.

Figure 4-2 shows a representative format for the Overview and Summary Information product. Blank lines on the format indicate likely areas for added user-defined information to be inserted.

Overview and Summary Information

- Identification
  - Name
  - Architect
  - Organizations Involved
  - When Developed
- Purpose
  - Analysis Needs
  - Decision Support Needs
  -
  -
- Scope
  - Views and Products Used
  - Time Frames Addressed
  -
  -
- Context
  - Mission
  - Geographical
  - Rules, Criteria, and Conventions Followed
  -
  -
- Findings
  - Results
  - Recommendations
- Tools and File Formats

Figure 4-2. Overview and Summary Information (AV-1) — Representative Format

4.2.1.2 Integrated Dictionary (AV-2)

All Views	Essential Product

Many of the architectural products have a graphical representation. However, there is textual information in the form of definitions and metadata (i.e., data about an item) associated with these graphical representations. The Integrated Dictionary provides a central source for all these definitions and metadata, including those that may be provided for convenience within another product as well. At a minimum, the Integrated Dictionary is a glossary with definitions of terms used in the given architecture description. The Integrated Dictionary makes the set of architecture products stand alone and allows it to be read and understood without reference to other documents.

Each labeled graphical item (e.g., icon, box, or connecting line) in the graphical representation of an architectural product should have a corresponding entry in the Integrated Dictionary. The type of metadata included in the Integrated Dictionary for each type of item will depend on the type of architectural product from which the item is taken. For example, the metadata about a labeled input/output connector from an activity model (e.g., an IDEF0 ICOM) will include a textual description of the type of input/output information designated by the label.

The contents for the Integrated Dictionary entries for each product type are evolving; current lists can be found in the "attribute" tables provided for each product in appendix A. These attributes are consistent with the CADM meta-model for the architecture products. The Integrated Dictionary product contains the instance values of the data for specific architecture projects, while the CADM describes the types and relationships of these values. Everything in the Integrated Dictionary could be stored in a CADM-based repository, just as all Framework architecture products could be stored in a CADM-based repository.

Architects should use standard terms where possible (i.e., terms from existing, approved dictionaries and lexicons). However, in some cases, new terms and/or modified definitions of existing terms will be needed. This can happen when a given architecture is at a lower level of detail than existing architectures or lexicons, or when new concepts are devised for objective architectures. In those cases, the *new terms contained in a given architecture's Integrated Dictionary should be submitted to the maintainer of the approved dictionaries*. All definitions that originate in existing dictionaries should provide a reference to show the source.

4.2.1.3 High-Level Operational Concept Graphic (OV-1)

Operational View	Essential Product

The High-level Operational Concept Graphic is the most general of the architecture-description products and the most flexible in format. Its main utility is as a facilitator of human communication, and it is intended for presentation to high-level decision makers. This kind of diagram can also be used as a means of orienting and focusing detailed discussions.

The High-level Operational Concept Graphic template is shown in figure 4-3.

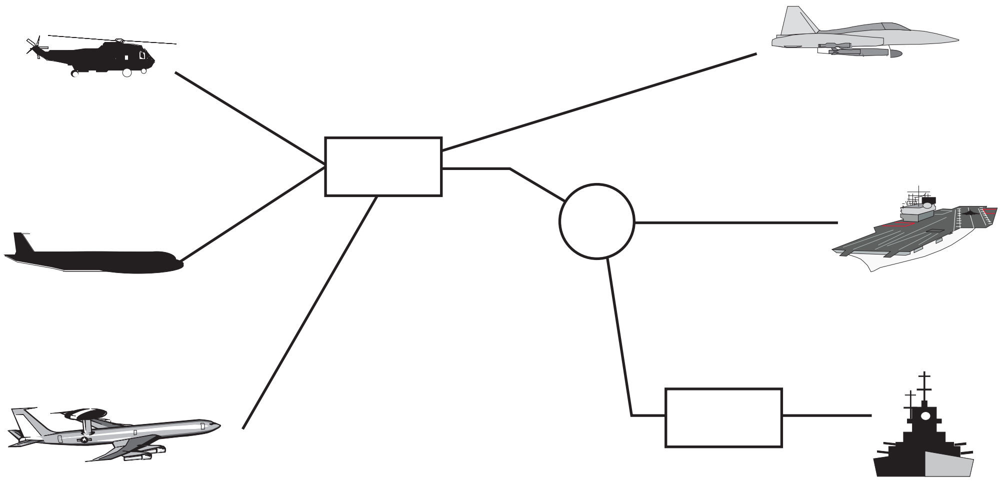


Figure 4-3. High-level Operational Concept Graphic (OV-1) — Template

The template shows generic icons that can be tailored as needed and used to represent various classes of players in the architecture (e.g., an aircraft icon can represent a particular type of aircraft, or a particular air organization, or the air assets of a Joint Task Force). The icons could also be used to represent missions or tasks (e.g., the aircraft icon could represent Air Operations, the ship icon could represent Maritime Operations, etc.). The lines connecting the icons can be used to show simple connectivity, or can be annotated to show what information is exchanged. How the template is tailored depends on the scope and intent of the architecture, but in general a High-level Operational Concept Graphic will show such things as the missions, high-level operations, organizations, and geographical distribution of assets.

Figures 4-4a and 4-4b show examples of the High-level Operational Concept Graphic.

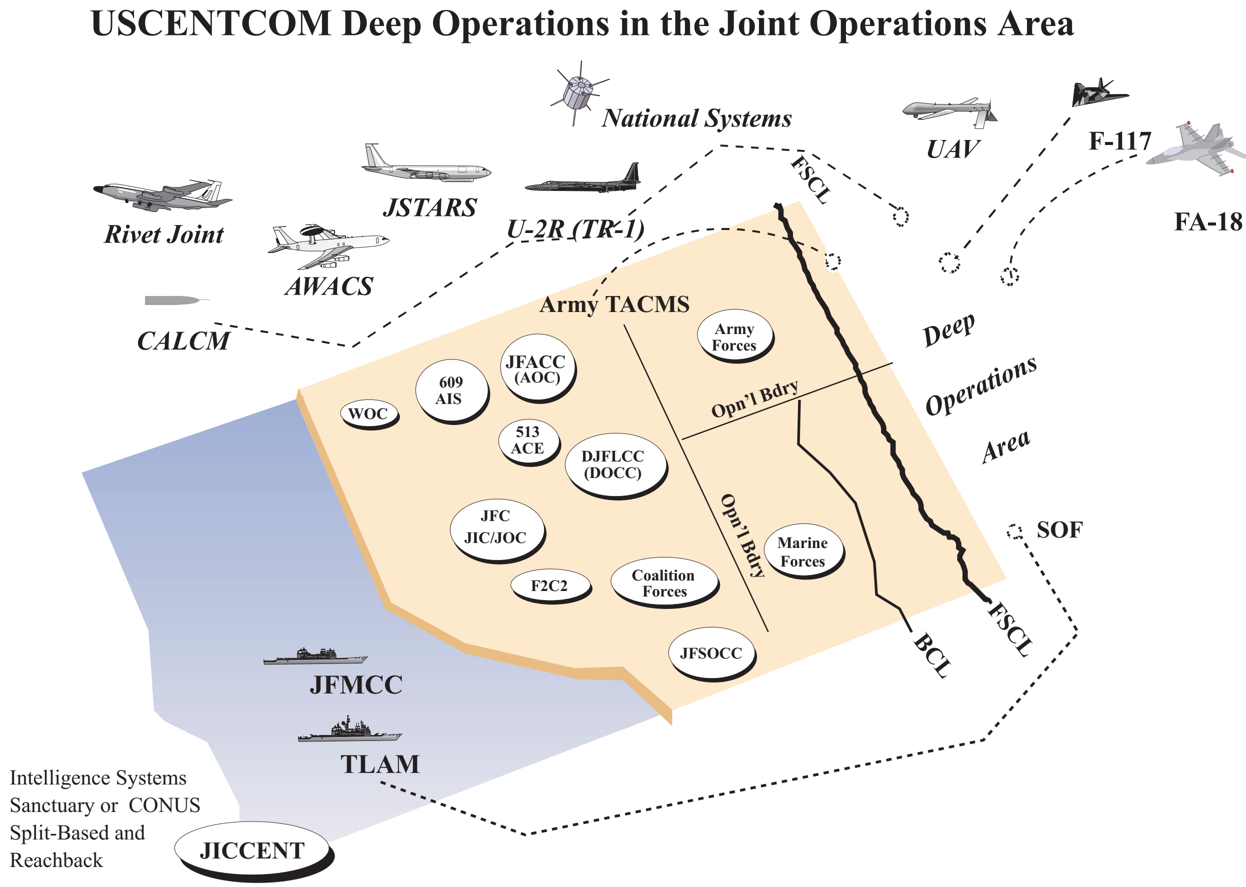


Figure 4-4a. High-level Operational Concept Graphic (OV-1) — USCENTCOM Example

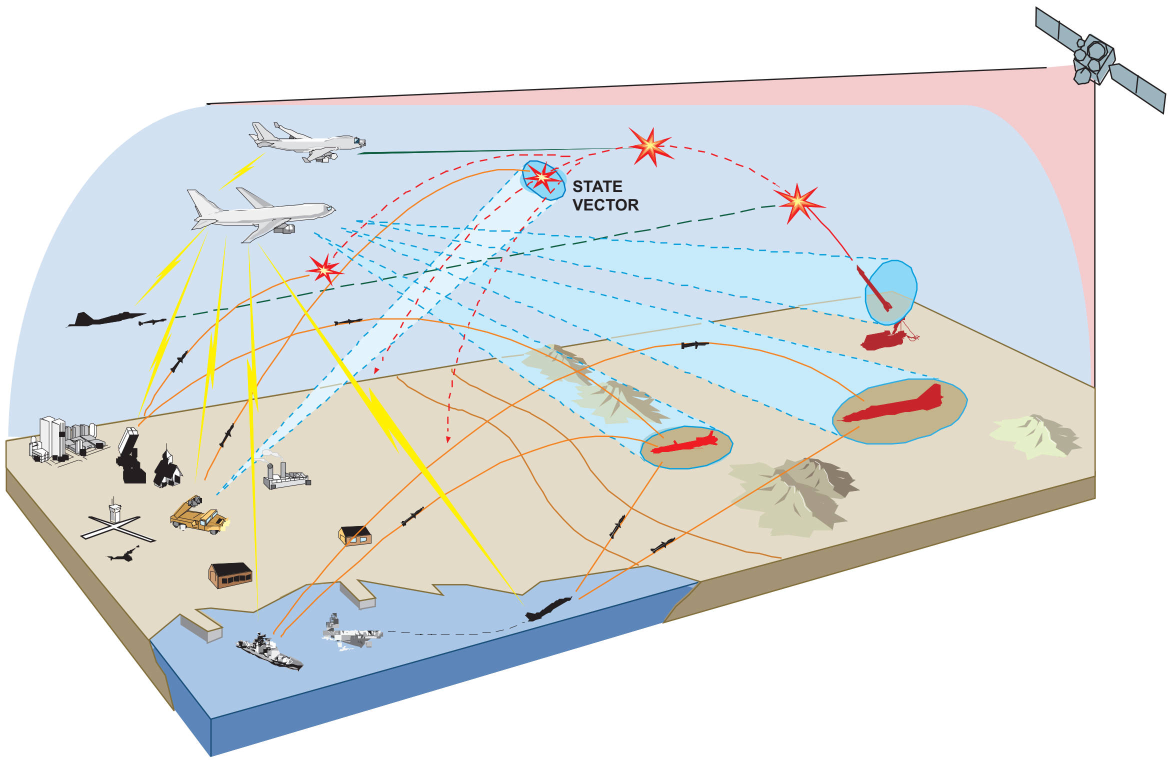


Figure 4-4b. High-level Operational Concept Graphic (OV-1) —
Theater Air Defense Example

4.2.1.4 Operational Node Connectivity Description (OV-2)

Operational View | **Essential Product**

The main features of this product are the operational nodes and elements, the needlines between them, and the characteristics of the information exchanged. Each information exchange is represented by an arrow (indicating the direction of information flow), which is annotated to describe the characteristics of the data or information (e.g., its substantive content, media [voice, imagery, text and message format, etc.]), volume requirements, security or classification level, timeliness, and requirements for information system interoperability (see the universal reference resources in section 4.3). Information-exchange characteristics can be shown selectively on the diagram, or more comprehensively in a matrix format (see section 4.2.1.5).

The information illustrated in the Operational Node Connectivity Description can be used to make decisions about which systems are needed to satisfy the business needs of an organization or functional area. However, it is the conduct of business/operations that is illustrated, not supporting systems.

Operational architecture views are not required to name real physical facilities as nodes. Operational architecture views can instead focus on “virtual” nodes, which could be based on operational “roles.” Thus, operational “nodes” would not always be directly integratable with real (physical) nodes from other architectures, but they could provide insight as to which real nodes might be able to assume the roles portrayed.

As mentioned earlier, what constitutes an operational node can vary from one organization to another, including, but not limited to, representing a role (e.g., Air Operations Commander), an organization (e.g., U.S. Air Force), an operational facility (e.g., Joint Intelligence Center), and so on. The notion of “node” will likewise vary depending on the level of detail addressed by the architecture effort.

In many instances in the past, organizations have represented some operational nodes in physical (and locational) terms if these nodes were intended to remain “constant” in the architecture analysis (e.g., determine the most cost-effective communications options between an in-garrison CINC and a JTF commander located at x, y, or z). On the other hand, organizations have tended to represent operational nodes much more generically, or notionally, if the entire “business” practice was being analyzed from scratch, with no constraints (e.g., current facilities) confronting the architect.

To emphasize the focus of the analysis and to ensure comparability and integratability across efforts, it is important therefore that each organization carefully document its use of the “operational node” concept.

The activities associated with a given information exchange should be noted in some way to provide linkages between each node and the activities performed there; this is especially true if no formal activity model is developed. (An Operational Node Connectivity Description, in effect, “turns the activity model inside out,” focusing first-order on the nodes, and second-order on the activities. An activity model, on the other hand, places first-order attention on activities, and second-order attention on nodes, which can be shown as mechanisms.) Activities may be associated with the node.

Figure 4-5 provides a template for the Operational Node Connectivity Description.

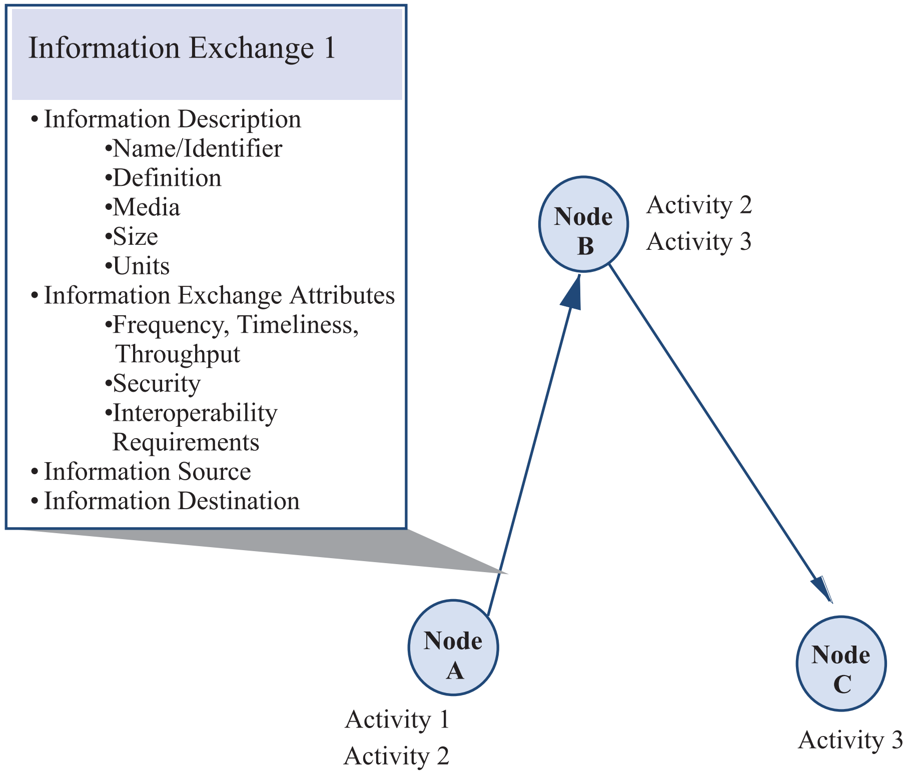


Figure 4-5. Operational Node Connectivity Description (OV-2) —Template

Figure 4-6 provides a notional example of an Operational Node Connectivity Description, and figures 4-6a through 4-6d provide specific examples.

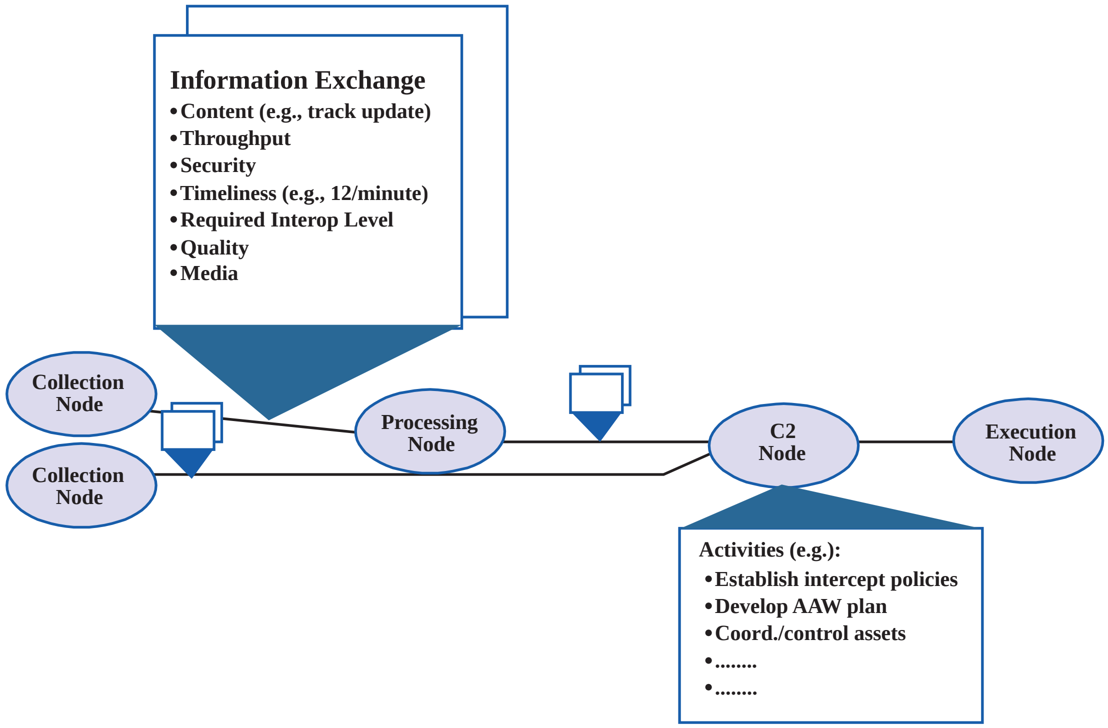


Figure 4-6. Operational Node Connectivity Description (OV-2) — Notional Example

NSFS Direct Support to Army Forces

Situation 3

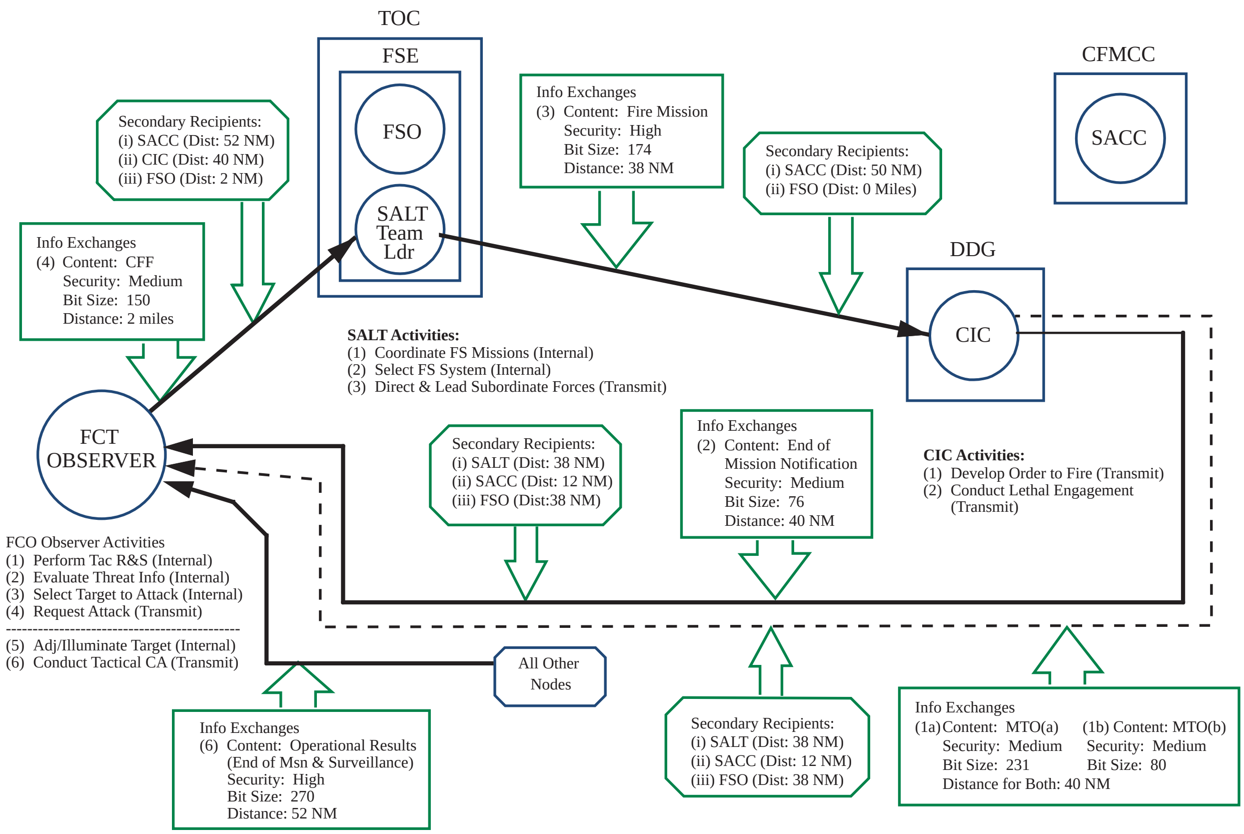


Figure 4-6a. Operational Node Connectivity Description (OV-2) — Naval Surface Fire Support to Army Forces Example

Figure 4-6b is taken from CISA's C4ISR Mission Assessment Final Report. Using the netViz automated tool, this diagram illustrates the operational node connectivities involved in the Close Support mission area for the 2006 timeframe. The attributes of interest were stored in a database and are available for display as needed. The diagram illustrates the attributes for one node and for one needline. An "activity background" is used to give a flavor of the operational activities performed by each node; i.e., the operational elements have been aligned to the high-level operational task(s) they perform.

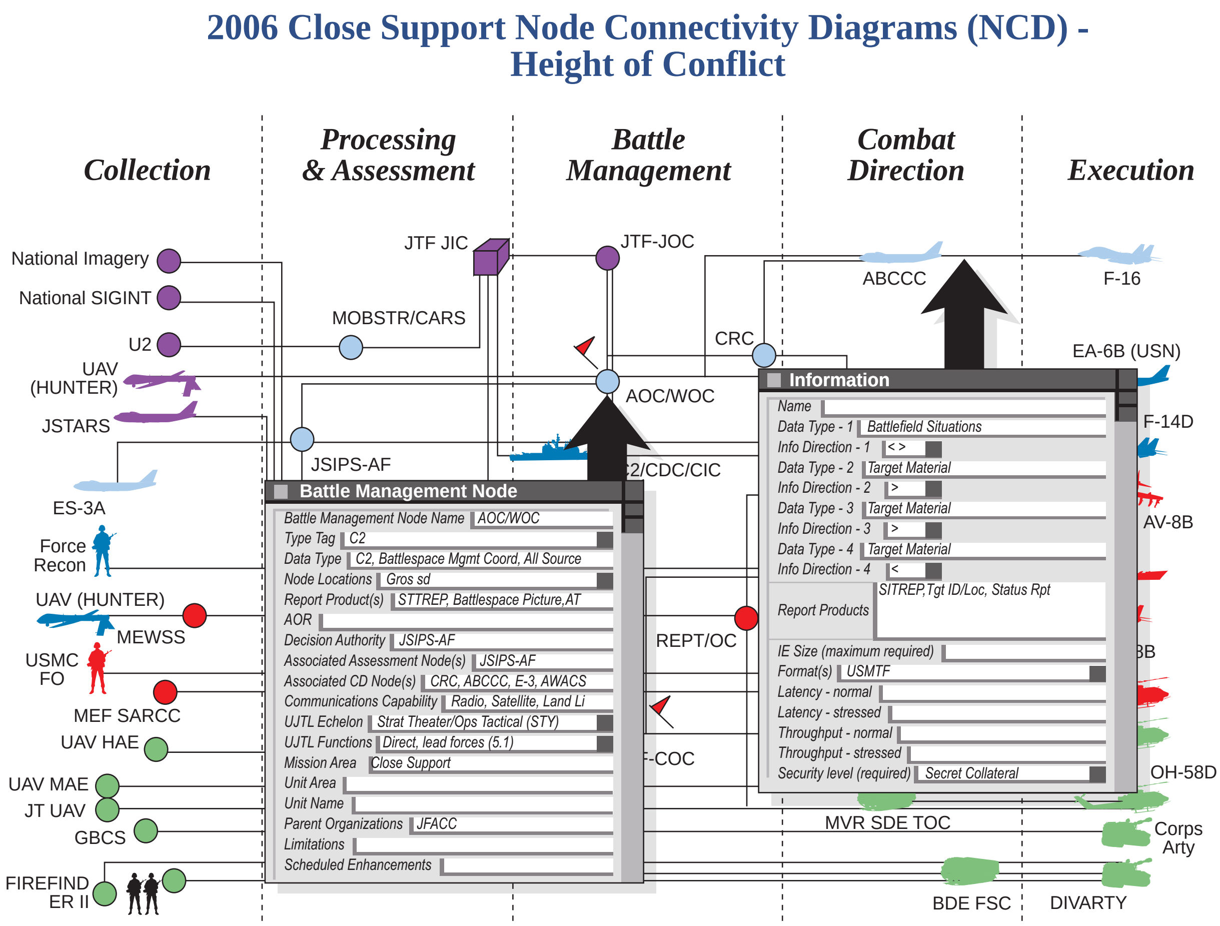


Figure 4-6b. Operational Node Connectivity Description (OV-2) — Close Support Joint Mission Area Example #1

Figure 4-6c is also taken from the C4ISR Mission Assessment Final Report, and is a companion to Figure 4-6b. Figure 4-6c specifies the policy-directed communications media (not specific communications systems or networks, as would be shown in a more detailed Systems Communications Description, described in section 4.2.2.5) associated with each of the generic connectivity needlines shown on the earlier figure. In the earlier figure, generic connectivities are shown as solid black lines, while in this figure those lines are shown in particular colors/line styles to indicate which communications medium is actually associated with the needline, e.g., a (red) dotted line indicates a radio link.

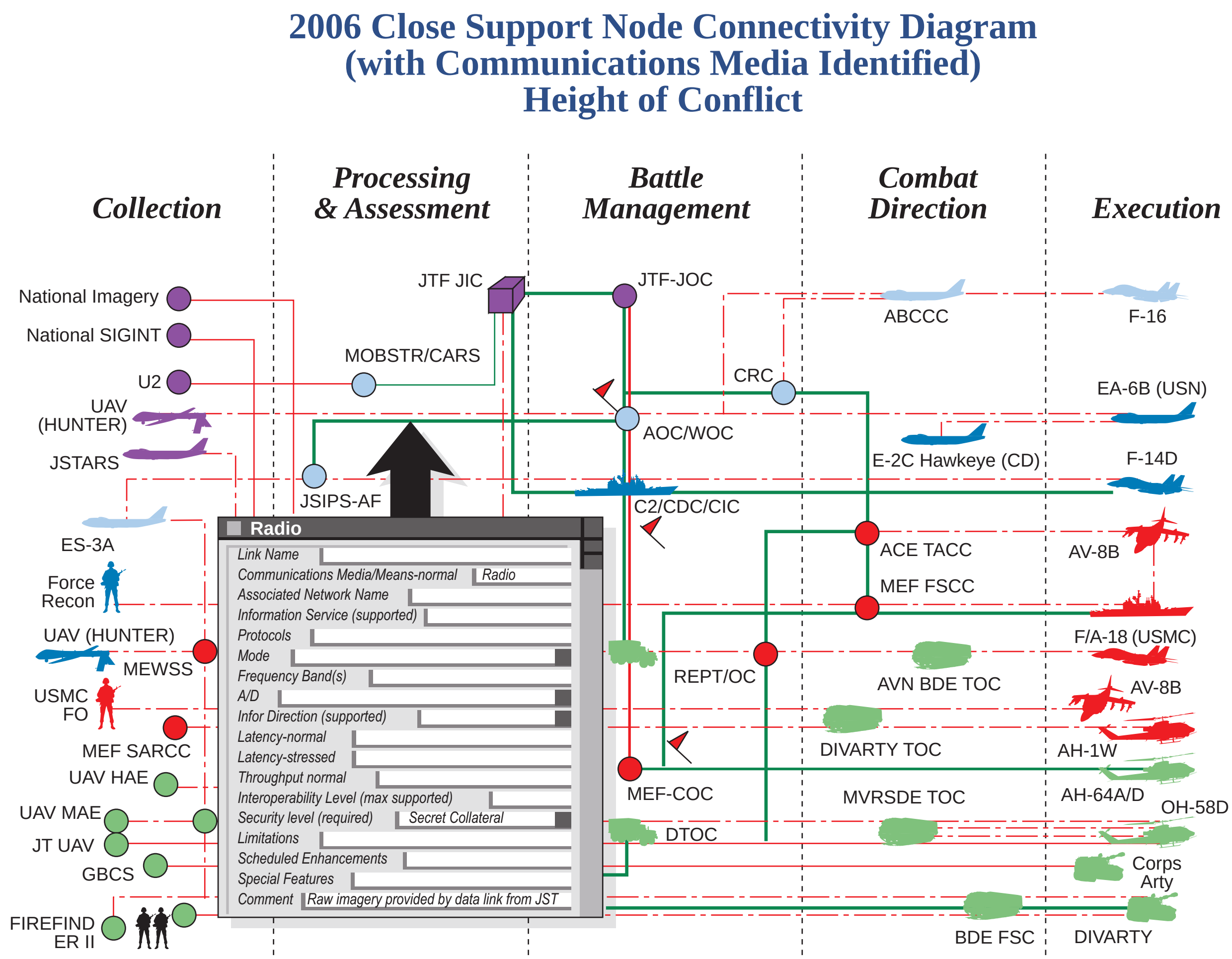


Figure 4-6c. Operational Node Connectivity Description (OV-2) — Close Support Joint Mission Area Example #2

Figure 4-6d provides yet another example of an Operational Node Connectivity Description. In this "target selection" example, a hierarchical, echelon presentation technique is used.

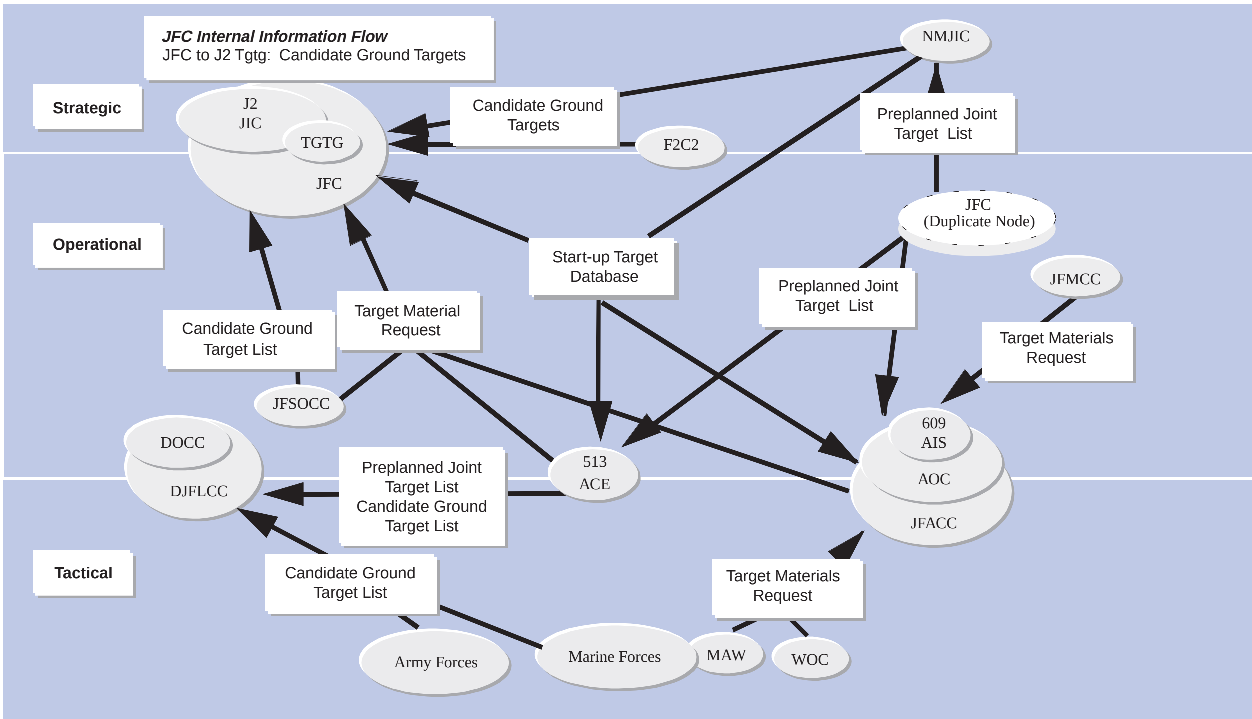


Figure 4-6d. Operational Node Connectivity Description (OV-2) —
Target Selection Example

Figures 4-6e and f provide additional examples.

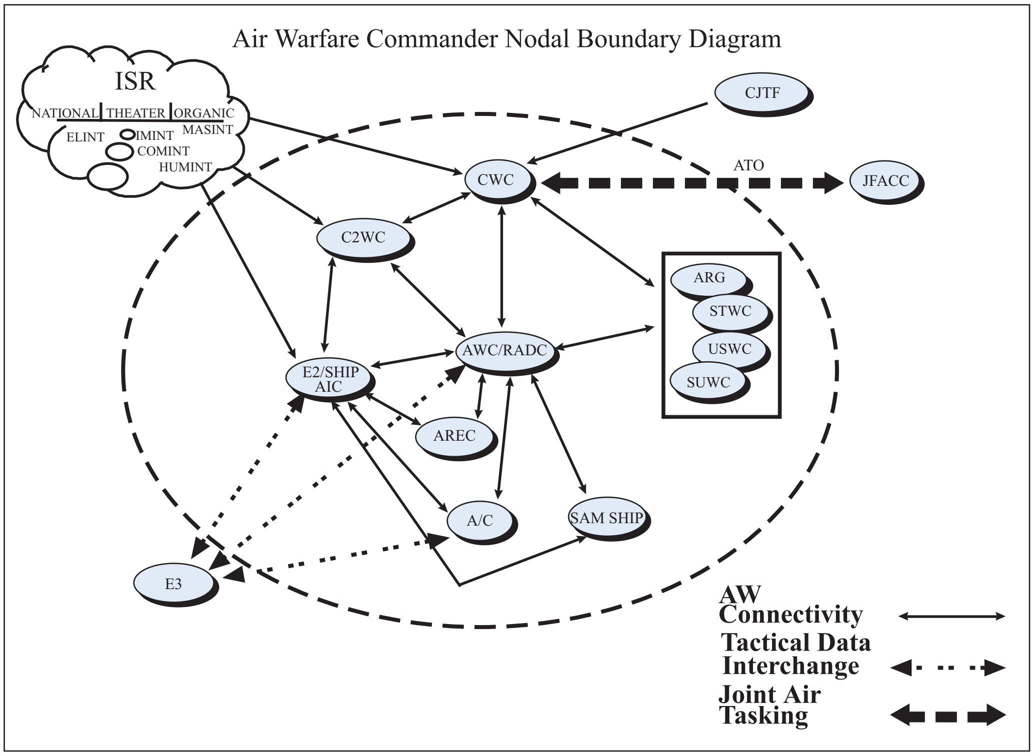
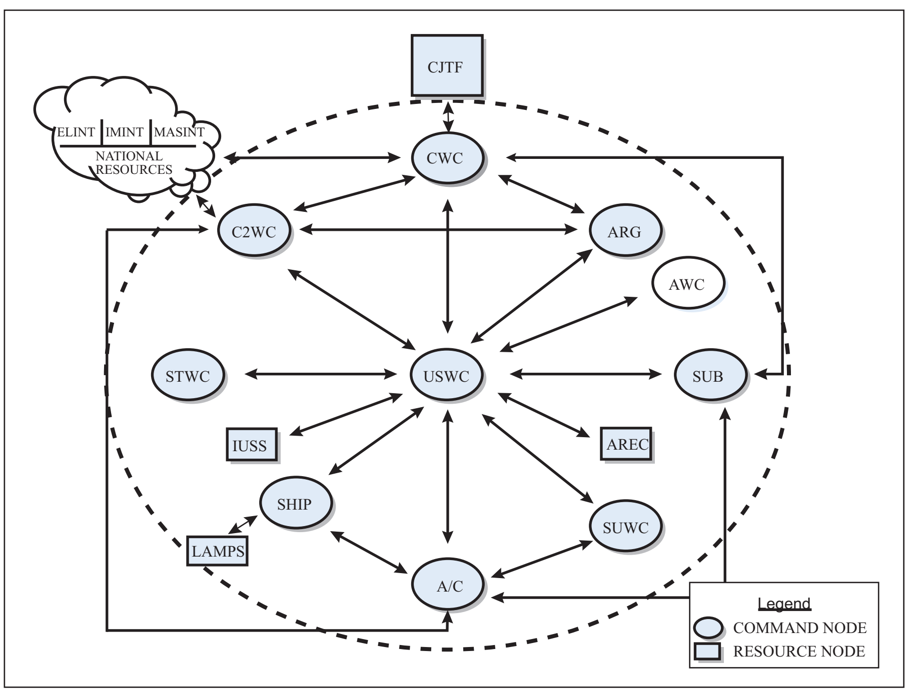


Figure 4-6e. Operational Node Connectivity Description (OV-2) —
Air Warfare Commander Example

Figure 4-6f. Operational Node Connectivity Description (OV-2) —



Example Showing Multiple Node Types

4.2.1.5 Operational Information Exchange Matrix (OV-3)

Operational View	Essential Product

Using the defined activities as a basis, Information Exchange Requirements (IERs) express the relationship across the three basic entities of an operational architecture (activities, operational elements, and information flow) with a focus on the specific aspects of the information flow. IERs identify *who* exchanges *what* information with *whom, why* the information is necessary, and in *what manner*. IERs identify the elements of warfighter information used in support of a particular activity and between any two activities. The node of the producing operational element and the node of the consuming operational element are identified. Relevant attributes of the exchange are noted. The specific attributes included are dependent on the objectives of the specific architecture effort, but may include the information media (e.g., data, voice, and video), quality (e.g., frequency, timeliness, and security), and quantity (e.g., volume and speed). Particular capabilities such as security level of communications may also be captured for each exchange. The emphasis in this product is on the logical and operational characteristics of the information (e.g., what information is needed by whom, from whom, and when).

The nature of the Operational IER Description lends itself to being described as a matrix, as in figure 4-7. However, the number of information exchanges associated with an architecture may be quite large. Also, in order to understand the nature of the information exchanges, the developers and users of the architecture may want to see the IER data sorted in multiple ways, such as by task, by node, or by attribute. Consequently, using a matrix to present that information is limiting and frequently not practical. Due to its highly structured format, the Operational Information Exchange Requirements Description lends itself readily to a spreadsheet or relational data base. In practice, hardcopy versions of this product should be limited to high-level summaries or highlighted subsets of particular interest.

A representative format for the Operational Information Exchange Matrix is illustrated in figure 4-7. Example extensions and refinements of the basic representative format are shown in figures 4-7a and 4-7b. Figure 4-7b illustrates a Navy-specific version of the Operational IER Matrix that contains information from the Hierarchical Data Dictionary and other Navy-specific reference resources. This example also shows the addition of administrative or configuration management information that might be added by tools. These two examples show how the basic information shown in figure 4-7 can be used as a starting point for project- or Service-specific tailoring and extension. The examples show additional or refined information columns in red (bold).

INFORMATION DESCRIPTION					INFORMATION SOURCE		INFORMATION DESTINATION		INFORMATION EXCHANGE ATTRIBUTES		
OPERATIONAL INFORMATION ELEMENT	DESCRIPTION	MEDIA	SIZE	UNITS	OPERATIONAL ELEMENT & ACTIVITY		OPERATIONAL ELEMENT & ACTIVITY		FREQUENCY, TIMELINESS, THROUGHPUT	SECURITY	INTEROPERABILITY REQUIREMENTS
NAME/ IDENTIFIER	DEFINITION	DIGITAL, VOICE, TEXT, IMAGE ETC.	RANGE LIMITS	FEET, LITERS, INCHES, ETC.	IDENTIFIER OF PRODUCING OE	PRODUCING ACTIVITY	IDENTIFIER OF CONSUMING OE	CONSUMING ACTIVITY			

Figure 4-7. Operational Information Exchange Matrix (OV-3) -- Representative Format

Identification & Connectivity			Operational Functionality			Information Description					Information Exchange Service Requirements			
Operational IER Identifier	Producing Operational Element	Consuming Operational Element	**Operational Task Supported**	Producing Operational Element Activity	Consuming Operational Element Activity	Warfighter Information	**Data Elements**	Information Security	Size/ Units	Media Type	Quantity	Quality	Timeliness	Interoperability
Unique Identifier			**UJTL or Service Mission Essential Task List**			Logical Grouping of Data Elements That Need to be Conveyed	**Data Element #1**	Could be by Information Element or Data Elements	Could be by Information Element or Data Elements	Voice, Video, Data	Frequency of exchanges over period of time	Acknowledge Authentication Message Error Rate	Precedence Perishability	Level of Interoperability And/or Identification of a TA Profile of Standards
							Data Element #2							
							Data Element #3							

Figure 4-7a. Operational Information Exchange Matrix (OV-3) -- Representative Format

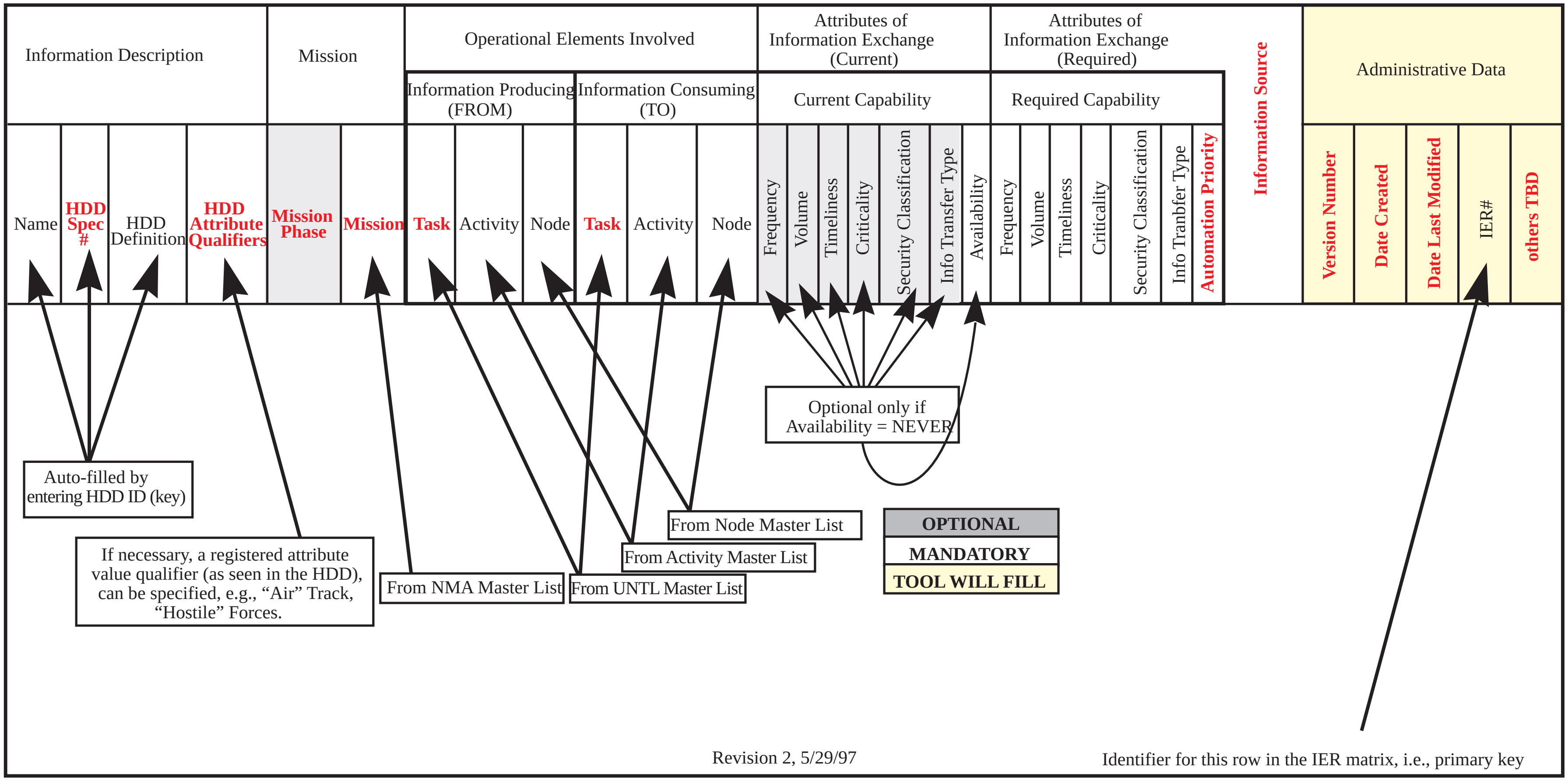


Figure 4-7b. Operational Information Exchange Matrix (OV-3) -- Example Related to Hierarchal Data Directory

4.2.1.6 System Interface Description (SV-1)

Systems View	Essential Product

The System Interface Description links together the operational and systems architecture views by depicting the assignments of systems and their interfaces to the nodes and needlines described in the Operational Node Connectivity Description. The Operational Node Connectivity Description for a given architecture shows operational nodes (not always defined in physical terms), while the System Interface Description depicts the corresponding systems nodes. Systems nodes include the allocations of specific resources (people, platforms, facilities, systems, …) that are being addressed for implementing specific operations.

The System Interface Description identifies the interfaces between systems nodes, between systems, and between the components of a system, depending on the needs of a particular architecture. A system interface is a simplified or generalized representation of a communications pathway or network, usually depicted graphically as a straight line (with amplifying information, e.g., “DISN”). Often, pairs of connected systems or system components have multiple interfaces between them. The System Interface Description depicts all interfaces between systems and/or system components that are of interest to the architect. Note that the detailed descriptions of each system interface, if required, are provided in the Systems Communications Description, a supporting product defined in section 4.2.2.5.

The graphic descriptions and/or supporting text for the System Interface Description should also provide details concerning the capabilities present in each system. For example, descriptions of information systems should include details concerning the procedures governing system implementation, the applications present within the system, the infrastructure capabilities and services that support the applications, and the means by which the system processes, manipulates, stores, and exchanges data.

The System Interface Description can be shown in three perspectives: internodal, intranodal, and intrasystem (system component). The following paragraphs describe these perspectives.

The *internodal perspective* of the System Interface Description identifies the systems nodes and the systems interfaces between the nodes, and may represent the systems at the nodes as well. The interfaces can be shown simply from node edge-to-node edge, or extended to show the interfaces between specific systems at each node and specific systems at other nodes. When specific systems are identified, the graphical description and/or supporting text should explicitly relate each system to the operational activities and the information-exchange needlines shown in the Operational Node Connectivity Description that the system supports.

Figure 4-8a provides a template of the *internodal perspective*, showing system interfaces between nodes from node edge-to-node edge. The pertinent systems within each node are also shown, but not with respect to their specific system-to-system interfaces.

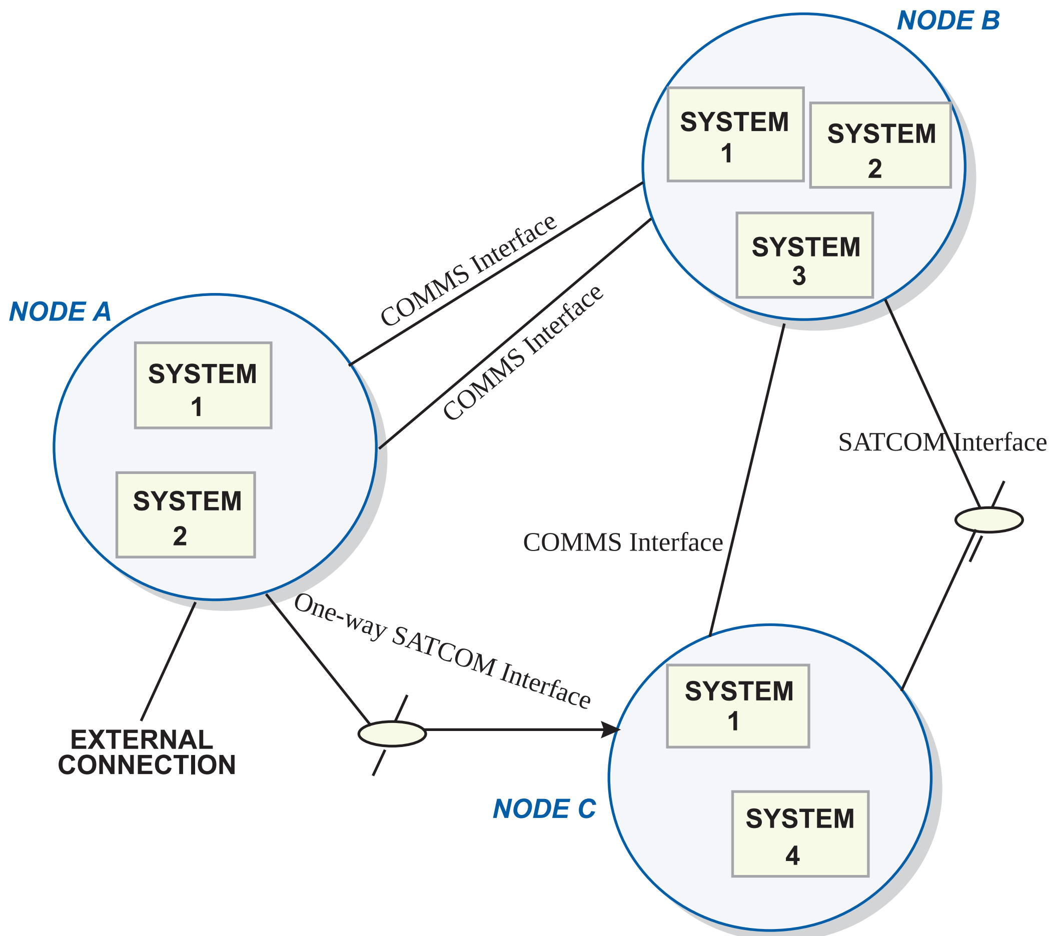


Figure 4-8a. System Interface Description, Internodal Perspective (SV-1) — Template Showing Node Edge-to-Node Edge Interfaces

Figure 4-8b provides a template of the *internodal perspective* of the System Interface Description that extends the node edge connections to specific systems.

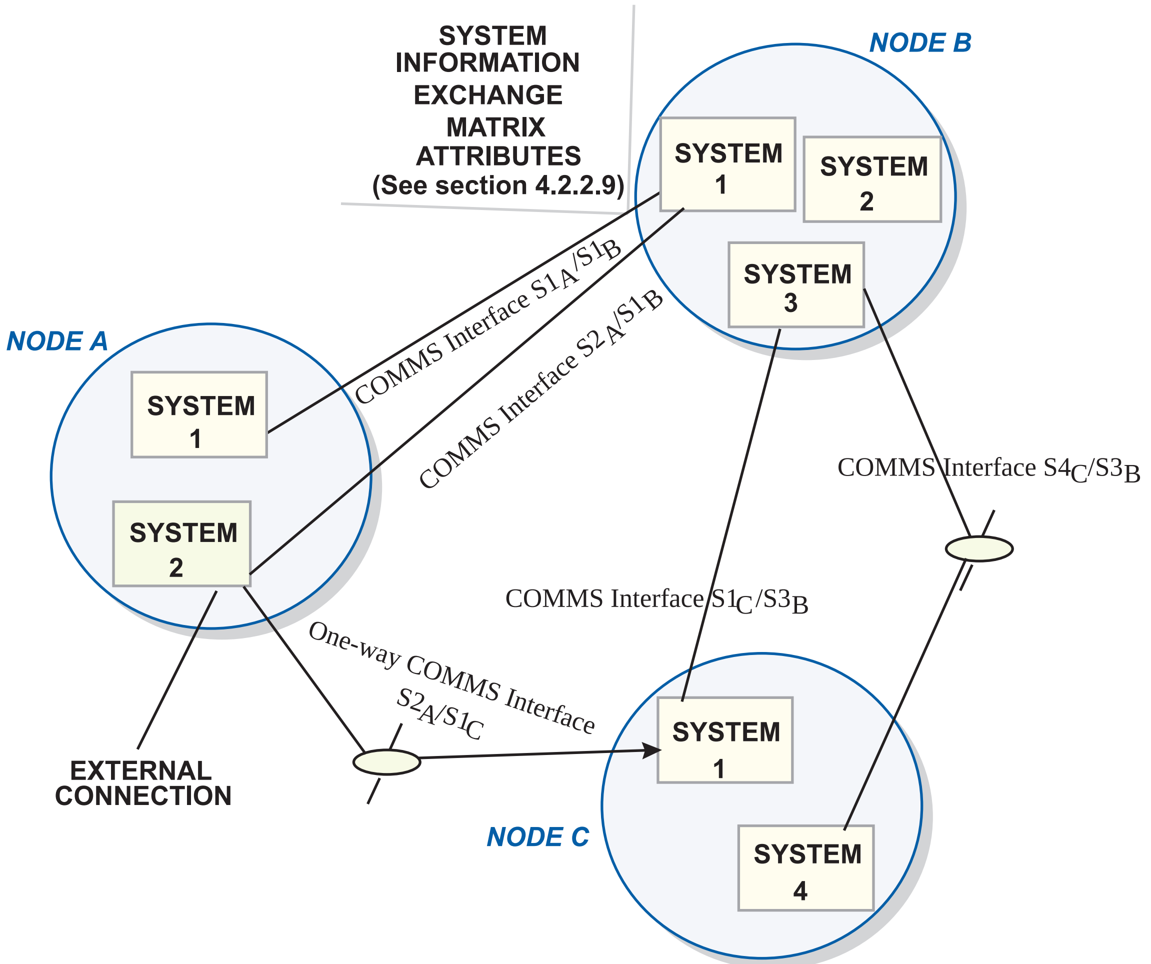


Figure 4-8b. System Interface Description, Internodal Perspective (SV-1) — Template Showing System-to-System Interfaces

Figures 4-8c and 4-8d provide a notional example and an actual example, respectively, of the *internodal perspective* of the System Interface Description.

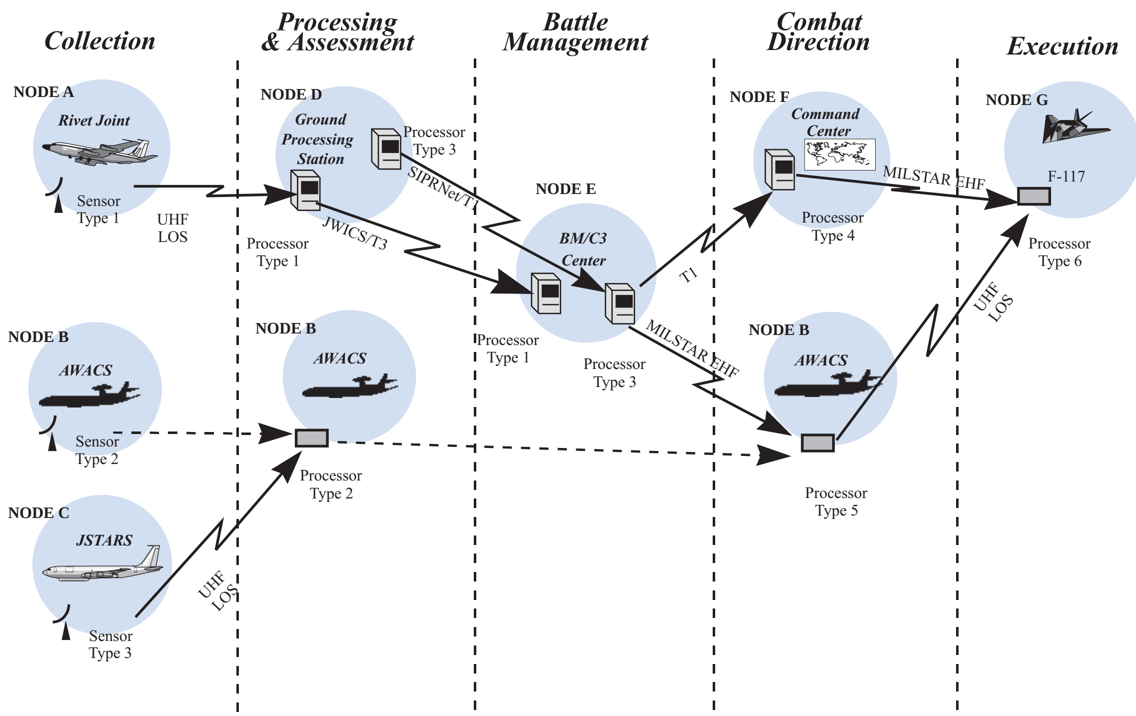


Figure 4-8c. System Interface Description, Internodal Perspective (SV-1) — Notional Example

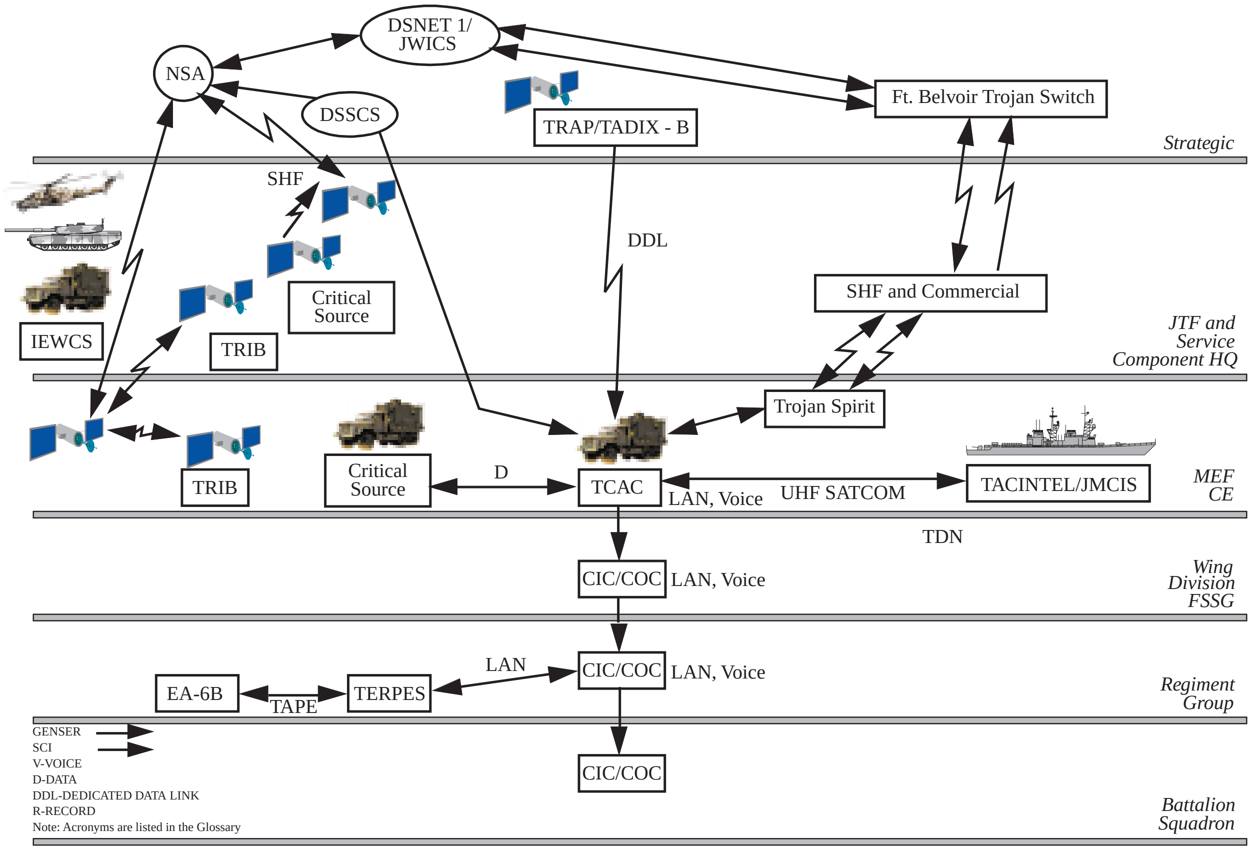


Figure 4-8d. System Interface Description, Internodal Perspective (SV-1) — USACOM CIAD Example with Nodes Depicted By Echelon

The *intranodal perspective* of the System Interface Description identifies the system-to-system interfaces within a node. Examples of interface elements include servers, security guards, any LAN and associated communications mechanisms (e.g., routers, gateways) that might provide a connectivity bus within the node, and communications mechanisms that provide node-external interfaces to or from each system. (In addition to identifying system-to-system interfaces, architecture developers are encouraged to associate the systems within a node to the activities identified in the Operational Node Connectivity Description for that node.)

Figure 4-9a provides a template of the intranodal perspective of the System Interface Description. Figures 4-9b through 4-9c present actual examples.

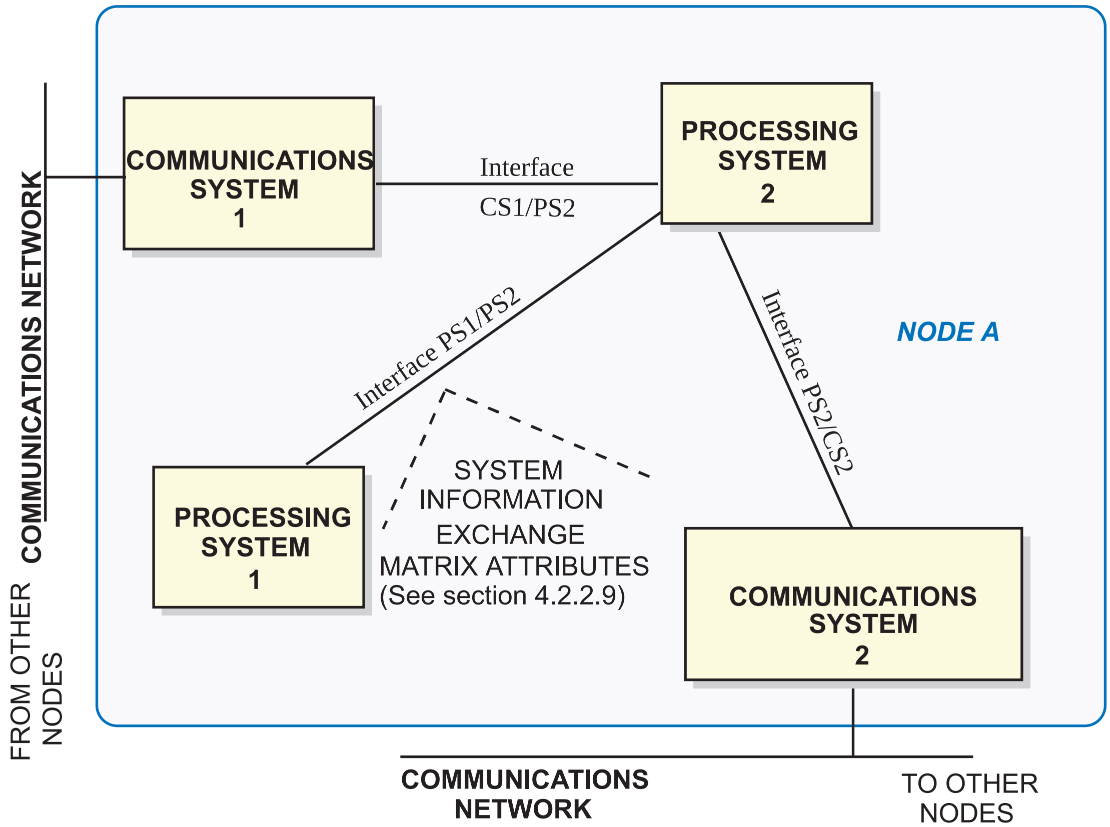


Figure 4-9a. System Interface Description, Intranodal Perspective (SV-1) — Template

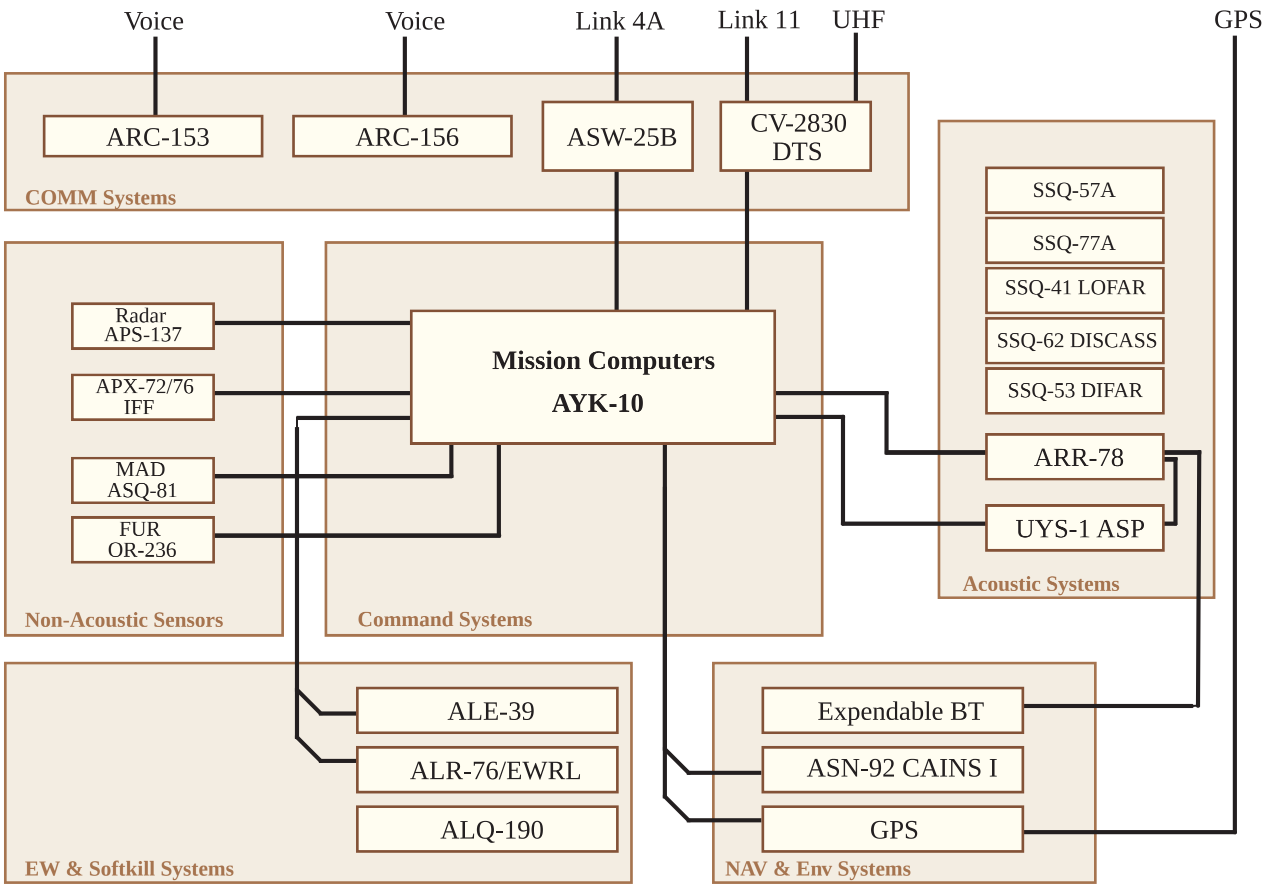
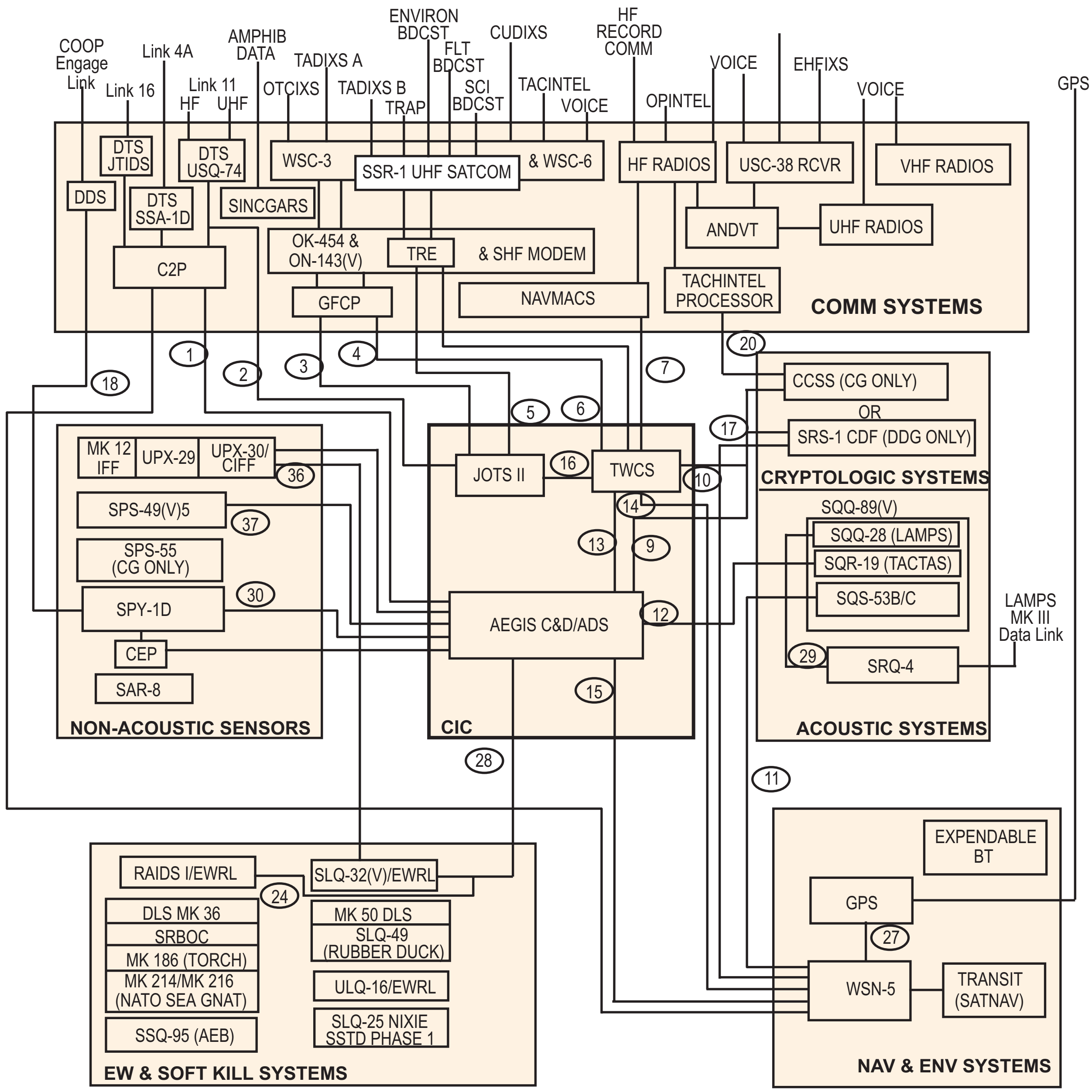


Figure 4-9b. System Interface Description, Intranodal Perspective (SV-1) — Navy Example



Interface Control Document

	From	To	REF No.
17	CCSS	AEGIS C&D/ADS	WS 19656/1
1	AEGIS C&D/ADS	C2P	WS 21324A
2	LINK 11	JOTS II	WS 21331
4	GFCP	TWCS	WS 21334
	• • •		

Figure 4-9c. System Interface Description, Intranodal Perspective (SV-1) — CG/DDG AEGIS CIC Example

The *intrasystem (or system component) perspective* of the System Interface Description decomposes each represented system to identify its internal components, component configurations, and component-to-component interfaces. Typically, for each component-level description, the functions of each system component, as well as the component-to-component inputs and outputs, are clearly defined. Note that the *intrasystem perspective* may not be needed in all cases, depending on the purpose of the architecture and the need to dwell on a specific system's configuration.

The intrasystem perspective can be used to analyze and improve the configuration of systems and system infrastructures (e.g., local area networks [LANs]), e.g., to determine more efficient distribution of software applications. In conjunction with the System Performance Parameters Matrix (described in section 4.2.2.8) and the Technical Architecture Profile (described in section 4.2.1.8), the system component perspective can be used to examine interoperability problems.

Figures 4-10a and 4-10b provide a template and a notional example, respectively, of the intrasystem perspective of the System Interface Description. Figures 4-10c and 4-10d present actual examples.

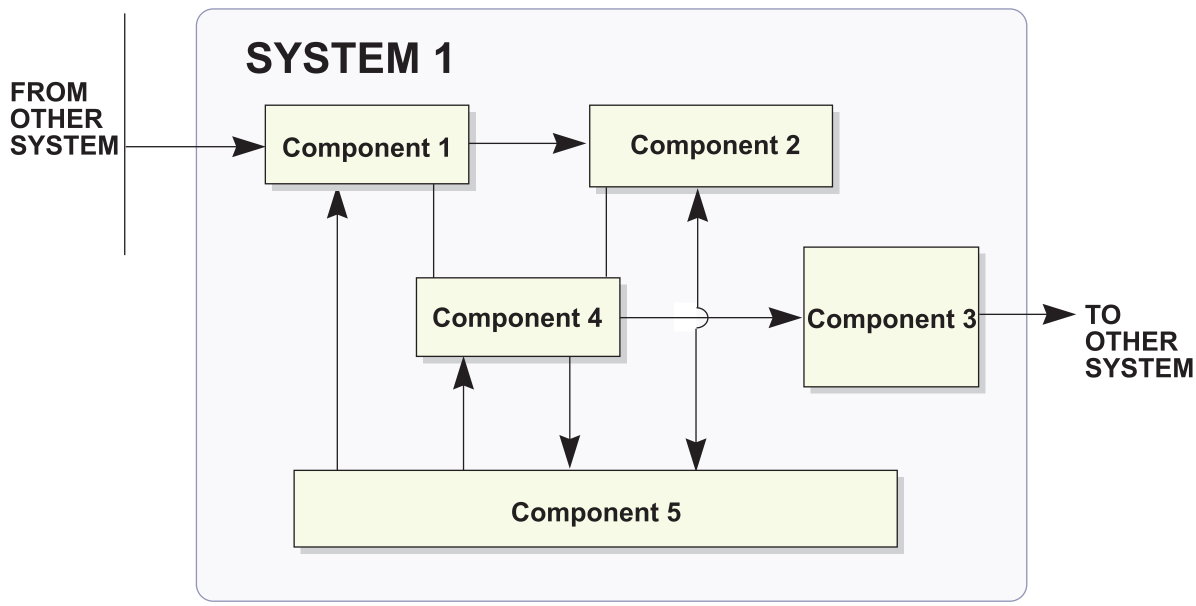


Figure 4-10a. System Interface Description, Intrasystem Perspective (SV-1) — Template

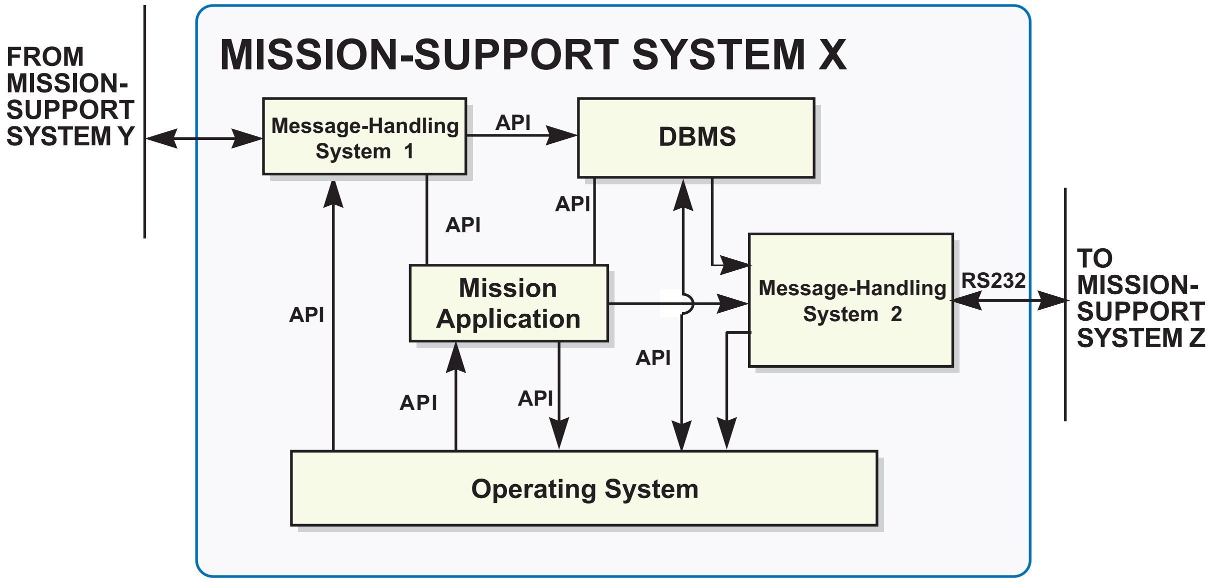


Figure 4-10b. System Interface Description, Intrasystem Perspective (SV-1) — Notional Example

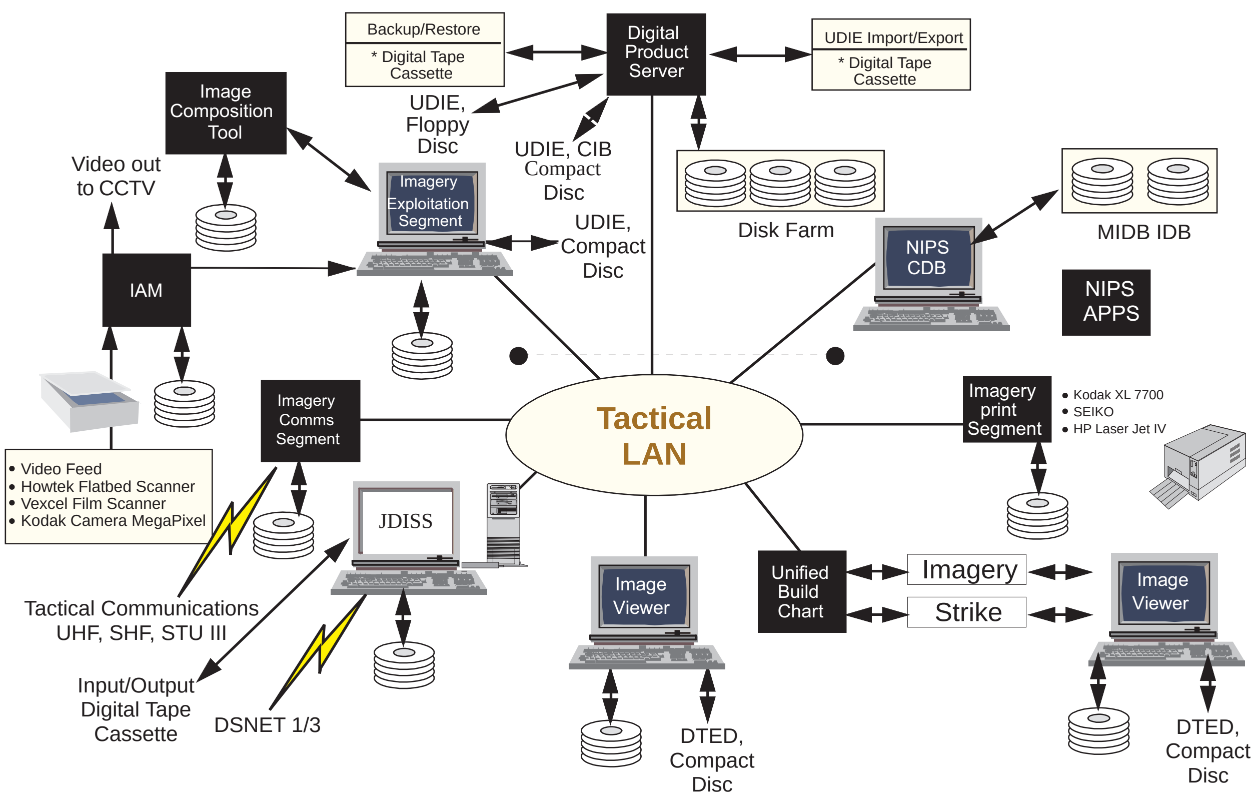


Figure 4-10c. System Interface Description, Intrasystem Perspective (SV-1) — USACOM CIAD 1997 Example

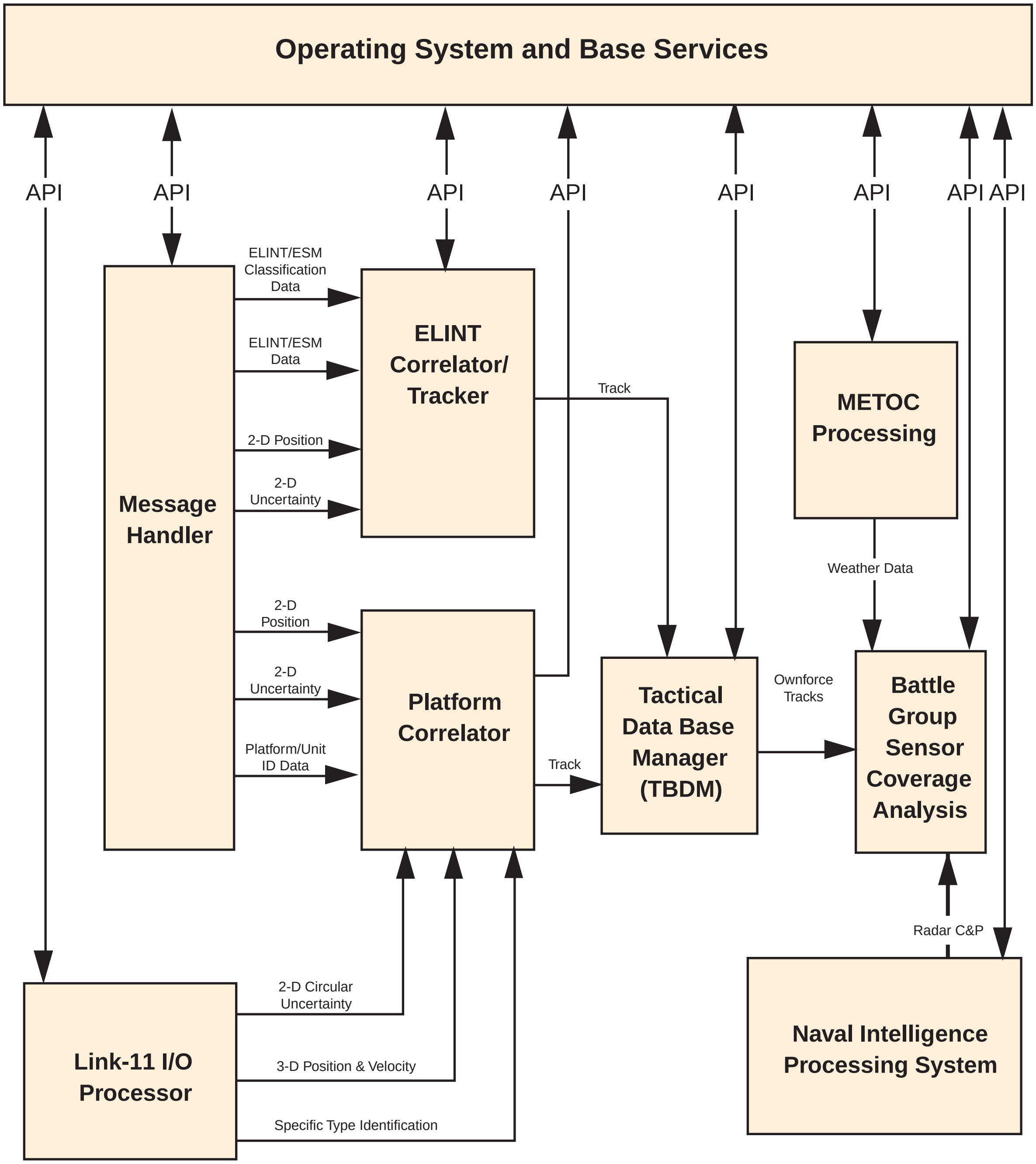


Figure 4-10d. System Interface Description, Intrasystem Perspective (SV-1) — Navy Software System Example

The System Interface Description is categorized as an essential product, meaning that every architecture description that addresses a systems view should include this product. The perspective or perspectives of the System Interface Description that are depicted by the architect will reflect the architecture's specific purpose and details of interest. In some cases, only "node edge-to-node edge" representations of internodal system interfaces may be needed. In other cases, all of the perspectives and representations discussed above may be required.

4.2.1.7 Technical Architecture Profile (TV-1)

Technical View	Essential Product

As defined earlier, the technical component of an architecture is the set of rules that governs system implementation and operation.

In most cases, especially in describing architectures with less than a Service-wide scope, “building” a technical architecture view really will consist of identifying the applicable portions of existing technical guidance documentation, tailoring those portions as needed in accordance within the latitude allowed, and filling in any gaps. Some of these existing guidance documents are described in section 4.3, Universal Reference Resources.

This product references the technical standards that apply to the architecture and how they need to be, or have been, implemented. The profile is time-phased to facilitate a structured, disciplined process of system development and evolution. Time-phasing also promotes the consideration of emerging technologies and the likelihood of current technologies and standards becoming obsolete.

A Technical Architecture Profile constructed as part of a given architecture will be structured appropriately and in accordance with the purposes for which the architecture is being built. Typically, this will involve starting with one or more overarching reference models to which the system is subject and selecting from them the service areas relevant to the system. For example, since real-time operating system variants are outside the scope of a non-real-time system, real-time services would be dropped from further consideration. The identification of relevant services within service areas subsequently points to agreed-upon standards, to which appropriate options and parameters are applied to create a relevant subset for the system. Project standards may be selected when there are no standards which apply to a relevant service.

A notional example of a Technical Architecture Profile with a data management focus is shown in figure 4-11. (Note: The technical criteria shown here are for illustration only.)

Service Area	Service	Standard
Operating System	Kernel	FIPS Pub 151-1 (POSIX.1)
	Shell and Utilities	IEEE P1003.2
Software Engineering Services	Programming Languages	FIPS Pub 119 (ADA)
User Interface	Client Server Operations	FIPS Pub 158 (X-Window System)
	Object Definition and Management	DoD Human Computer Interface Style Guide
	Window Management	FIPS Pub 158 (X-Window System)
	Dialogue Support	Project Standard
Data Management	Data Management	FIPS Pub 127-2 (SQL)
Data Interchange	Data Interchange	FIPS Pub 152 (SGML)
	Electronic Data Interchange	FIPS Pub 161 (EDI)
Graphics	Graphics	FIPS Pub 153 (PHIGS)
. . .		

Figure 4-11. Technical Architecture Profile (TV-1) — Notional Example

4.2.2 Supporting Framework Products

As stated earlier, the supporting products are products that provide additional, supporting data that may sometimes be needed to supplement the essential products. They may provide a graphical representation to facilitate human communication; they may serve as a tabular format for information captured on graphical products, to facilitate populating and manipulating supporting databases; or they may represent incremental steps in producing other products. Depending on the purpose of an architecture description, some of these products may be necessary.

4.2.2.1 Command Relationships Chart (OV-4)

Operational View	**Supporting Product**

The Command Relationships Chart illustrates the relationships among organizations or resources in an architecture. These relationships can include command and control, coordination relationships (which influence what connectivity is needed), and many others, depending on the purpose of the architecture. These relationships are important to show in an operational view of an architecture because they illustrate fundamental roles and management relationships. For example, command and control relationships may differ under different circumstances, as in the three Joint Task Force contingency types. Differing command relationships may mean that activities are performed differently or by different units. Different coordination relationships may mean that connectivity requirements are changed. A template is shown in figure 4-12.

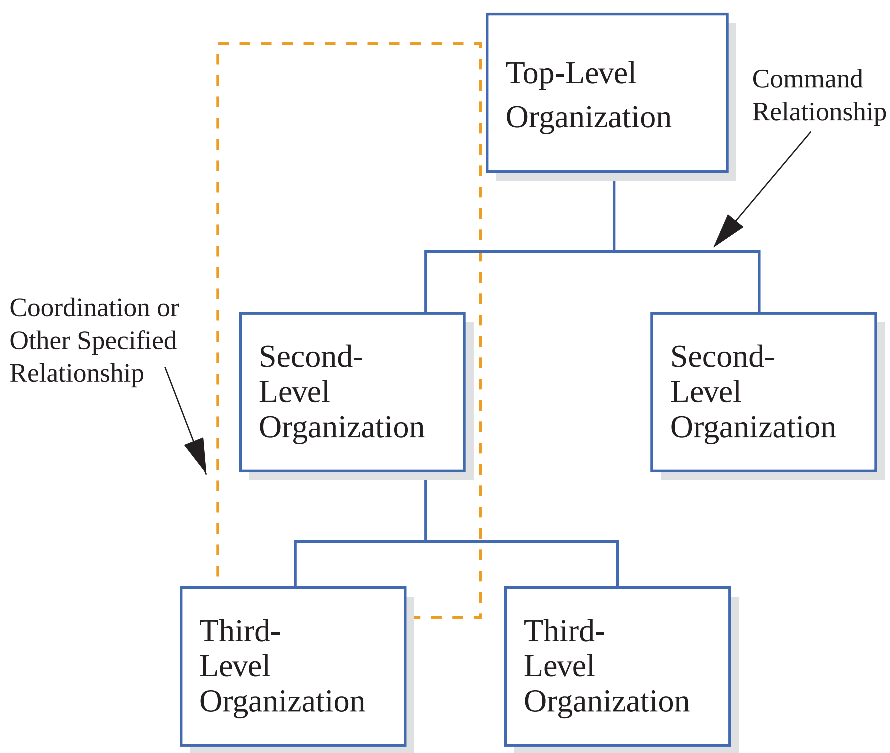


Figure 4-12. Command Relationships Chart (OV-4) — Template

As the template illustrates, boxes can show hierarchies of organizations, and different colors or styles of lines can indicate various types of relationships among the organizations.

Two examples of Command Relationships Charts are illustrated in figures 4-13a and 4-13b.

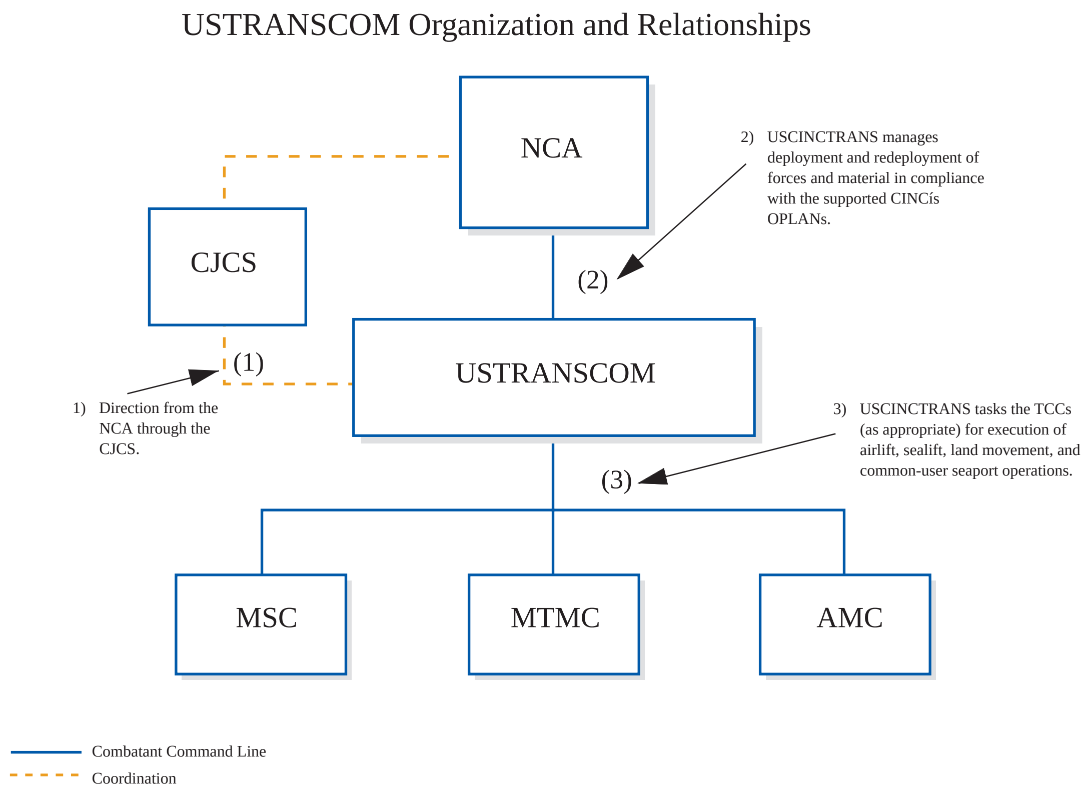


Figure 4-13a. Command Relationships Chart (OV-4) — USTRANSCOM Example

USCENTCOM Targeting

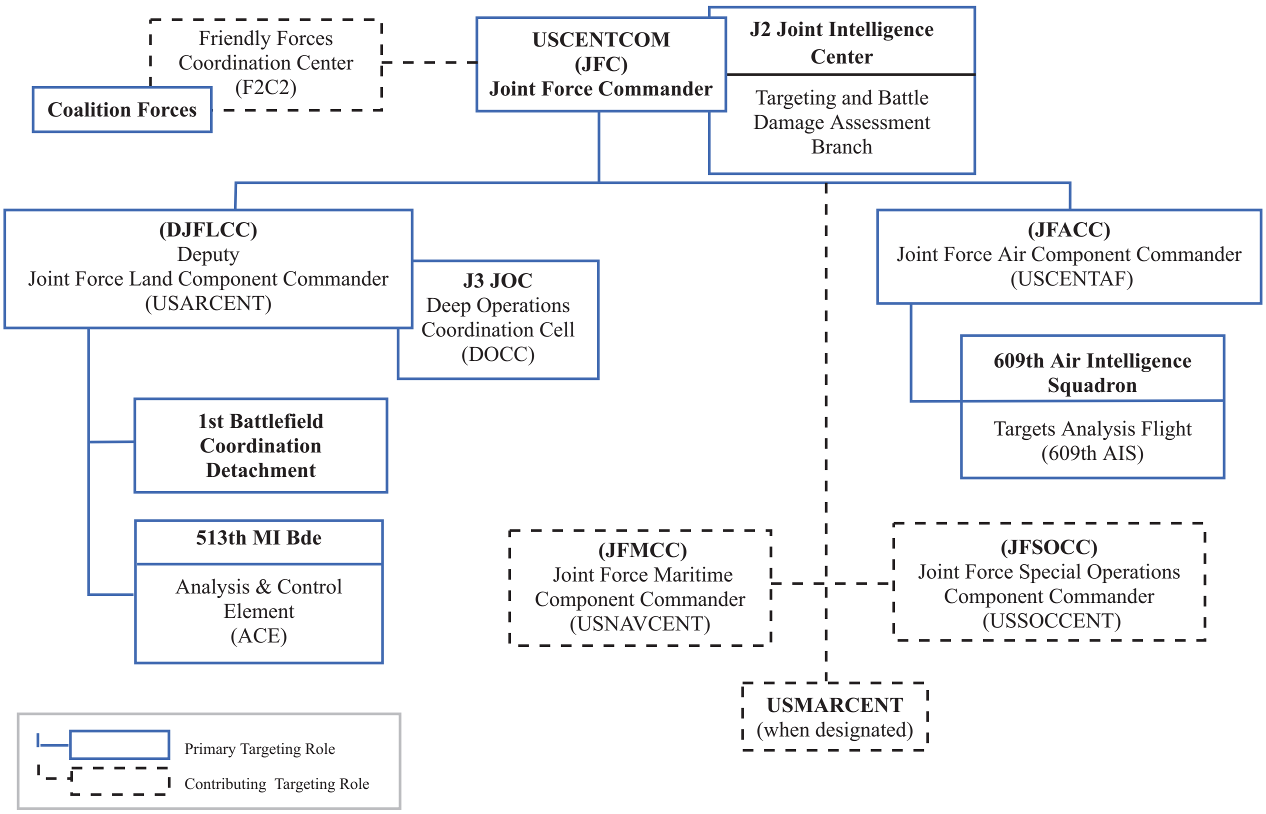


Figure 4-13b. Command Relationships Chart (OV-4) — USCENTCOM Targeting Community Example

4.2.2.2 Activity Model (OV-5)

Operational View	Supporting Product

The Activity Model describes the applicable activities associated with the architecture, the data and/or information exchanged between activities, and the data and/or information exchanged with other activities that are outside the scope of the model (i.e., external exchanges). The models are hierarchical in nature; that is, they begin with a single box that represents the overall activity and proceed successively to decompose the activity to the level required by the purpose of the architecture.

The Activity Model captures the activities performed in a business process or mission and their ICOMs (Inputs, Controls, Outputs, and Mechanisms). Mechanisms are the resources that are involved in the performance of an activity. In addition, the Activity Model identifies the mission domain covered in the model and the viewpoint reflected in the model. Activity definitions and business flows should be provided in additional text, as needed. Annotations to the model may identify the nodes where the activities take place or the costs (actual or estimated) associated with performing each activity.

The Activity Model contributes greatly to the definition and appropriate understanding of an operational architecture. While high-level, conceptual architectures with broad scope and diffused focus may not include activity models, serious consideration should be given to including an activity model in all other architecture efforts.

The Activity Model can capture valuable information about an architecture and can promote the necessary common understanding of the subject area under examination. However, care must be taken to ensure that the modeling process is performed efficiently and usefully, and that the needed information is captured without excessive layers of decomposition and without the inclusion of extraneous information. One way to achieve this efficiency is by using the template model approach. Using this approach, an Activity Model template is constructed and used as a guideline for building multiple models that cover the same set of activities, but from different viewpoints and/or emphasizing different aspects of the activities. The template model specifies the activities, generic ICOM categories, and specific characteristics to be captured in the models. The different viewpoints can be those of multiple organizations that perform similar activities; in that case, the template approach allows those organizations' processes to be compared easily. The objective of this technique is to focus the modeling effort so that a number of small, quickly-developed models can be produced instead of a large, many-layered model that may be cumbersome to use and time-consuming to develop.

The Activity Model generally includes a chart of the hierarchy of activities covered in the model, facing-page text for each diagram that provides any required detail, and a dictionary that defines all activities and terms used in the diagrams. The Integrated Dictionary product serves as this dictionary, and contains all terms used in all of the products constructed for a given architecture, including, but not limited to, the Activity Model.

(Note that in this discussion some terms, such as "ICOM," are used in describing Activity Models. These terms are specific to the Integrated Definition [IDEF0] modeling technique. These terms are used for convenience, because a large community is familiar with them. The use of these terms is not meant to prohibit use of other activity modeling techniques.)

Figure 4-14 depicts templates for the Activity Hierarchy Chart and one level of the Activity Model.

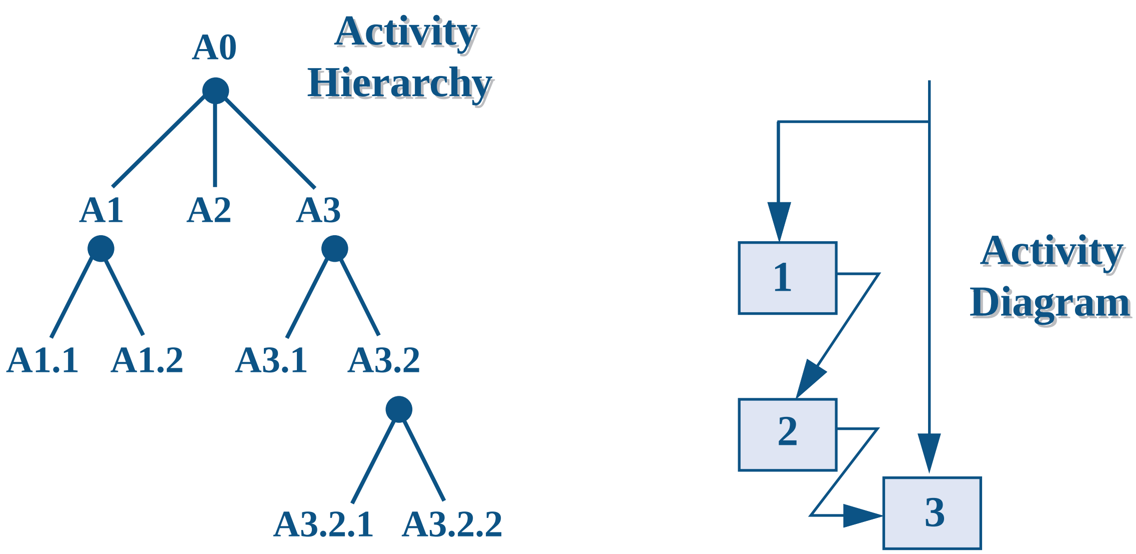


Figure 4-14. Activity Hierarchy Chart and Activity Diagram (OV-5) — Templates

Figures 4-15a through 4-15d provide examples of the Activity Model.

The Activity Model in Figure 4-15a was taken from CISA's *Unifying Guidance for C4I Architecture Development and Representation*. The example illustrates a generic model of intelligence processes and a set of related models that describe intelligence processes at various echelons for support to a deployed Joint Task Force.

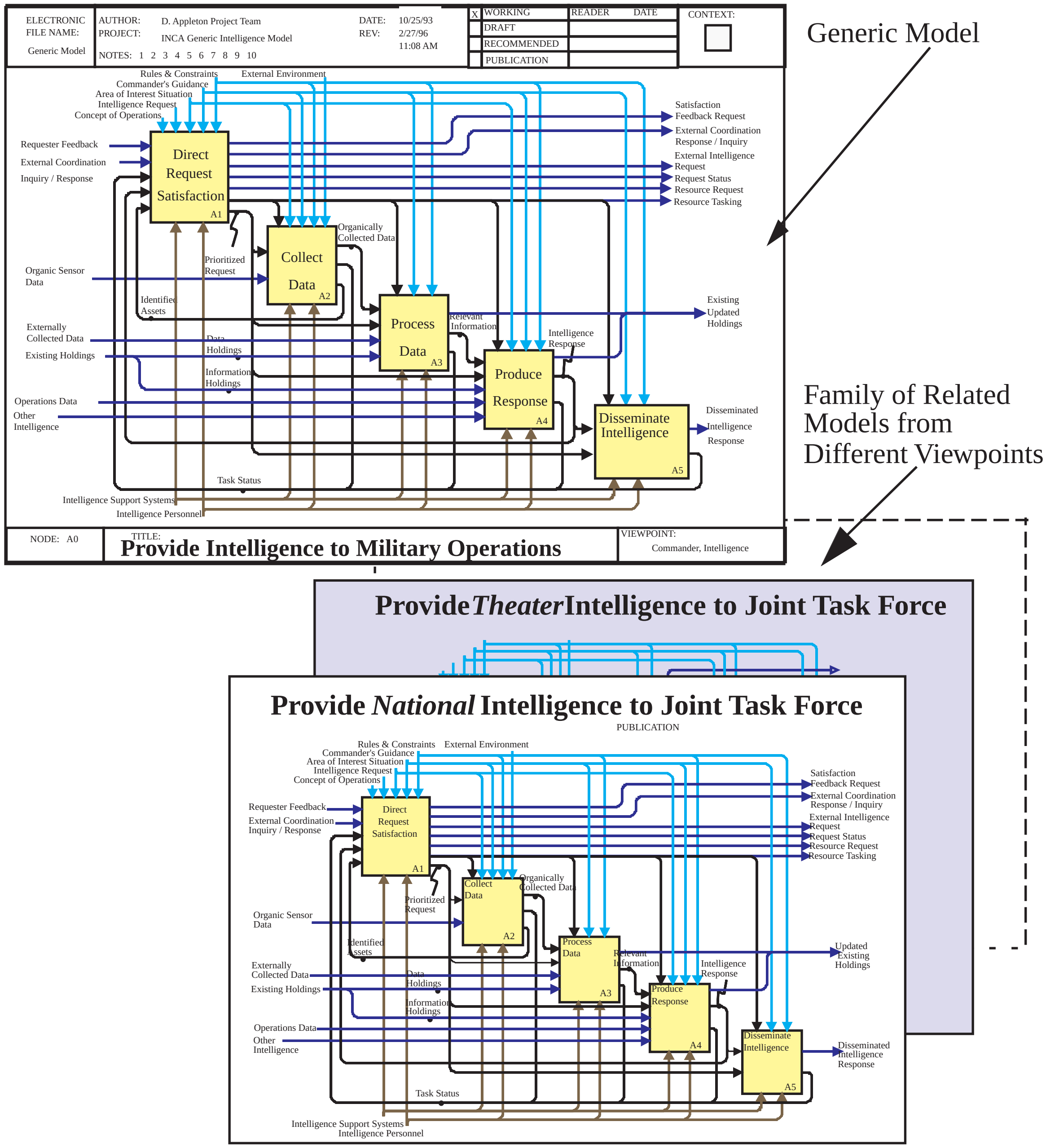


Figure 4-15a. Activity Model (OV-5) — Joint Task Force Intelligence Processes Example

The example in Figure 4-15b depicts the hierarchy of targeting activities from a Joint Task Force perspective.

Figure 4-15c provides an example of a multi-node activity model. This example is very similar to an Operational Node Connectivity Description but with activities at each node portrayed in detail, rather than at the high level usually shown in an Operational Node Connectivity Description.

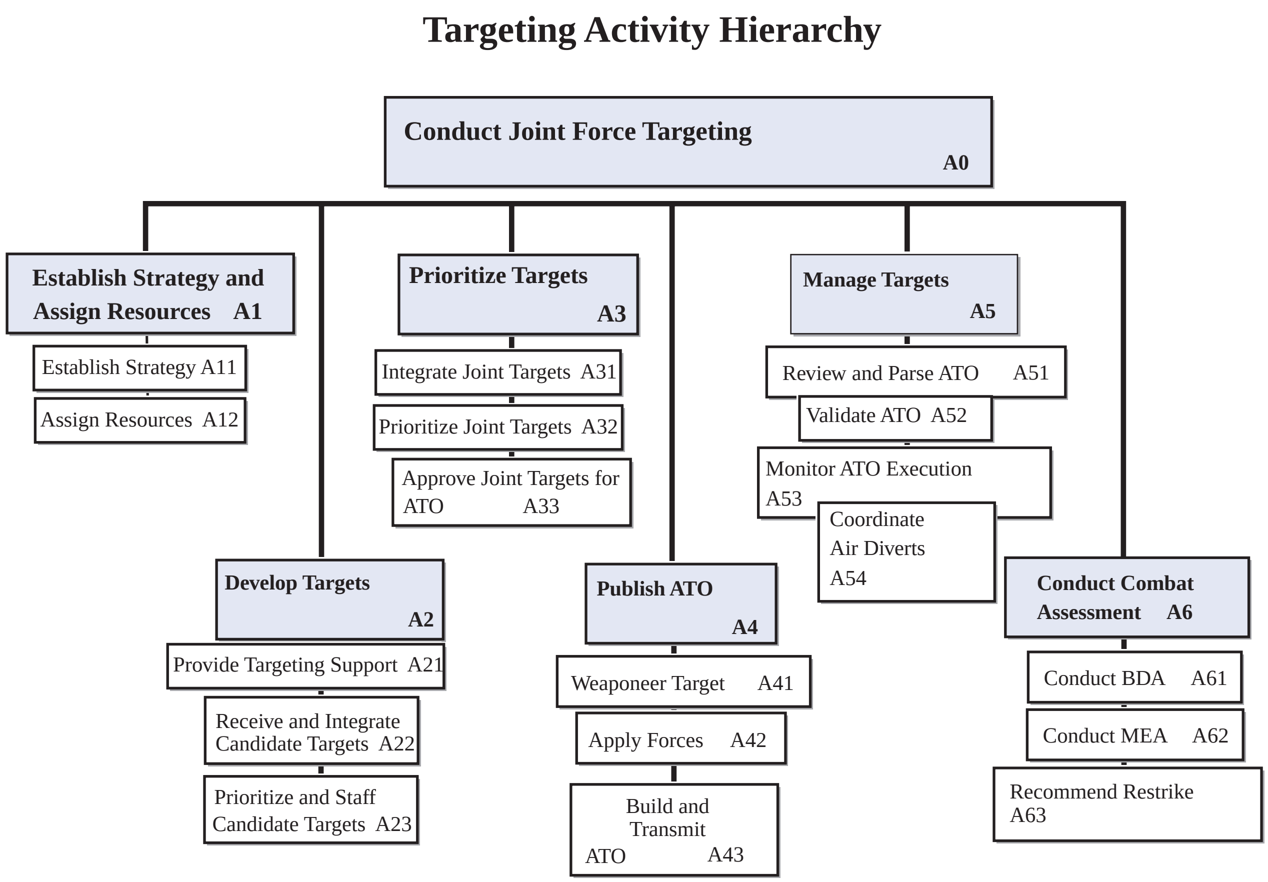


Figure 4-15b. Activity Model (OV-5) — Joint Task Force Targeting Example

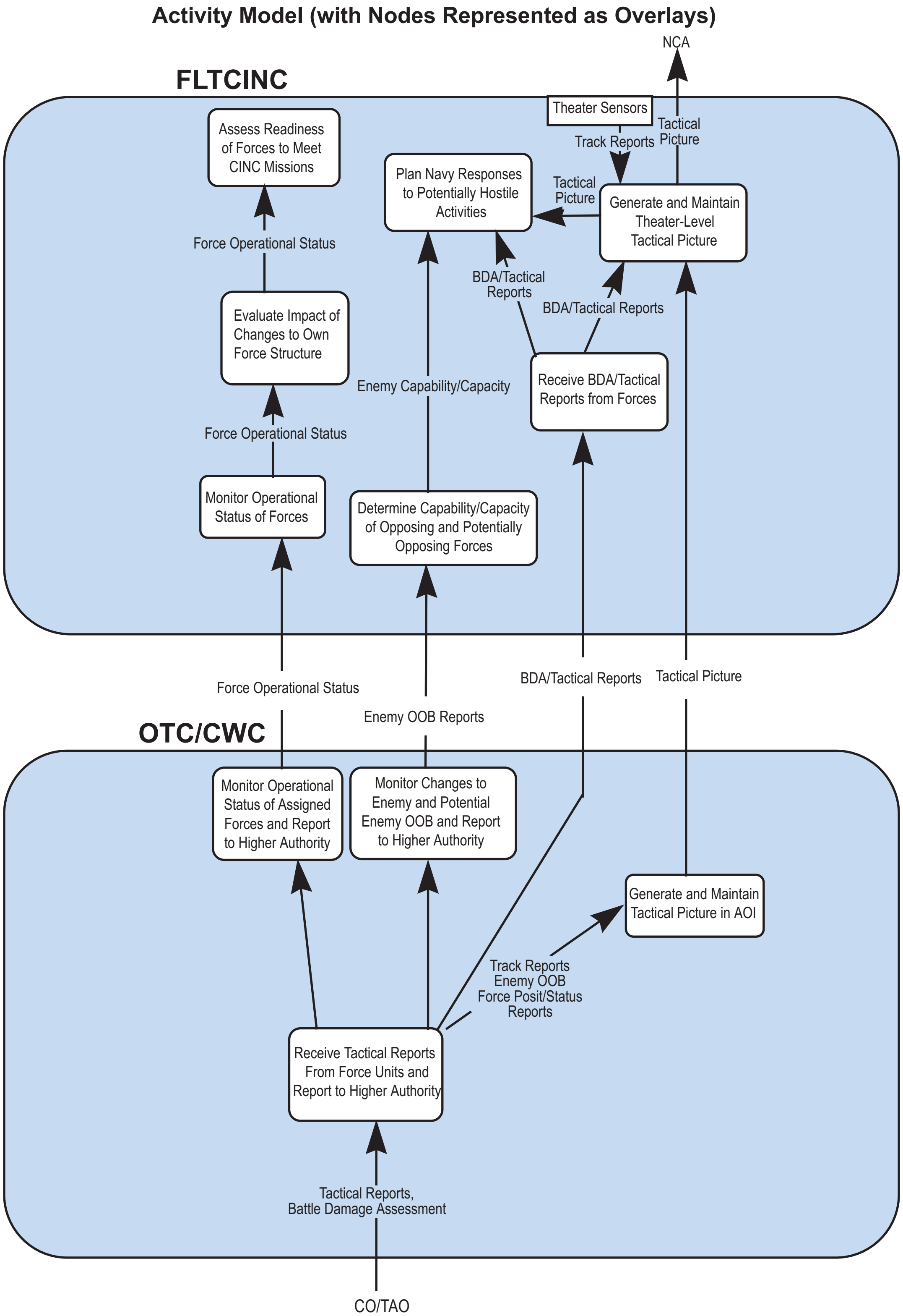


Figure 4-15c. Activity Model (OV-5) — Multi-Node Example

The example in figure 4-15d is taken from the Intelligence Systems Secretariat (ISS) *Broadcast/Receive Working Group Final Report* that CISA produced, and shows the high-level depiction of the activities performed by the TRAP/TDDS intelligence broadcast service. In the working group's effort, four broadcast services were compared for the purpose of highlighting relationships and opportunities for streamlining and consolidation. A generic model of UHF intelligence broadcast activities was developed, then the generic model was tailored to depict each broadcast service's individual variations on the generic activities. Thus, the dotted-line boxes, with no inputs or outputs, represent generic activities that are not performed by TRAP/TDDS, although they are performed by one or more of the other broadcast services. In this way, the single-diagram, high-level activity models of the four broadcast services were readily compared.

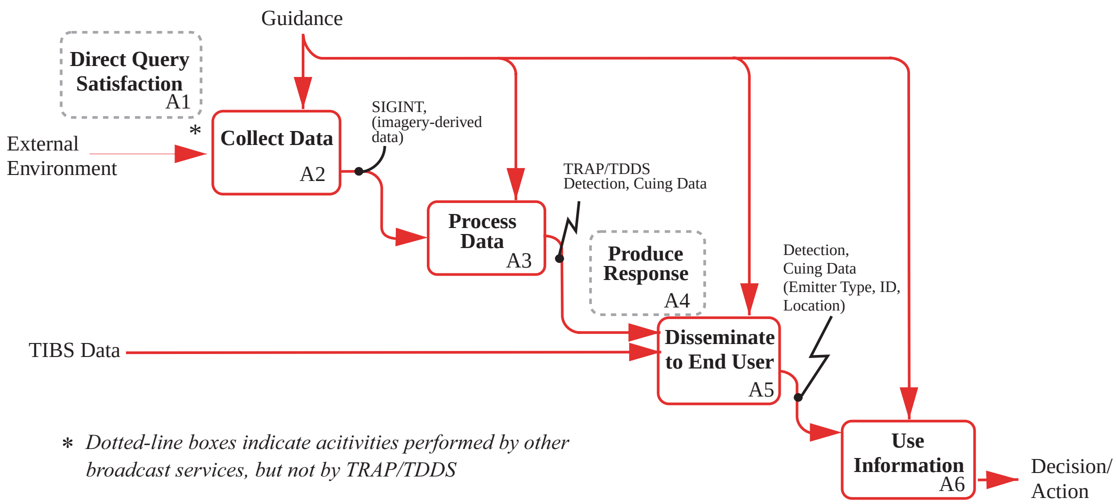


Figure 4-15d. Activity Model (OV-5) — Intelligence UHF Broadcast Service Example

Overlays to Activity Models. One way to get the most out of a relatively small activity modeling effort is to overlay additional information onto the basic diagrams to gain greater insight without the need for additional decomposition. Nodes that perform an activity can be indicated on the appropriate activity box. (Note: This kind of annotation is a standard part of the IDEF0 methodology, and is used in the preceding example. This kind of annotation could also be added when other methodologies are used.) For example, costs of performing the activity can be indicated, and specific attributes of exchanged information can be added to the arrow labels. If such annotations and overlays are designed carefully, the purposes of the architecture description can be furthered with relatively little extra effort.

Figure 4-16 is an Activity Model template showing some notional overlays.

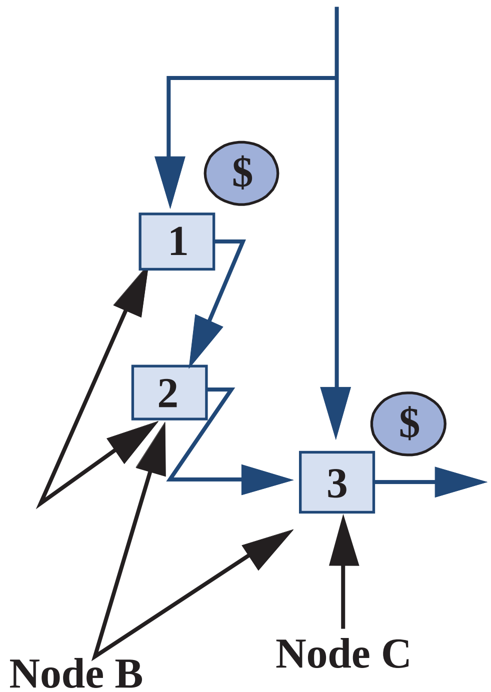


Figure 4-16. Activity Model (OV-5) — Template with Notional Overlays

The black arrows indicate which nodes (e.g., "mechanisms" in IDEF0 terminology) perform which activities (this information can be used to uncover unnecessary functional redundancy). The dollar signs indicate that the costs of performing an activity could be appended as well. Activity-based cost information can be used to make decisions about streamlining, combining, or omitting activities. Overlays can also be used to set "flags" regarding issues, opportunities, or areas to be scrutinized further.

Figure 4-17 shows an Activity Model with overlays that identify the nodes that perform given activities. This figure is the same as figure 4-15d, with the addition of IDEF0 mechanism arrows (the arrows that enter the boxes from the bottom edge, and that indicate who or what performs the activity). Note that in addition to the nodes, arrows have also been overlaid to indicate selected systems; this is not an activity-model convention prescribed in the Framework, but it was effective for this particular effort.

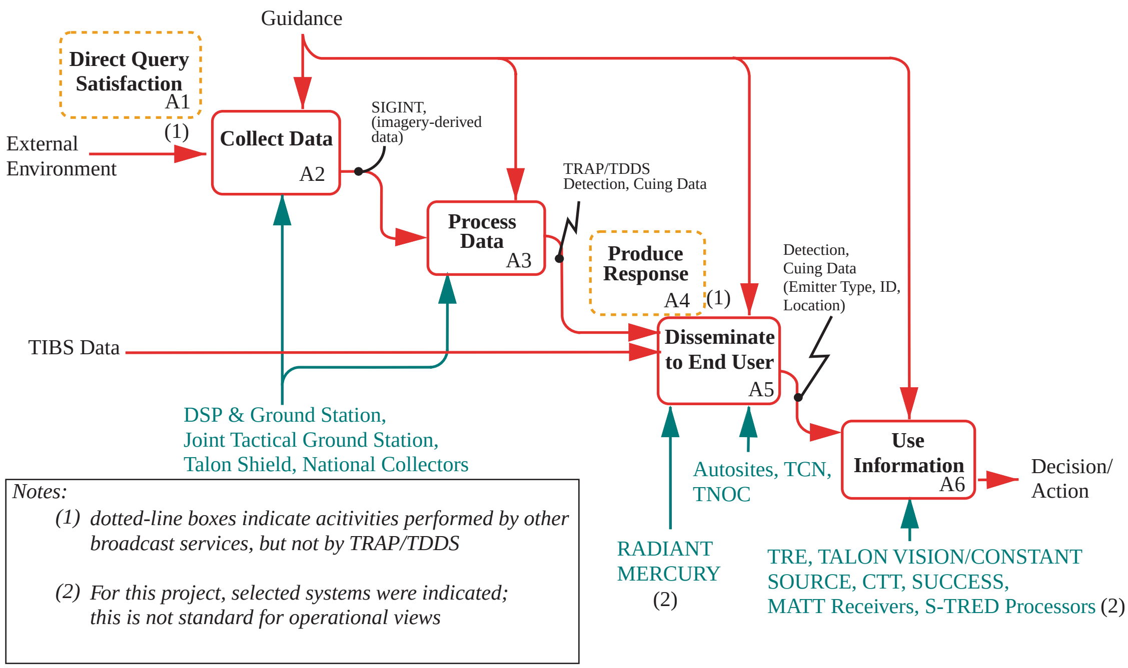


Figure 4-17. Activity Model (OV-5) — UHF Intelligence Broadcast Service Example (with Overlays)

4.2.2.3 Operational Activity Sequence and Timing Descriptions

(OV-6a, 6b, and 6c)

Operational View	**Supporting Product**

Many of the critical characteristics of an architecture are only discovered when an architecture's dynamic behaviors are defined and described. The dynamic behavior referred to here concerns the timing and sequencing of events that capture operational behavior of a business process. Three types of models are needed to refine and extend the architecture's operational view to adequately describe the dynamic behavior and performance characteristics of an architecture. These three models are:

- Operational Rules Model (OV-6a)
- Operational State Transition Description (OV-6b)
- Operational Event/Trace Description (OV-6c)

The Operational State Transition Description and the Operational Event/Trace Description may be used separately or together, as necessary, to describe critical timing and sequencing behavior in the operational view. Both types of diagrams are used by a wide variety of different Business Process methodologies.

The Operational State Transition Description and the Operational Event/Trace Description describe business-process responses to sequences of events. Events may also be referred to as inputs, transactions or triggers. When an event occurs, the action to be taken may be subject to a rule or set of rules as described in the Operational Rules Model.

4.2.2.3.1 Operational Activity Sequence and Timing Descriptions – Operational Rules Model (OV-6a)

Operational View	**Supporting Product**

Rules are statements that define or constrain some aspect of the enterprise. The Operational Rules Model is part of the architecture's operational view and extends the capture of business requirements and concept-of-operations information introduced by the Logical Data Model. (The Logical Data Model is described in section 4.2.2.4.) Rules can be grouped into the following categories:

- Structural Assertion: Concerns (business domain) terms and facts that are usually captured by the entities and relationships of entity-relationship models; these reflect static aspects of business rules already captured in the Logical Data Model.

  - Terms: Entities
  - Facts: Association between two or more terms (i.e., relationship)

- Action Assertion: Concerns some dynamic aspect of the business and specifies constraints on the results that actions produce.

  - Condition: Guard or "if" portion of "if-then" statement; if the condition is true, it may signal enforcing or testing of additional action assertions

  - Integrity Constraint: Must always be true (e.g., a declarative statement)

  - Authorization: Restricts certain actions to certain roles or users

- Derivation: Concerns algorithm used to compute a derivable fact from other terms, facts, derivations, or action assertions.

Since the Structural Assertion rules are captured in the Logical Data Model, the Operational Rules Model can focus on the more dynamic Action Assertions and Derivations rules. Additional characteristics of rules include the following:

- Independent of the modeling paradigm used

- Declarative (non-procedural)

- Atomic (indivisible yet inclusive)

- Expressed in a formal language such as:

  - Decision trees and tables
  - Structured English
  - Mathematical logic

- Distinct, independent constructs

- Business-oriented

Each group may select the formal language in which to record its Operational Rules Model, as long as the notation selected is referenced and well-documented.

Example rules are illustrated here using a Logical Data Model fragment extracted from Ballistic Missile Defense (BMD), Active Defense, as shown in figure 4-18a**.** Figure 4-18b provides a legend for the IDEF1X notation used in figure 4-18a**.** Note that the data elements in these figures consist of all the names inside the rounded boxes. The entity name represents a grouping of data elements that make logical sense for the architectural focus area.

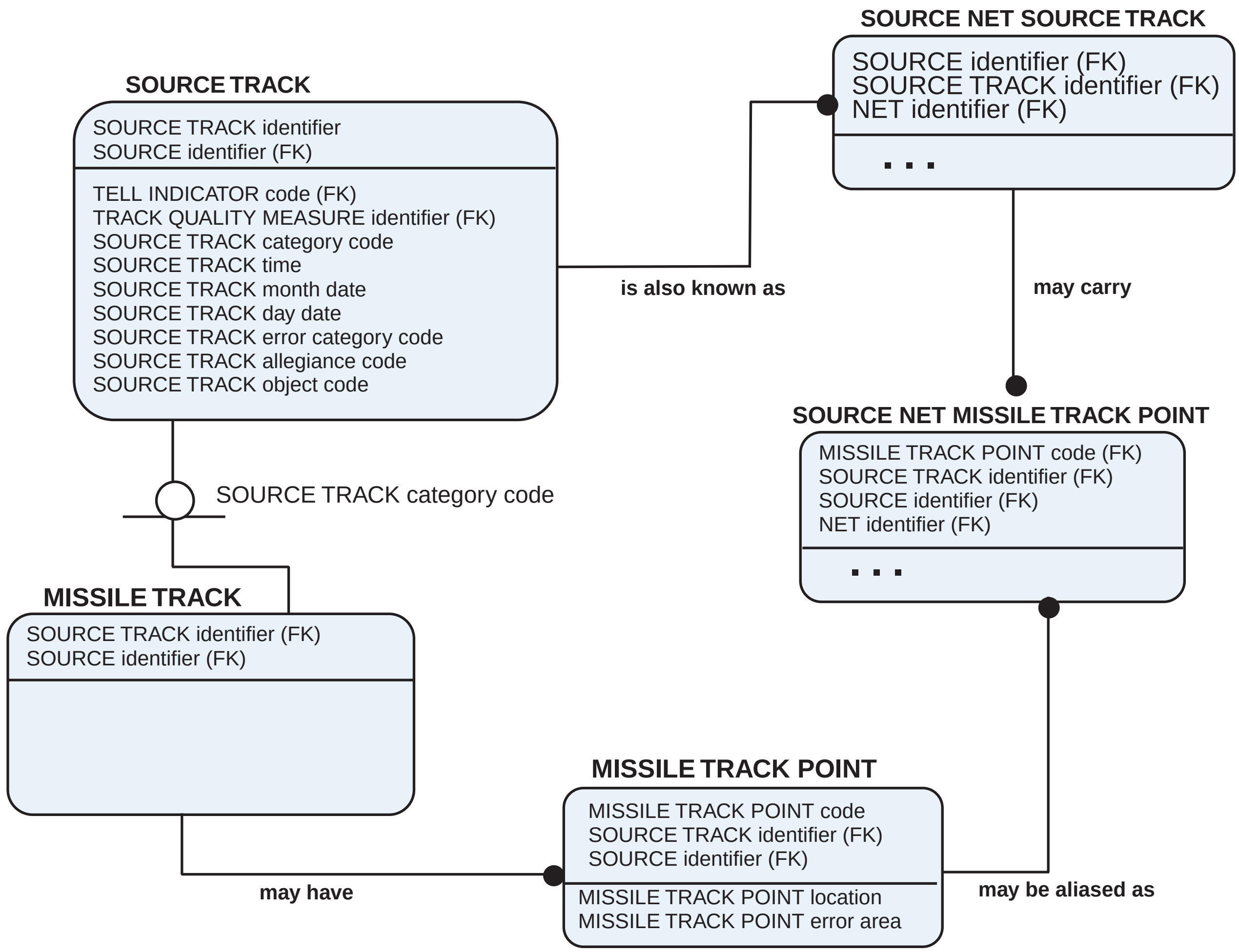


Figure 4-18a. Operational Rules Model (OV-6a) — BMD Active Defense Example Employing a Logical Data Model

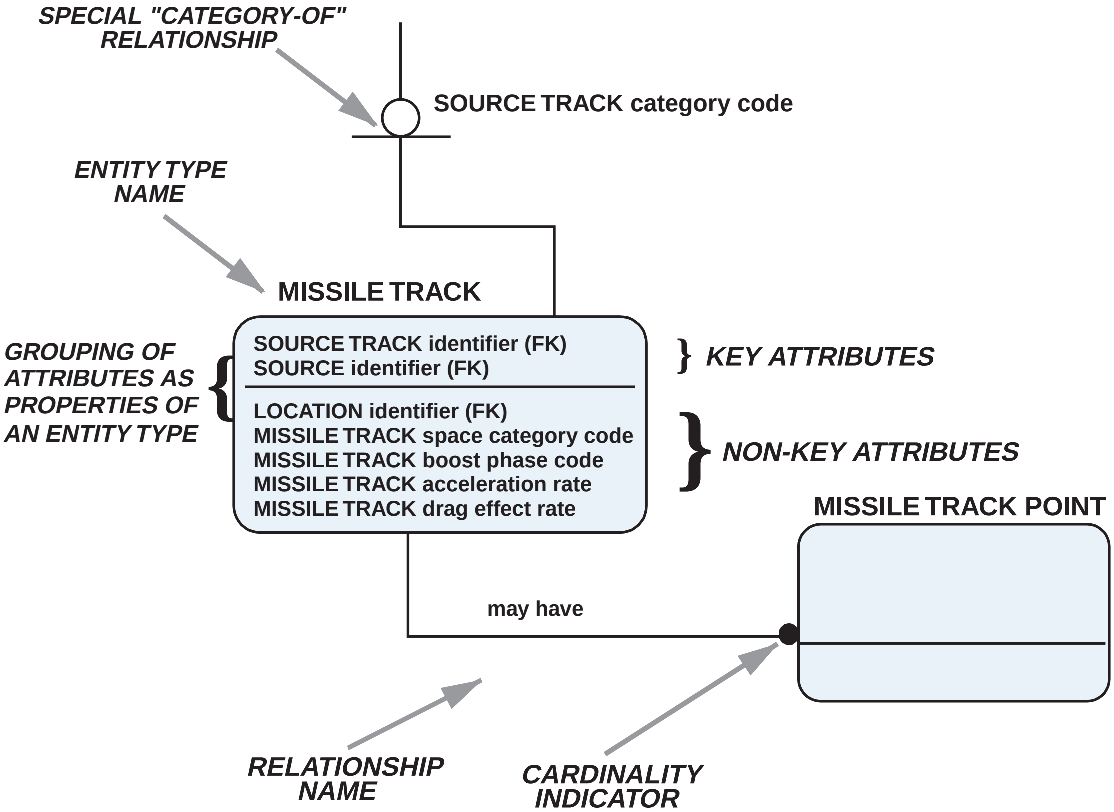


Figure 4-18b. Operational Rules Model (OV-6a) — BMD Active Defense Example Illustrating the Legend for the Logical Data Model

Descriptions of the "operational rules" associated with the definitions of relationships are stored in the Integrated Dictionary. While some operational rules are simple and pertain solely to the relationship, others are more complex and describe the conditions under which potentially null attributes (i.e., data elements that don't have to receive values) must have values and when optional relationships must be present. For example, with respect to the BMD examples in the figures, a possible operational rule is that tracks of missiles in the boost phase (i.e., with boost phase code positive) must have a value for the attribute that represents the acceleration of the missile (i.e., MISSILE TRACK acceleration rate), while tracks of missiles not in the boost phase (i.e., no longer under acceleration) must have a value for the attribute that represents the drag effect of the atmosphere on the missile (i.e., MISSILE TRACK drag effect rate) and an associated entity that records the estimated impact point of the missile (i.e., a related Missile Track Point entity).

Figures 4-19a and 4-19b illustrate the same set of related Action Assertions, stated above in informal English, using two different formal languages: a form of structured English (i.e., pseudo-code); and mathematical logic (i.e., predicate calculus). These rules are operational rules because they reflect constraints on the actual business process and not constraints imposed by system design or implementation decisions.

For Each MISSILE TRACK entity instance
    If MISSILE TRACK boost phase code > 0,
        Then MISSILE TRACK acceleration rate *is non-null*
        Else MISSILE TRACK drag effect rate *is non-null*
            And
                There Exists a MISSILE TRACK POINT entity instance *Such That*
                    MISSILE TRACK.SOURCE TRACK identifier = MISSILE TRACK POINT.SOURCE TRACK identifier
                And

                    MISSILE TRACK POINT.SOURCE identifier
    End If
End For

Figure 4-19a. Operational Rules Model (OV-6a) — BMD Example Illustrating Action Assertion Rules in Structured English

∀ X ε MISSILE TRACK
    (X.boost phase code > 0 ⇒ X.acceleration rate ≠ null
    &
    (X.boost phase code = 0 ⇒ X.drag effect rate ≠ null
        &
        ∃ Y ε MISSILE TRACK POINT ∋
            (X. SOURCE TRACK identifier = Y. SOURCE TRACK identifier
            &
            X. SOURCE identifier = Y. SOURCE identifier)))

Figure 4-19b. Operational Rules Model (OV-6a) — BMD Example Illustrating Action Assertion Rules in Mathematical Logic

Since the Operational Rules Model is a text-oriented product, the Integrated Dictionary captures the type of the rule (e.g., Action Assertion or Derivation) and the text for the rule. Integrated Dictionary attributes derived from this product are under development and include other entries such as the name and description of each action assertion and derivation. See appendix A for a more complete attribute listing with corresponding example values and explanations.

4.2.2.3.2 Operational Activity Sequence and Timing Descriptions — Operational State Transition Description (OV6-b)

Operational View	Supporting Product

A state specifies the response of a system or business process to events. The response may vary depending on the current state and the rule set or conditions. The Operational State Transition Description relates events and states. When an event occurs, the next state depends on the current state as well as the event. A change of state is called a transition. Actions may be associated with a given state or with the transition between states. For example, Operational State Transition Descriptions can be used to describe the detailed sequencing of activities or work flow in the business process. This explicit time sequencing of activities in response to external and internal events is not fully expressed in the Activity Model. The Operational State Transition Description captures this information at the business process level.

Figure 4-20 provides a template for a simple Operational State Transition Description. Initial states (usually one per diagram) are pointed to by the black dot and incoming arrow while terminal states are identified by an outgoing arrow pointing to a black dot with a circle around it. States are indicated by rounded corner box icons and labeled by name or number and, optionally, any actions associated with that state. Transitions between states are indicated by directed lines (i.e., one-way arrows) labeled with the event that causes the transition and the action associated with the transition.

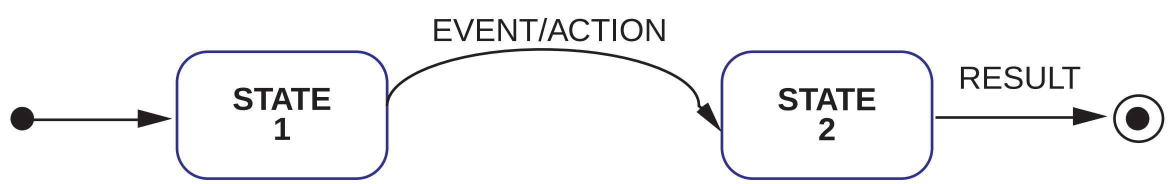


Figure 4-20. Operational State Transition Description (OV-6b) — High-Level Template

Figures 4-20a through 4-20c provide templates for layered structures that can be used to build up a more complex type of state transition diagram known as a Harel State Chart. There is a variety of logically equivalent forms of state transition diagram, but the Harel State Chart is the easiest to use for describing potentially complex, real-world situations, since it allows the diagram to be decomposed in layers showing increasing amounts of detail. Figures 4-20a and 4-20b provide templates for layered states, while figure 4-20c provides a template for a complex transition involving synchronized activities.

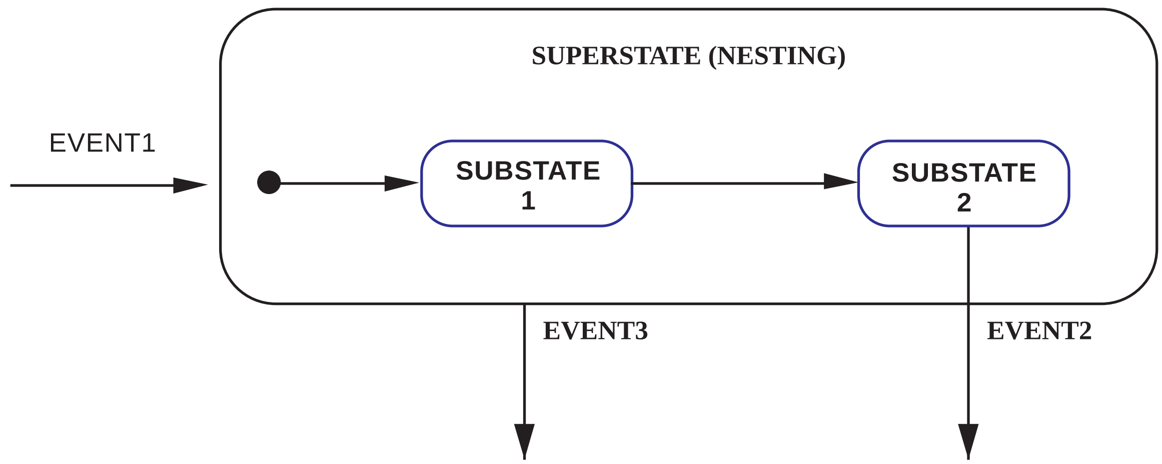


Figure 4-20a. Operational State Transition Description (OV-6b) —
Nested State Structure Template

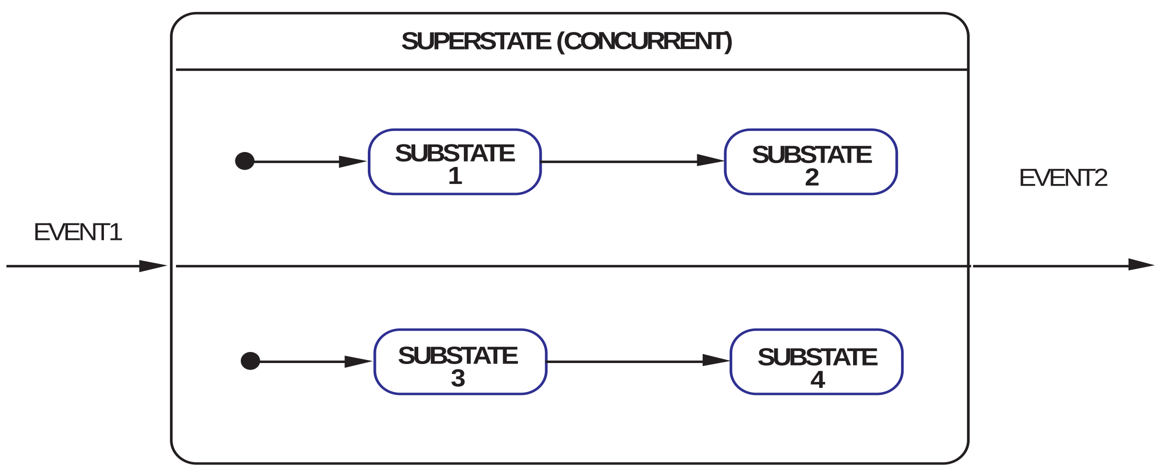


Figure 4-20b. Operational State Transition Description (OV-6b) —
Concurrent Activity State Structure Template

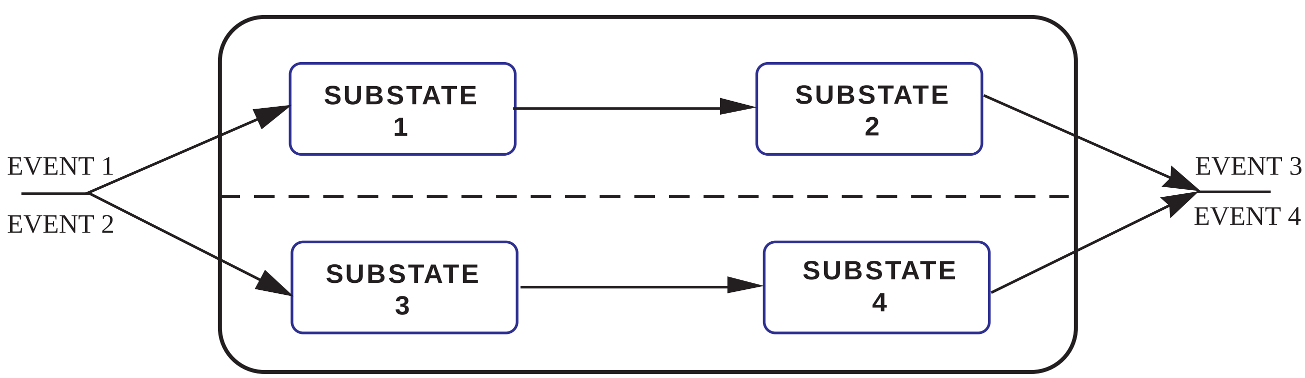


Figure 4-20c. Operational State Transition Description (OV-6b) —
Complex Transition Template

Figure 4-21 illustrates a simple form of Operational State Transition Description for Air Traffic Operations.

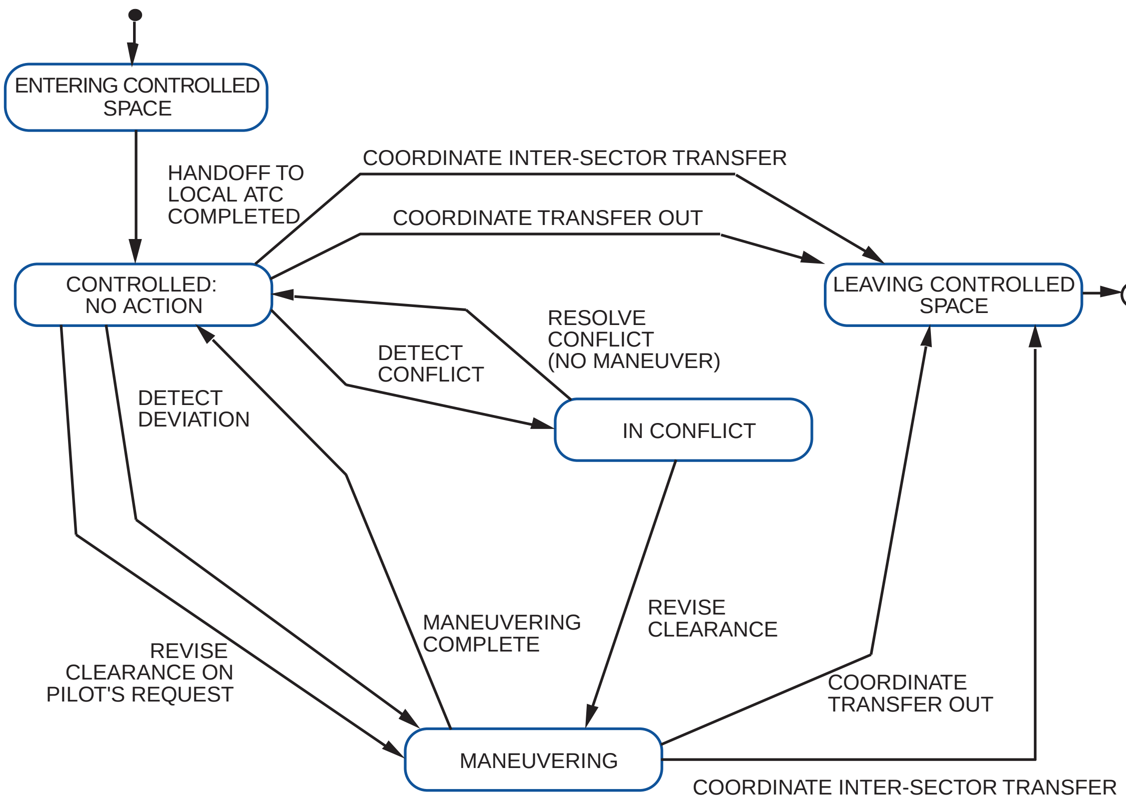


Figure 4-21. Operational State Transition Description (OV-6b) — Air Traffic Operations Example

For activities at the business process level, the Operational State Transition Description captures the states, their names, descriptions, and types (e.g., simple, concurrent superstate), and any actions associated with the states, as well as the transitions, their labels, associated triggering events and resultant actions. Integrated Dictionary attributes derived from this product are under development and describe box types (e.g., state name, description, associated action) and various transition types (e.g., simple, splitting, synchronizing). See appendix A for a more complete attribute listing with corresponding example values and explanations.

4.2.2.3.3 Operational Activity Sequence and Timing Descriptions – Operational Event/Trace Description (OV-6c)

Operational View	**Supporting Product**

Operational Event/Trace Descriptions, sometimes called sequence diagrams, event scenarios, and timing diagrams, allow the tracing of actions in a scenario or critical sequence of events. The Operational Event/Trace Description can be used by itself or in conjunction with an Operational State Transition Description to describe dynamic behavior of processes.

Figure 4-22 provides a template for an Operational Event/Trace Description. The items across the top of the diagram are nodes, usually roles or organizations, which must take action based on certain types of events. Each node has a timeline associated with it which runs vertically. Specific points in time can be labeled running down the left hand side of the diagram. Directed lines between the node time lines represent events, and the points at which they intersect the timelines represent the times at which the nodes become aware of the events. The direction of the event lines represents the flow of control from one node to another based on the event.

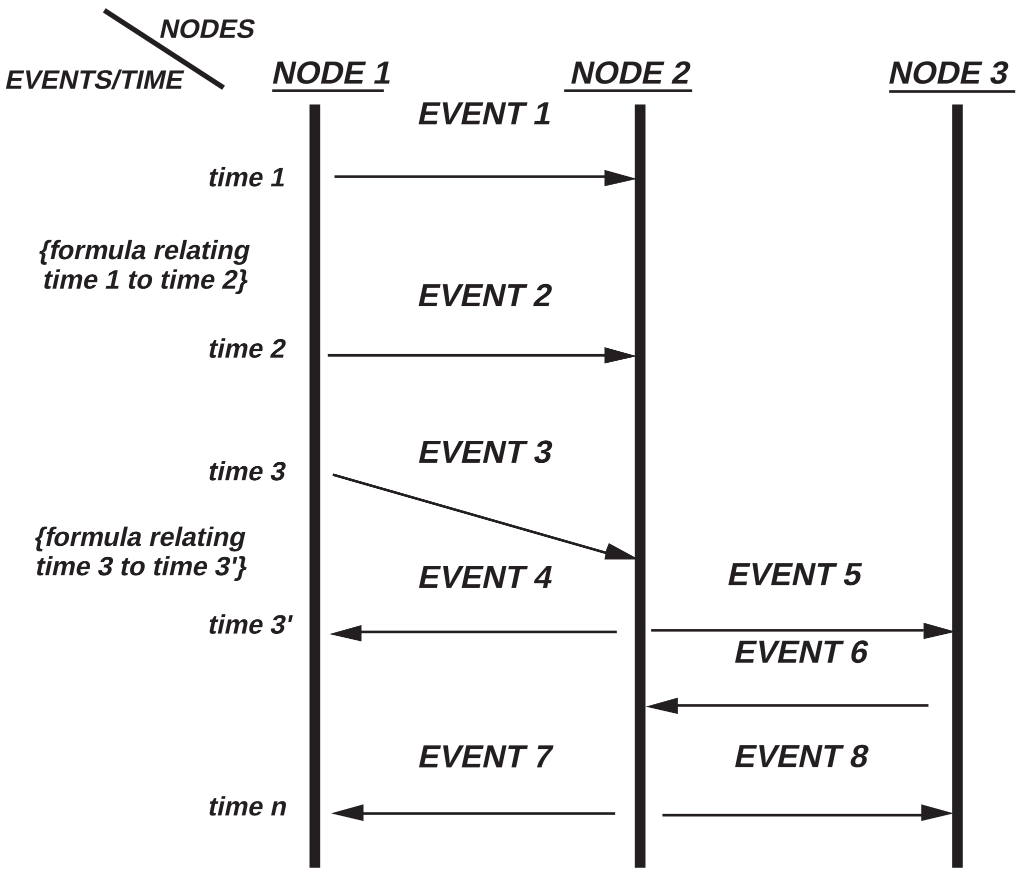


Figure 4-22. Operational Event/Trace Description (OV-6c) — Template

Figure 4-23a provides another example of the Operational Event/Trace Description

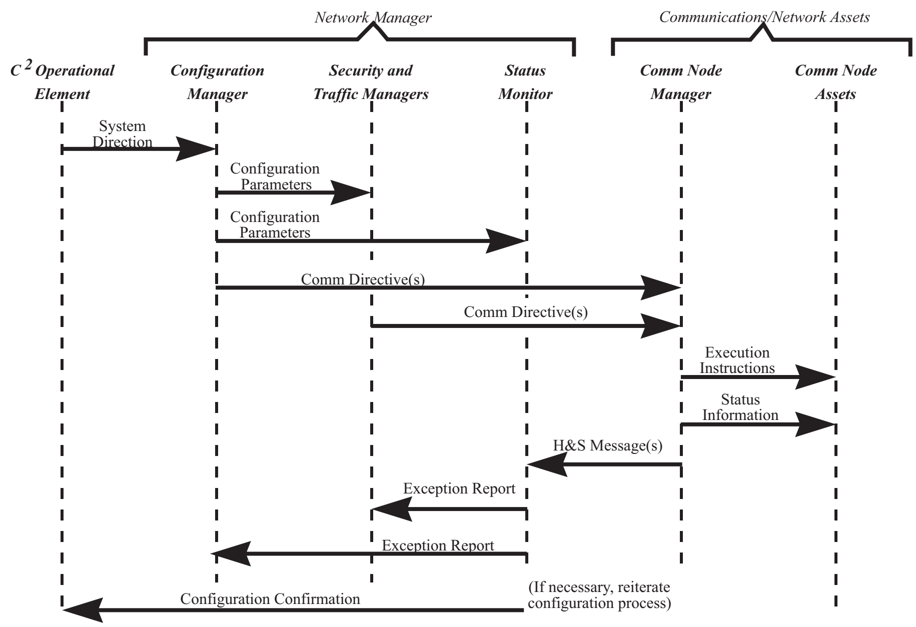


Figure 4-23a. Operational Event/Trace Description (OV-6c) —
Communications Net Management Example

Figure 4-23b provides an example of an Operational State Transition Description (OV-6b) that is related to the Operational Event/Trace Description shown in figure 4-23a.

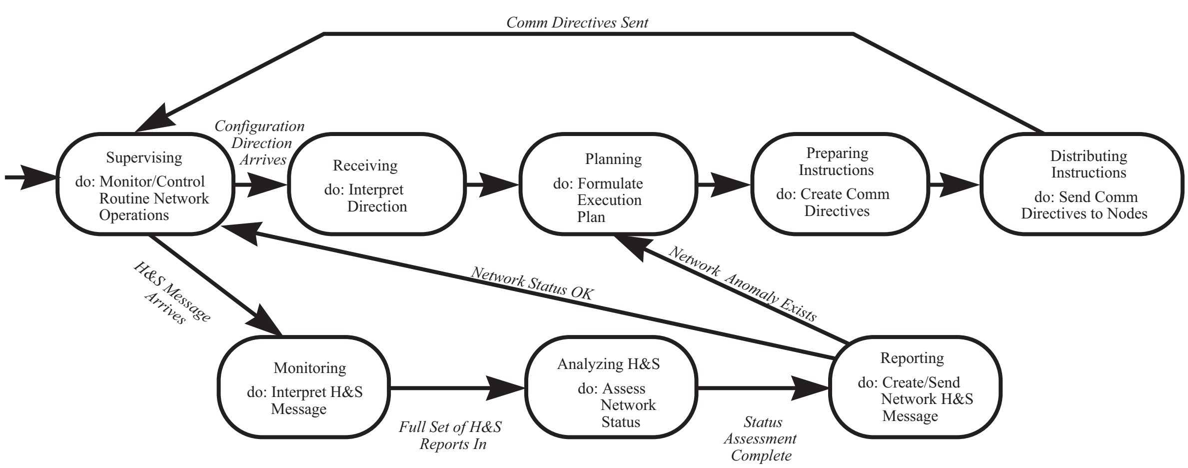


Figure 4-23b. Operational State Transition Description (OV-6b) – Communications Net Management Example

The Operational Event/Trace Description associates nodes with event timelines and cross links that show how events cause related actions in different nodes and the relative time of these actions. Integrated Dictionary attributes derived from this product are under development and include entries describing the node event timeline and cross links (e.g., name, description, originating/terminating node). See appendix A for a more complete attribute listing with corresponding example values and explanations.

4.2.2.4 Logical Data Model (OV-7)

Operational View	**Supporting Product**

The Logical Data Model (LDM) is used to document the data requirements and structural business process rules of the architecture's operational view. It describes the data and information that is associated with the information exchanges of the architecture, within the scope and to the level of detail required for the purposes of the architecture. Included are information items and/or data elements, their attributes or characteristics, and their interrelationships.

Although they are both called data models, the Logical Data Model should not be confused with the C4ISR Core Architecture Data Model (CADM). The Logical Data Model is an architecture product and describes architecture-specific information exchanges. The CADM is not an architecture product. The CADM describes the generic form (i.e., meta-model) of a Logical Data Model, and CADM-based repositories can store Logical Data Models from any Framework-based architecture project. Thus, the CADM addresses the definitions and relationships of generic entities and attributes, while a Logical Data Model for missile defense, for example, might address definitions and relationships for missile tracks and points of impact.

As described earlier, the purpose of a given architecture helps to determine the level of detail needed in this product. A formal "data" model (e.g., IDEF1X) that is detailed down to the level of data, their attributes, and their relationships is required for some purposes, such as when validation of completeness and consistency is required. However, for other purposes, a higher-level information-focused data model of the domain of interest will suffice, such as an entity-relation model without entity attributes. The term "data model" is used here in this context, regardless of the level of detail the model exhibits.

Whatever the purpose of the architecture and the level of detail it exhibits, a Logical Data Model can help discover and document operational information requirements and "business rules." The Logical Data Model can be used as an alternative to the Activity Model, for architectures where an "information-focused" view is desired, or in conjunction with the Activity Model. For example, an information-focused view may be necessary for interoperability when shared data syntax and semantics form the basis for greater degrees of information systems interoperability, or when a shared database is the basis for integration and interoperability among business processes and systems.

There is not a one-to-one mapping between the information items that are shown in the Activity Model and the information/data elements that are described in the Logical Data Model; however, there is considerable mutual influence between these models, and they should be developed together when both are being used.

Figure 4-24a provides a template for a Logical Data Model (with attributes). The format is intentionally generic to avoid implying a specific methodology.

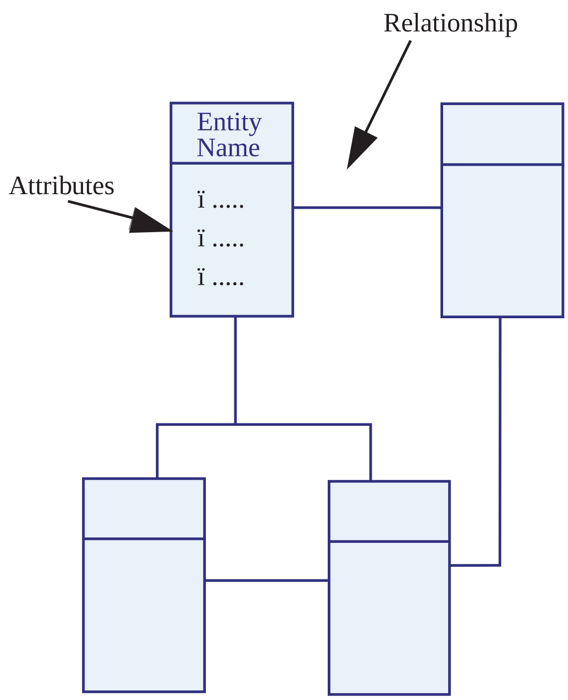


Figure 4-24a. Logical Data Model (OV-7) — Template

A portion of a Logical Data Model (with attributes) that uses the IDEF1X methodology is shown in figure 4-24b. This example illustrates a view of some of the information associated with an Air Tasking Order, and is taken from the document *Battle Damage Assessment (BDA) - Related Military Intelligence (GMI) Production, Dissemination, and Use Functional Process Improvement (FPI) Case Study*, U.S. Air Force.

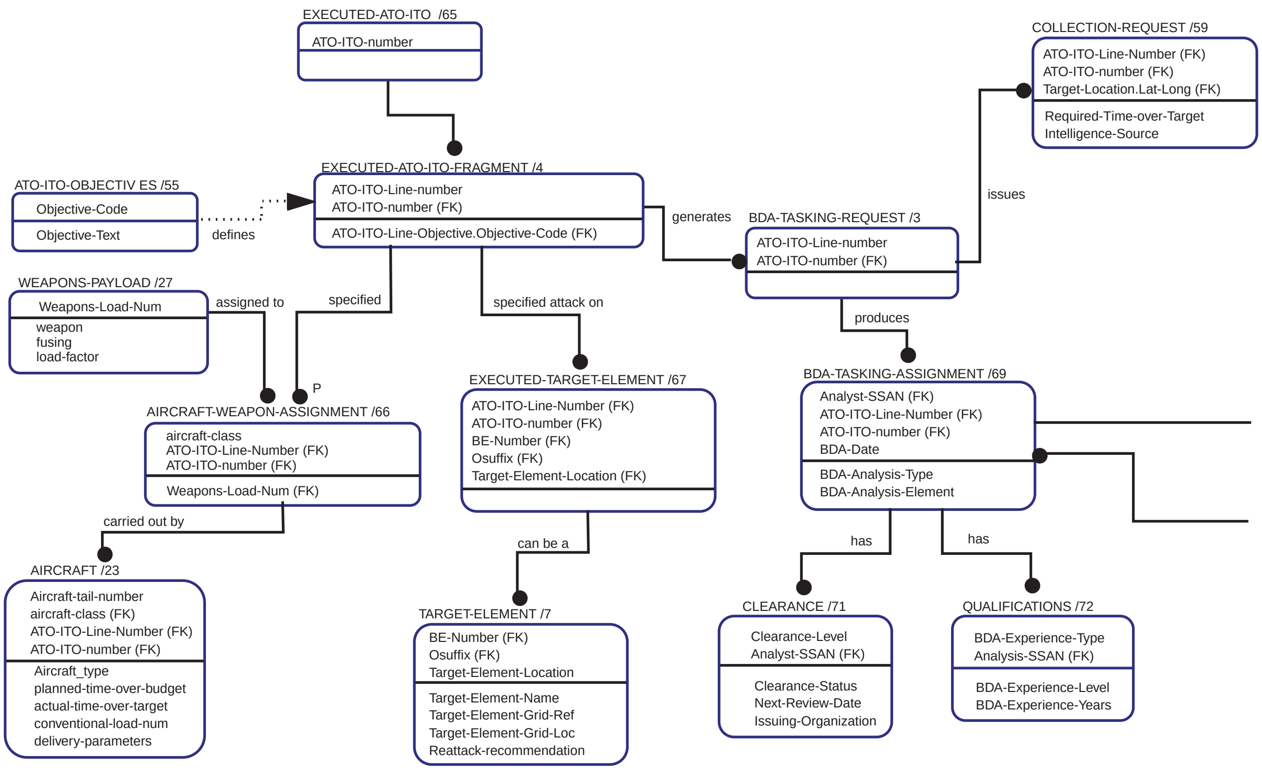


Figure 4-24b. Fully Attributed Logical Data Model (OV-7) — Air Tasking Order Example

4.2.2.5 Systems Communications Description (SV-2)

Systems View	Supporting Product

The Systems Communications Description represents the specific communications systems pathways or networks (e.g., DSCS, Intelink, or JWICS) and the details of their configurations through which the physical nodes and systems interface. This product focuses on the physical aspects of the information needlines represented in the Operational Node Connectivity Description (e.g., text, message standards, etc.), and also depicts pertinent information about communications elements and services (e.g., the kind of processing performed onboard a satellite, the locations of network switches or routers, the existence of amplifiers or repeaters in a particular communications path, or the location of cable “bulkheads” on both shores of an ocean). The graphical presentation and/or supporting text should describe all pertinent communications attributes (e.g., waveform, bandwidth, radio frequency, packet or waveform encryption methods).

Depending on the analytical focus of the architecture, Systems Communications Descriptions detail the interfaces described in the System Interface Description (section 4.2.1.6) and can present either *internodal* or *intranodal* perspectives.

The *internodal perspective* details the communications paths and/or networks that interconnect systems nodes or specific systems (from one node to other nodes).

Figure 4-25a provides a template for the *internodal perspective* of the System Communications Description. Note that this figure translates the single-line representations of interfaces (as shown in figure 4-8a, System Interface Description, Internodal Perspective) into a more detailed representation of the communications infrastructure that provides the connections.

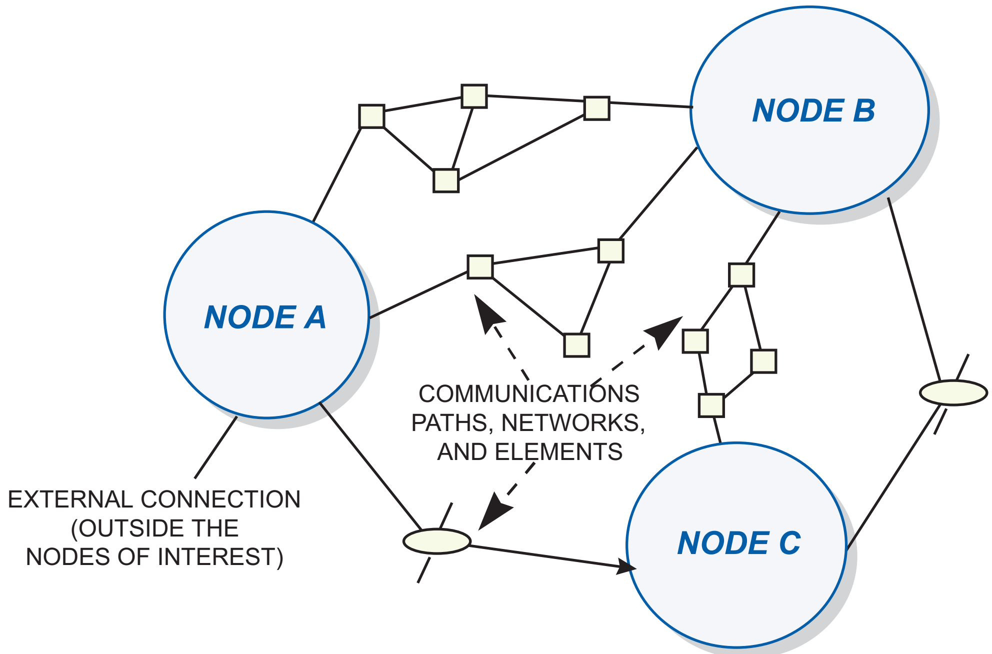


Figure 4-25a. Systems Communications Description, Internodal Perspective (SV-2) —Template

The *intranodal perspective* of the Systems Communications Description looks inside each of the represented nodes to illustrate the interfaces between specific systems.

Figure 4-25b provides a template for the *intranodal perspective* of the Systems Communications Description.

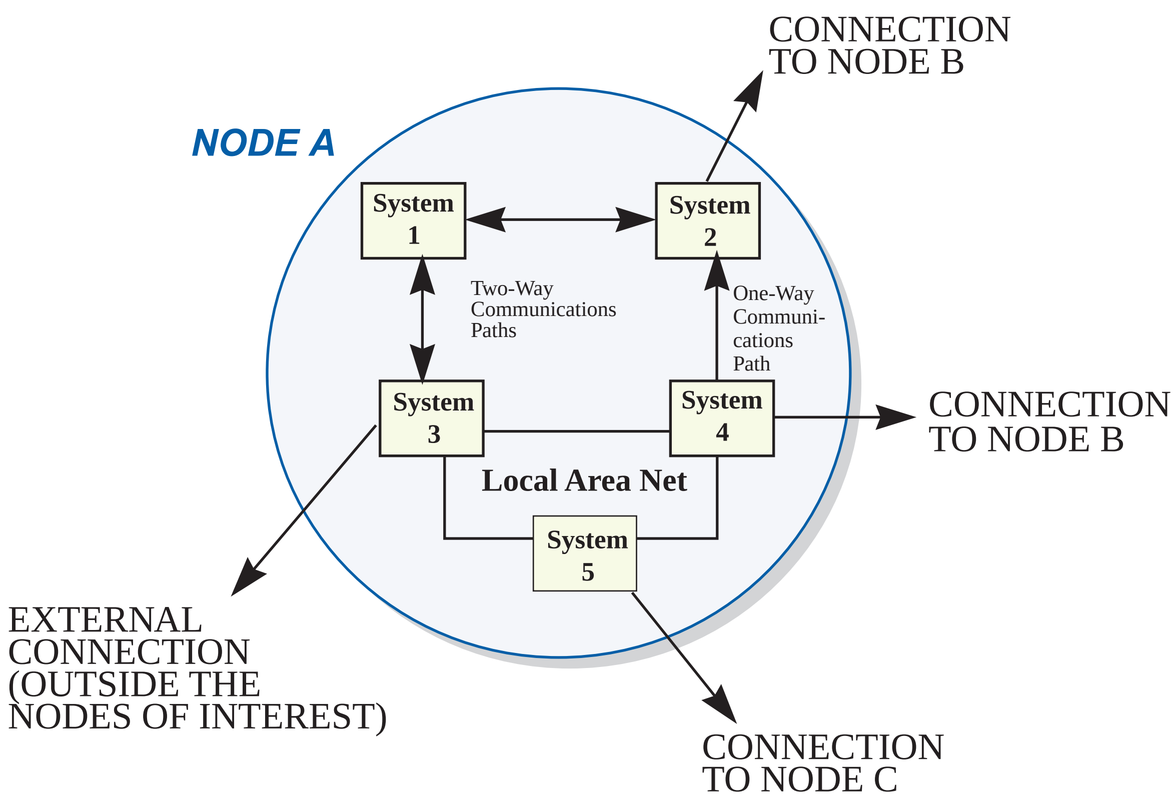


Figure 4-25b. Systems Communications Description, Intranodal Perspective (SV-2) — Template

Figure 4-25c provides a notional example of the *intranodal perspective*, and figure 4-25d provides an actual example.

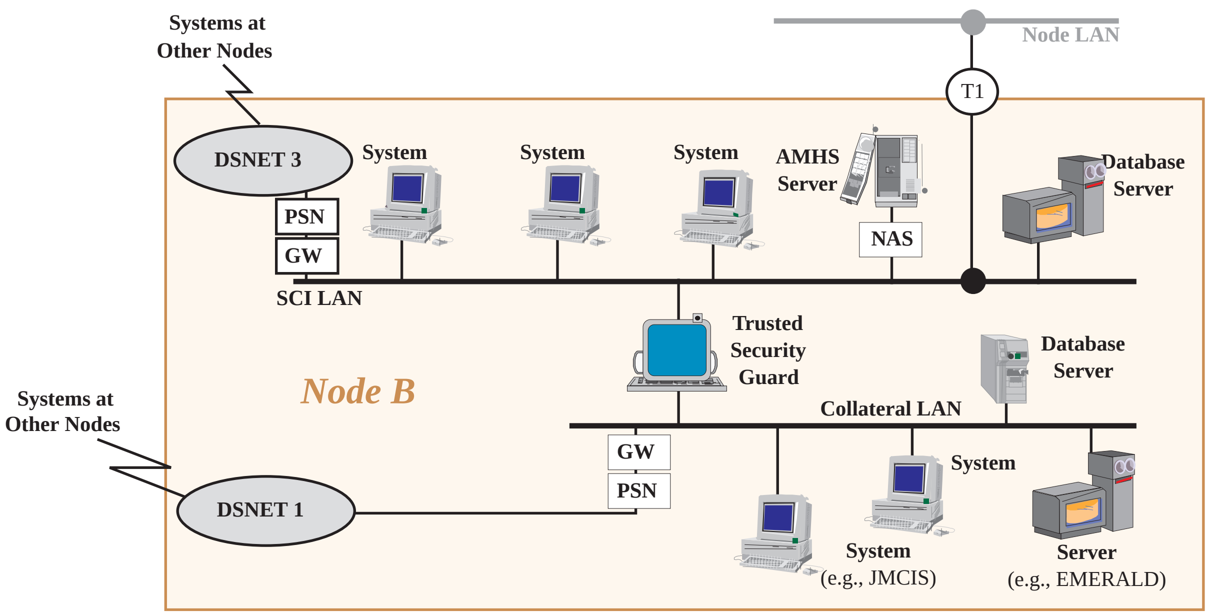


Figure 4-25c. Systems Communications Description, Intranodal Perspective (SV-2) — LAN-Based Notional Example

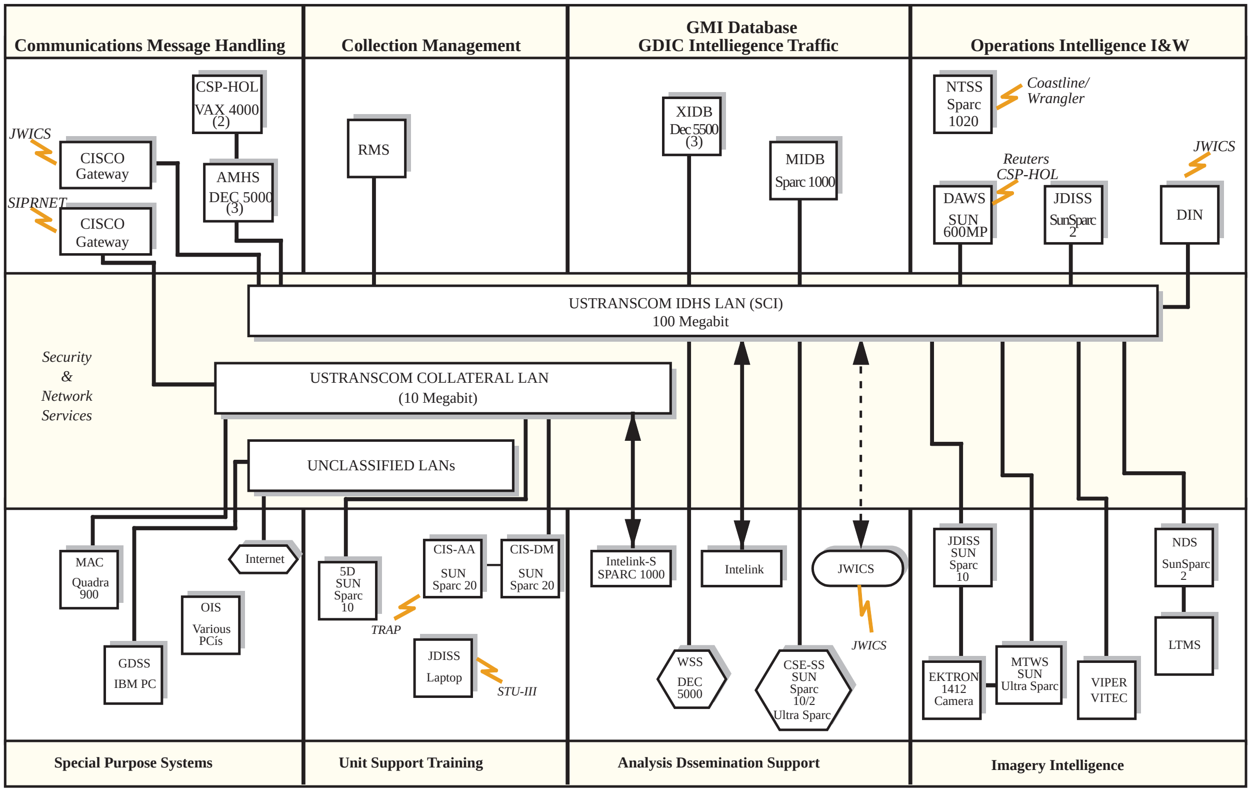


Figure 4-25d. Systems Communications Description, Intranodal Perspective (SV-2) — TRANSCOM CIAD Example

4.2.2.6 Systems² Matrix (SV-3)

Systems View	**Supporting Product**

The Systems ² Matrix is a description of the system-to-system relationships identified in the internodal and intranodal perspectives of the System Interface Description. The Systems ² Matrix is similar to an "N ²"-type matrix where the systems are listed in the rows and the columns of the matrix, and each cell represents a system pair intersection, if one exists.

There are many types of information that can be presented using a Systems ² Matrix. The system-to-system interfaces can be represented using different symbols and/or color coding that depicts different interface characteristics, for example:

- status (e.g., existing, planned, potential, de-activated)
- category (e.g., C2, intelligence, logistics)
- classification level (e.g., Secret, TS/SCI)
- means (e.g., JWICS, SIPRNet)

The Systems^2 Matrix can be organized in a number of ways (e.g., by domain, by operational phase) to emphasize the association of groups of system pairs in context with the architecture's purpose. The Systems^2 Matrix can be a useful tool for managing the evolution of systems and system infrastructures, the insertion of new technologies/capabilities, and the redistribution of systems and processes in context with evolving operational requirements.

Figure 4-26a provides a notional example of the Systems^2 Matrix.

	GCCS	MCS/P	FBCB2	M1A2 SEP	M2A3	ASAS	CGS	GBCS	IMETS	IREMBAS
GCCS		●								
MCS/P	●		●	●	●	●				
FBCB2		●		●	●	●				
M1A2 SEP		●	●		●	●				
M2A3		●	●	●		●				
ASAS	●	●	●	●	●		●	●	●	●
CGS						●				
GBCS						●				
IMETS						●				
IREMBAS						●				
AFATDS	●	●	●	●	●	●				
BFIST			●	●	●					
Paladin										
FAAVS										
MLRS										
FAADC3I	●	●	●	●	●	●				
Avenger										
BSFV-E										
GBS										
CSSCS		●	●			●				
SAMS										
SAAS										
SPDS-R										
DAMMSR										
ULLS										
....										
....										

Figure 4-26a. Systems ^{2} Matrix (SV-3) — Army First Digital Division Notional Example

Figures 4-26b and 4-26c present actual examples of the Systems² Matrix.

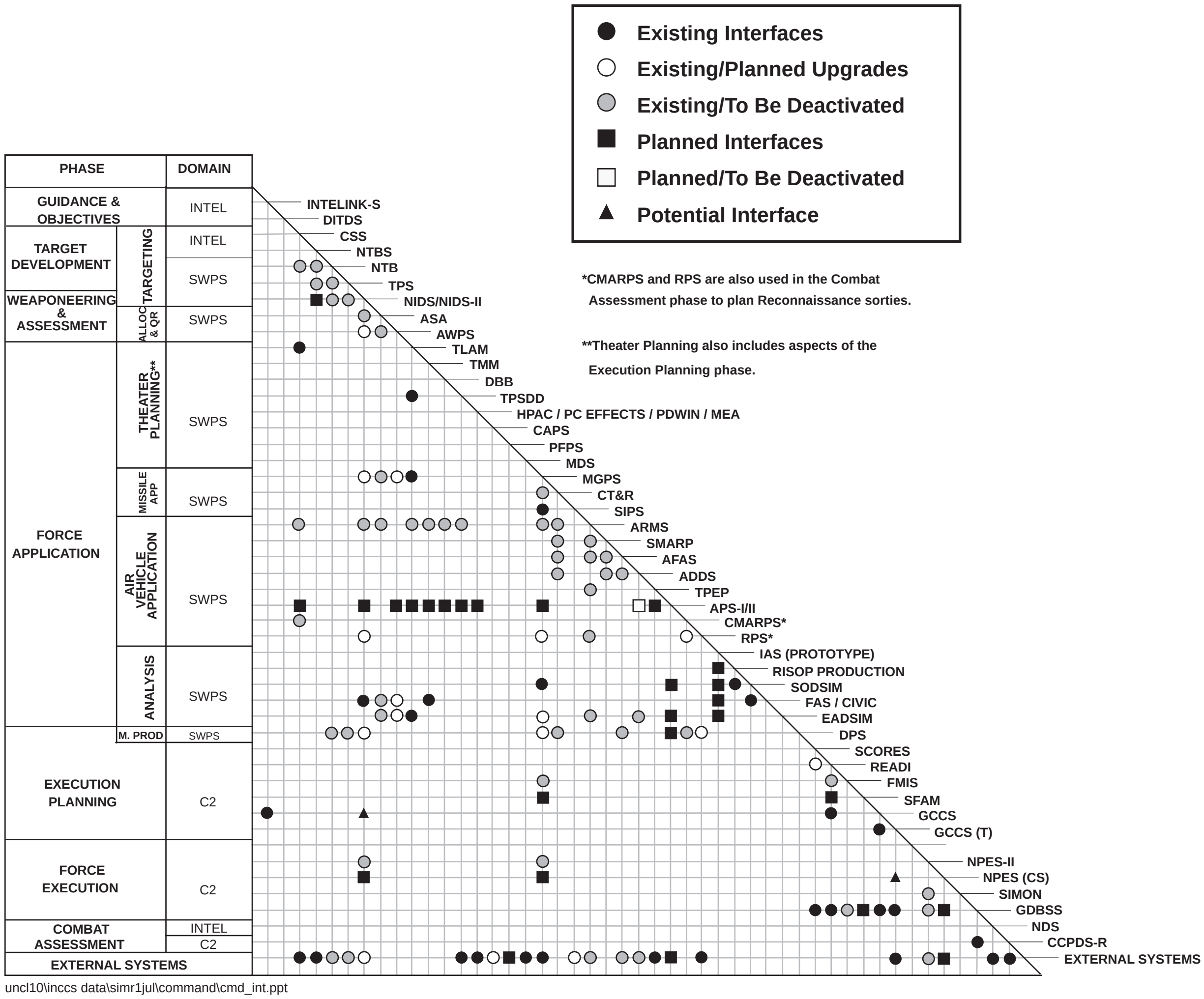


Figure 4-26b. Systems² Matrix (SV-3) — USSTRATCOM Functional Interfaces Example

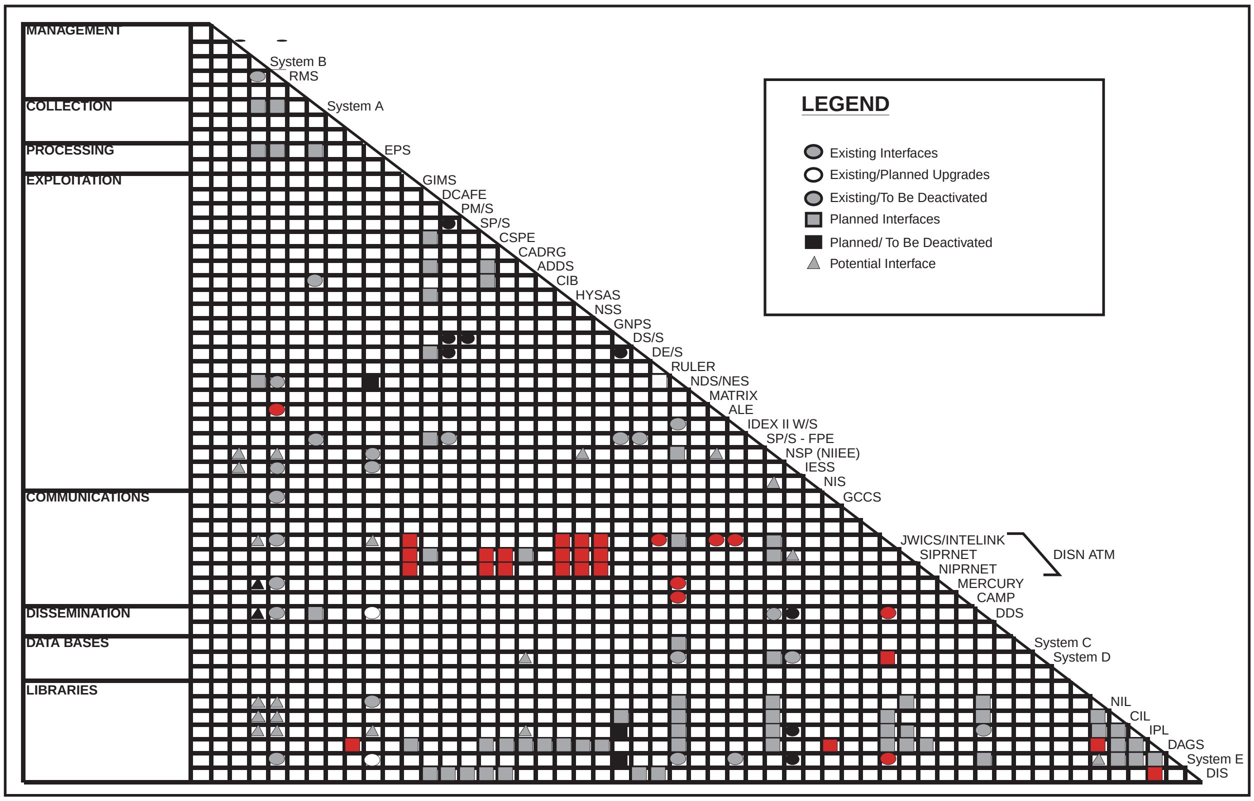


Figure 4-26c. Systems² Matrix (SV-3) —
U.S. Imagery and Geospatial System Interoperability Profile Example

4.2.2.7 Systems Functionality Description (SV-4)

Systems View	Supporting Product

The Systems Functionality Description is based on the notion of data flow diagrams. The product focuses on describing the flow of data among system functions, and on the relationships between systems or system functions and activities at nodes. Some analysts may use this product to depict the allocation of system functions to specific nodes using overlays and/or annotations, although this level of description will not always be needed for the purposes of the architecture effort. Additional foci for some versions of the description include intranode and internode data flow (i.e., within and across nodes), as well as data flow without node considerations.

Figure 4-27a shows a Systems Functionality Description template for functional decomposition.

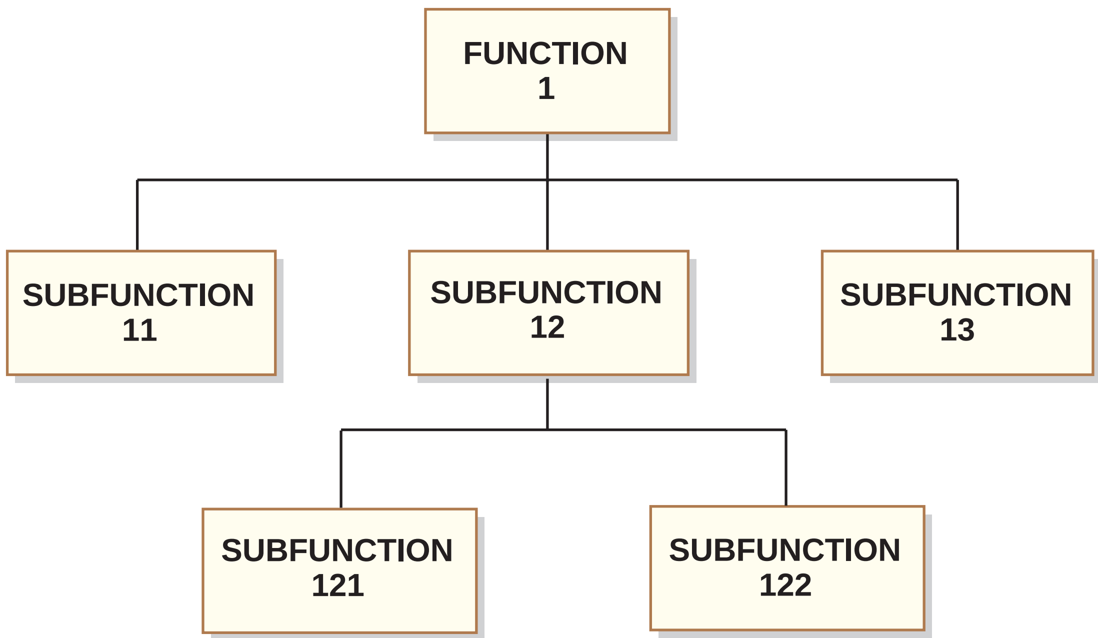


Figure 4-27a. Systems Functionality Description (SV-4) — Template (Functional Decomposition)

Figure 4-27b shows a Systems Functionality Description template for functional data flows.

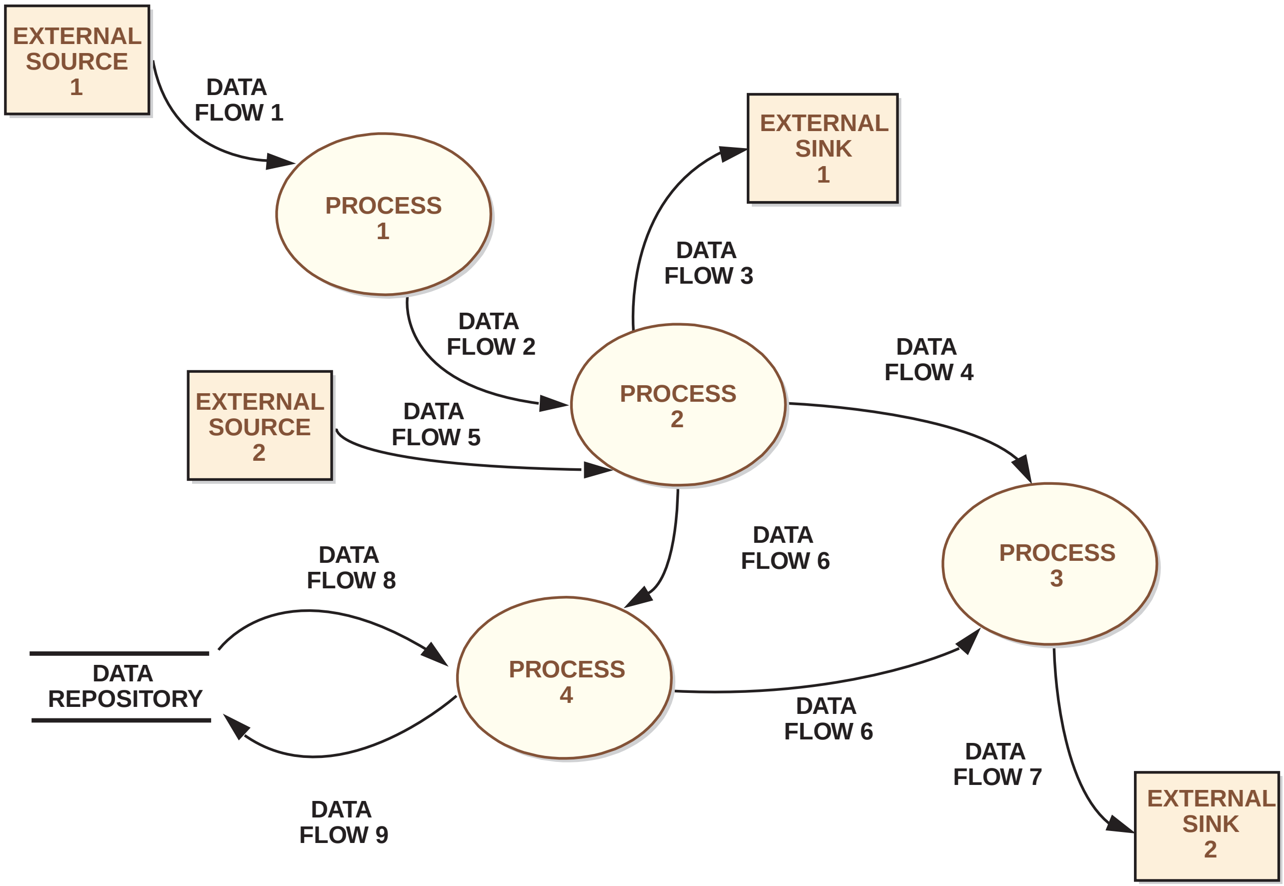


Figure 4-27b. Systems Functionality Description (SV-4) — Template (Data Flows)

Figure 4-28 provides an example.

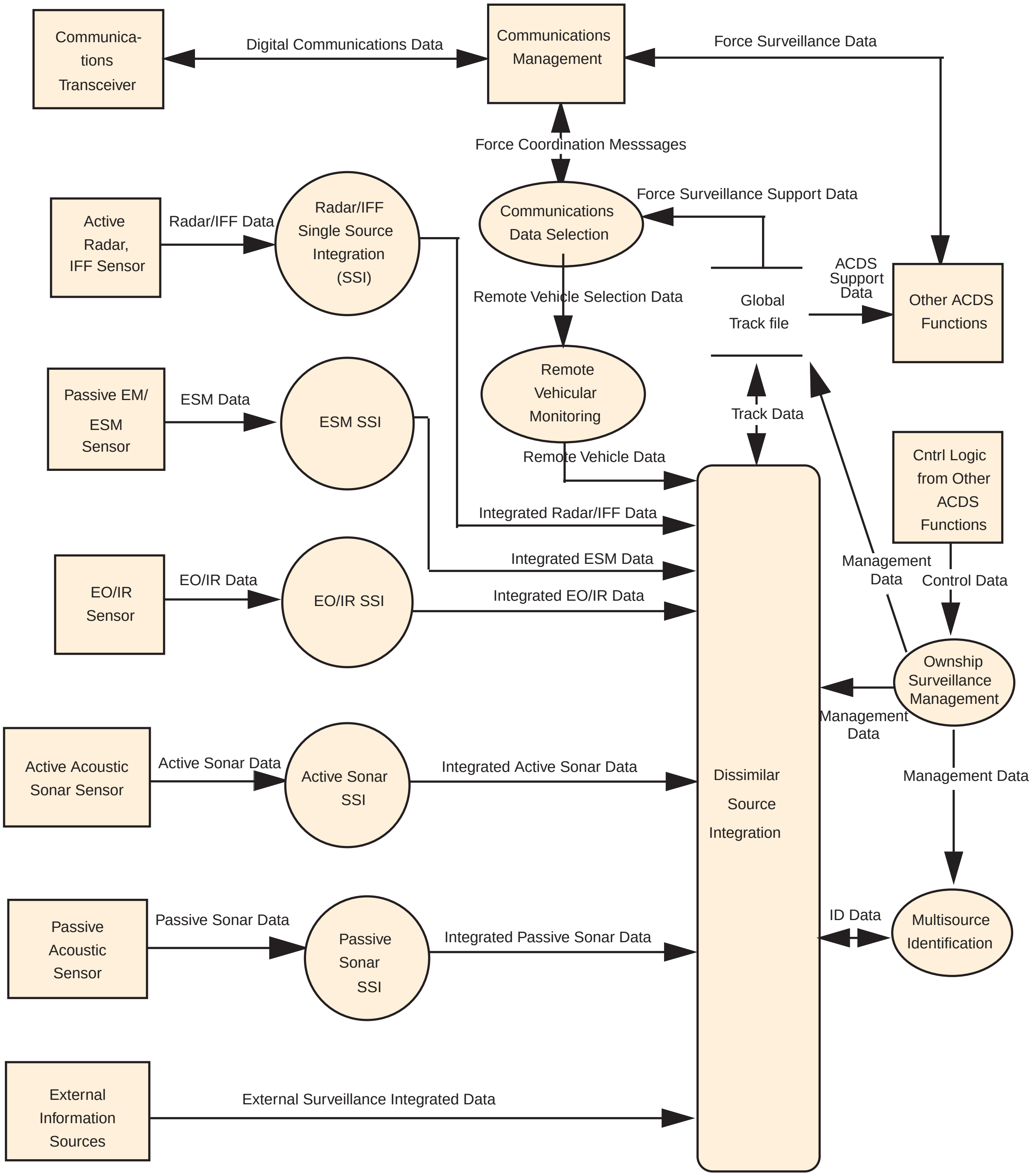


Figure 4-28. Systems Functionality Description (SV-4) —
Naval Sensor Functional Flow Diagram Example

4.2.2.8 Operational Activity to System Function Traceability Matrix (SV-5)

Systems View	Supporting Product

The Operational Activity to System Function Traceability Matrix provides a link between the operational and systems architecture views. The matrix depicts the mapping of operational activities to system functions, and thus in essence identifies the transformation of an operational need into a purposeful action performed by a system component. The systems functions associated with materiel items (i.e., processing hardware, software, or data) and mapped to an activity identify automated activities. On the other hand, activities mapped to systems functions associated with the human component of the system(s) constitute the manually-oriented activities. Depending on the purpose of the architecture, the Operational Activity to Systems function Traceability Matrix can have automated and/or manual systems functions identified and mapped to the operational activities.

The relationship between operational activities and systems functions can be expected to be "many-to-many;" that is, one activity may be supported by multiple system functions, and one system function normally supports multiple activities. Figure 4-29 provides a notional example.

Operational Activities

System Functions	3.11 Establish Mission Objective (Military planners use the term ìobjectiveî in two different senses)	3.11.3 Establish Missions Priorities	3.12 Identify Considerations Affecting Possible Courses of Action;	3.12.1 Operating Area	3.12.2 Relative Strengths and Weaknesses of Opposing Forces	3.12.3 Adequacy of Own Force	3.13 Develop Tentative/Alternative Courses of Action	3.14 Compare and Test Courses of Action	3.14.1 Consider Predicted Probable Outcomes	3.14.2 Consider Logistics and Communications	3.14.3 List/Consider Advantages and Disadvantages	3.14.4 Identify Actions to Overcome Disadvantages	3.15 Select Prospective Course of Action	3.16 Select Contingency Course of Action	3.17 Generate/Support Development of Plans and Updates	3.17.1 Mission Objectives and Tasks
Warfare Functions																
1 Sense																
1.1 Sensors																
1.1.1 ELINT Sensor																
1.1.1.1 Passive Search & Detection																
1.1.1.2 Parametric Reporting																
1.1.1.3 Test & Training Sensor Simulation																
1.1.2 COMINT Sensor																
1.1.2.1 Search & Tracking																
1.1.2.2 Signal Processing																
1.1.2.3 Test & Training Sensor Simulation																
1.1.3 Acoustic Sensor Active/Passive Search (Hull Mounted, Towed																
1.1.3.1 Arrays, Sonobuoys, Fathometer, VDS)																
1.1.3.2 Acoustic Signature Reporting																
1.1.3.3 Signal Processing																
1.1.3.4 Test & Training Sensor Simulation																

Figure 4-29. Operational Activity to System Function Traceability Matrix (SV-5) — Notional Example

4.2.2.9 System Information Exchange Matrix (SV-6)

Systems View	Supporting Product

The System Information Exchange Matrix describes, in tabular format, information exchanges between systems within a node and from those systems to systems at other nodes. The focus of the System Information Exchange Matrix, however, is on how the data exchanges actually are (or will be) implemented, in system-specific details covering such characteristics as specific protocols, and data or media formats. These aspects of exchanges are critical to understanding the potential for overhead and constraints introduced by the physical aspects of the implementation.

The nature of the System IER Description lends itself to being described as a matrix, as in figure 4-30. However, the number of information exchanges associated with an architecture may be quite large. Also, in order to understand the nature of the information exchanges, the developers and users of the architecture may want to see the IER data sorted in multiple ways, such as by source system, by media, or by destination system. Consequently, using a matrix to present that information is limiting and frequently not practical. Due to its highly structured format, the System Information Exchange Requirements Description lends itself readily to a spreadsheet or relational data base. In practice, hardcopy versions of this product should be limited to high-level summaries or highlighted subsets of particular interest.

Figure 4-30 shows a template for this product.

	Inputs						System Functions	Outputs					
	Source System or System Element	Content	Media	Data/Media Format	Security Level	Frequency, Timeliness Throughput		Destination System or System Element	Content	Media	Data/Media Format	Security Level	Frequency, Timeliness Throughput
System or System Element 1													
S/W App/Svc 1													
• • •													
S/W App/Svc *n*													
System or System Element 2													
S/W App/Svc 1													
• • •													

Figure 4-30. System Information Exchange Matrix (SV-6) — Representative Format

4.2.2.10 System Performance Parameters Matrix (SV-7)

Systems View	Supporting Product

The System Performance Parameters Matrix builds on the System Element Interface Description to depict the current performance characteristics of each system, and the expected or required performance characteristics at specified times in the future. Characteristics are listed separately for the hardware elements and the software elements. The future performance expectations are geared to the Standards Technology Forecast of the technical architecture view. Figure 4-31 is a notional example of a System Performance Parameters Matrix, listing representative performance characteristics. (Note that the term "platform" is used here to indicate a combination of hardware and operating system software.)

System Name	Performance Thresholds/Measures		
	Time $_0$ (Baseline)	Time $_1$	Time $_n$ (Objective)
Hardware Element 1			
MTBF/MTTR			
Maintainability			
Availability			
System Initialization Time			
Data Transfer Rate			
Program Restart Time			
S/W Element 1 / H/W Element 1			
Data Capacity (e.g., throughput or # of input types)			
Automatic Processing Responses (by input type, # processed/unit time)			
Operator Interaction Response Times (by type)			
Effectiveness			
Availability			
Mean Time Between S/W Failures			
Organic Training			
S/W Element 2 / H/W Element 1			
Hardware Element 2			

Figure 4-31. System Performance Parameters Matrix(SV-7) — Notional Example

4.2.2.11 System Evolution Description (SV-8)

Systems View	Supporting Product

The System Evolution Description describes plans for "modernizing" a system or suite of systems over time. Such efforts typically involve the characteristics of *evolution* (spreading in scope while increasing functionality and flexibility), or *migration* (incrementally creating a more streamlined, efficient, smaller and cheaper suite), and will often combine the two thrusts. This product builds on the previous diagrams and analyses in that information requirements, performance parameters, and technology forecasts must be accommodated. Two examples of the System Evolution Description are below in figures 4-32a and 4-32b.

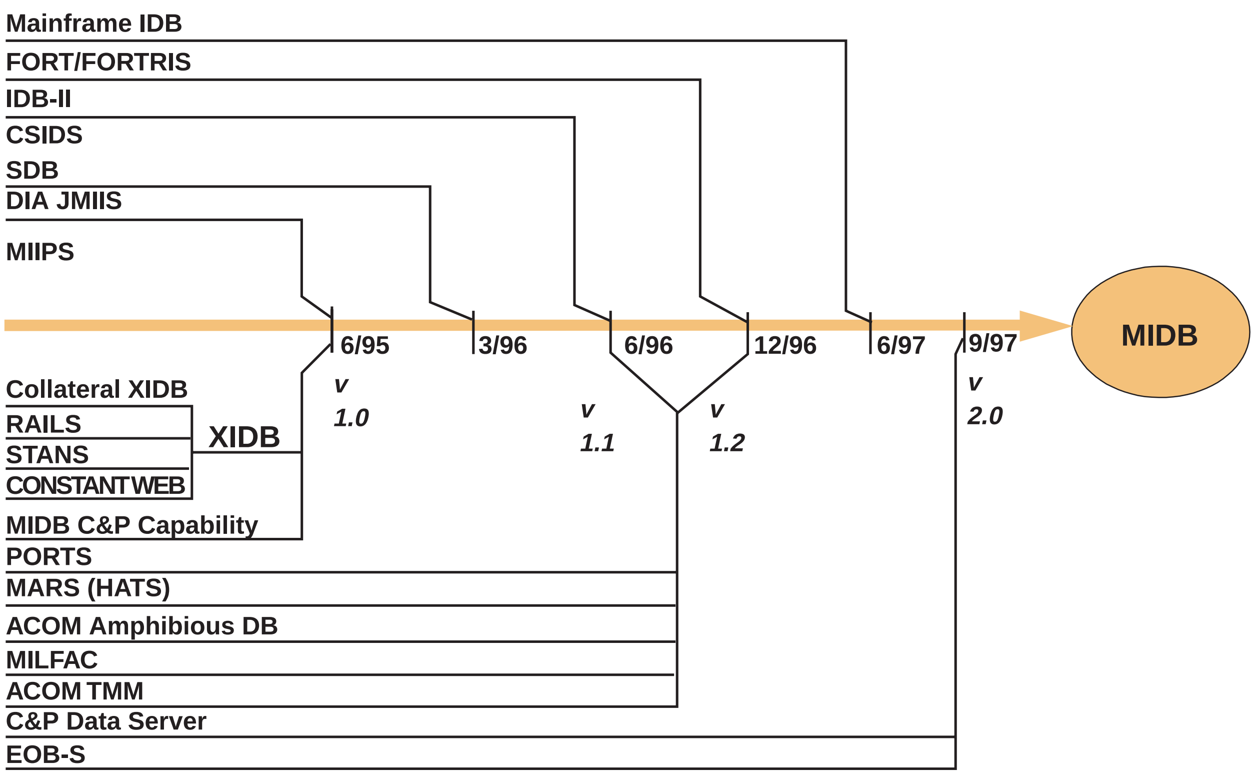


Figure 4-32a. System Evolution Diagram (SV-8) — Migration Example

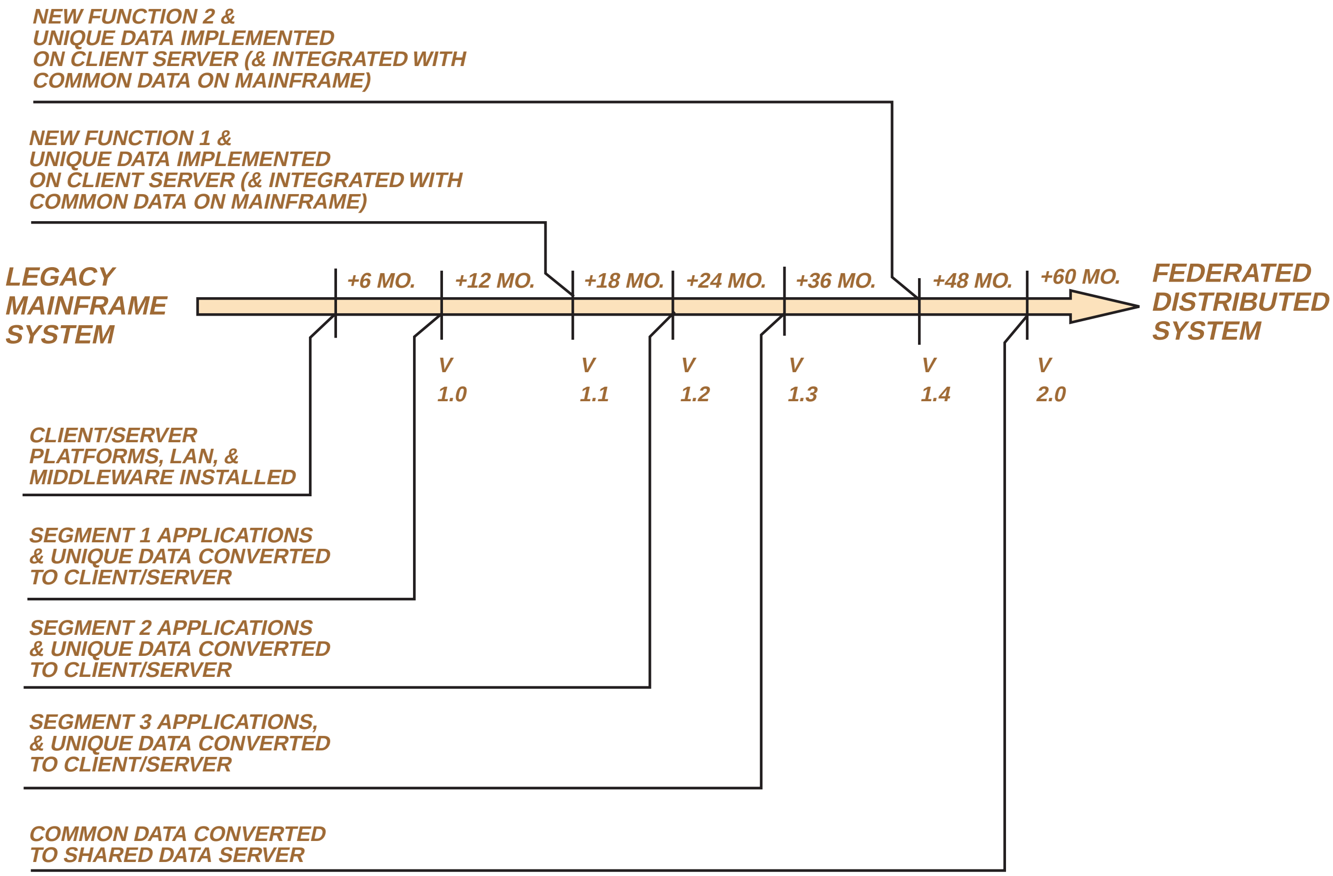


Figure 4-32b. System Evolution Diagram (SV-8) — Evolution Example

4.2.2.12 System Technology Forecast (SV-9)

Systems View	Supporting Product

A System Technology Forecast is a detailed description of emerging technologies and specific hardware and software products. It contains predictions about the availability of emerging capabilities and about industry trends in specific timeframes (e.g., 6-month, 12-month, 18-month intervals), and confidence factors for the predictions. The forecast includes potential technology impacts on current architectures, and thus influences the development of transition and objective architectures. The forecast should be tailored to focus on technology areas that are related to the purpose for which a given architecture is being built, and should identify issues that will affect the architecture. Figure 4-33 provides an example of a System Technology Forecast focused on the area of data production and management.

Technology Domain: Data Production and Management

Forecast

Technology Areas & Capabilities	*Short Term* 0 - 6 Months	*Mid Term* 6 - 18 Months	*Long Term* 18+ Months
Forecast of Industry Developments			
Distributed Heterogeneous Databases	• Middleware and/or proprietary interfaces • CGI-BIN connections to Web	• KQM • Development of APIs for Web access • Dynamic updates using Java	
Security	• Limited RSA & significant OS Level	• COTS	• *Fortezza &* RS
Hyperlink Management	• Limited Tools	• Wider availability of better tools	• Intelligent Agents
Document Creation Tools	• SGML, HTML, VRML - WWW	• SGM , HTML, VRML	• SGML, Java, VRML
Formats	• GIF, JPEG, PDF, Java (Netscape)		• Universal w/ NITF
Data Management	• Middleware Dependent		• Transparent to User
Throttling Capability		• Firewalls	
Data Replication		• Network Mirroring	

Figure 4-33. System Technology Forecast (SV-9) —
Data Production and Management Example

4.2.2.13 System Activity Sequence and Timing Descriptions (SV-10a, 10b, and 10c)

Systems View	Supporting Product

Many of the critical characteristics of an architecture are only discovered when an architecture's dynamic behaviors are defined and described. The dynamic behavior referred to here concerns the timing and sequencing of events that capture system performance characteristics of an executing system. Three types of models are needed to refine and extend the systems view of an architecture to adequately describe the dynamic behavior and performance characteristics of an architecture. These three models are:

- Systems Rules Model (SV-10a)
- Systems State Transition Description (SV-10b)
- Systems Event/Trace Description (SV-10c)

The Systems State Transition Description and Systems Event/Trace Description may be used separately or together, as necessary to describe critical timing and sequencing behavior in the systems architecture view. Both types of diagrams are used by a wide variety of different systems methodologies.

Both Systems State Transition Descriptions and Systems Event/Trace Descriptions describe systems responses to sequences of events. Events may also be referred to as inputs, transactions, or triggers. When an event occurs, the action to be taken may be subject to a rule or set of rules as described in the Systems Rule Model.

4.2.2.13.1 Systems Activity Sequence and Timing Descriptions — Systems Rules Model (SV-10a)

Systems View	Supporting Product

Rules are statements that define or constrain some aspect of the enterprise. The Systems Rules Model focuses on constraints imposed on business processes or systems functionality due to some aspect of systems design or implementation. Rules can be grouped into the following categories:

- <u>Structural Assertion</u>: Concerns (business domain) terms and facts that are usually captured by the entities and relationships of entity-relationship models; these reflect static aspects of business rules already captured in the Physical Data Model.

  - Terms: Entities
  - Facts: Association between two or more terms (i.e., relationship)

- Action Assertion: Concerns some dynamic aspect of the business or system functioning and specifies constraints on the results that actions produce.

  - Condition: Guard or "if" portion of "if-then" statement; if the condition is true, it may signal enforcing or testing of additional action assertions

  - Integrity Constraint: Must always be true (e.g., a declarative statement)

  - Authorization: Restricts certain actions to certain roles or users

- Derivation: Concerns algorithm used to compute a derivable fact from other terms, facts, derivations, or action assertions.

Since the Structural Assertion rules are captured in the Physical Data Model, the Systems Rules Model can focus on the more dynamic Action Assertions and Derivations rules. Additional characteristics of rules include the following:

- Independent of the modeling paradigm used

- Declarative (non-procedural)

- Atomic (indivisible yet inclusive)

- Expressed in a formal language such as:

  - Decision trees and tables
  - Structured English
  - Mathematical logic

- Distinct, independent constructs

- Business-oriented

Each group may select the formal language in which to record its Systems Rules Model, as long as the notation selected is referenced and well-documented.

Figure 4-34 illustrates an example Action Assertion that might be part of a Systems Rules Model. The assertion is an example of one that might be necessary mid-way through a system migration, when the databases that support three Forms (FORM-X, FORM-Y, and FORM-Z) have not yet been integrated, so explicit user or application action is needed to keep related data synchronized. The example is given in a form of structured English.

If field A *in* FORM-X *is set to* value T,
Then field B *in* FORM-Y *must be set to* value T
And field C *in* FORM-Z *must be set to* value T
End If

Figure 4-34 System Rules Model (SV-10a) — Action Assertion Example

4.2.2.13.2 Systems Activity Sequence and Timing Descriptions — Systems State Transition Description (SV-10b)

Systems View	**Supporting Product**

A state specifies the response of a system to events. The response may vary depending on the current state and the rule set or conditions. The Systems State Transition Description relates events and states. When an event occurs, the next state depends on the current state as well as the event. A change of state is called a transition. Actions may be associated with a given state or with the transition between states. The Systems State Transition Description is used to relate events and states at the systems level, such as describing the detailed sequencing of functions in a system. This explicit time sequencing of systems activities in response to external and internal events is not fully expressed in the Systems Functionality Description.

Figure 4-35 provides a template for a simple Systems State Transition Description. Initial states (usually one per diagram) are pointed to by the black dot and incoming arrow while terminal states are identified by an outgoing arrow pointing to a black dot with a circle around it. States are indicated by rounded corner box icons and labeled by name or number and, optionally, any actions associated with that state. Transitions between states are indicated by directed lines (i.e., one way arrows) labeled with the event that causes the transition and the action associated with the transition.

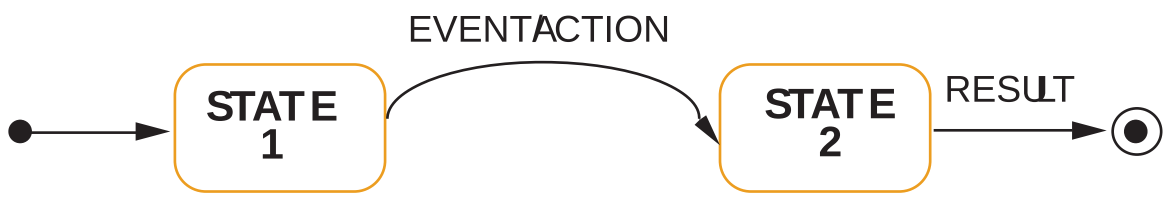


Figure 4-35. System State Transition Description (SV-10b) — High-Level Template

Figures 4-35a through 4-35c provide templates for layered structures that can be used to build up a more complex type of State Transition Diagram known as a Harel State Chart. There are a variety logically equivalent forms of State Transition Diagram, but the Harel State Chart is the easiest to use for describing potentially complex, real-world situations, since it allows the diagram to be decomposed in layers showing increasing amounts of detail. Figures 4-35a and 4-35b provide templates for layered states while figure 4-35c provides a template for a complex transition involving synchronized activities.

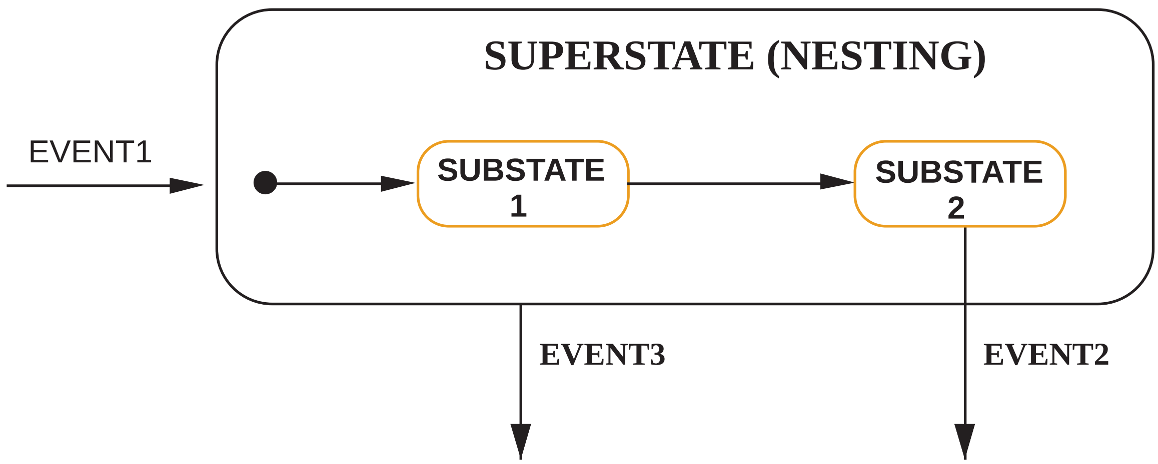


Figure 4-35a. System State Transition Description (SV-10b) —
Nested State Structure Template

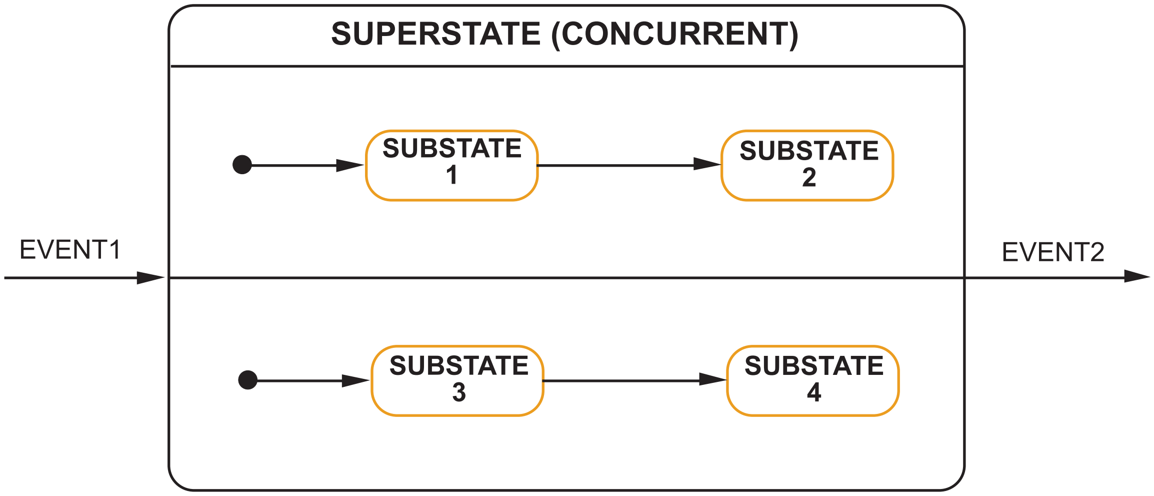


Figure 4-35b. System State Transition Description (SV-10b) —
Concurrent Activity State Structure Template

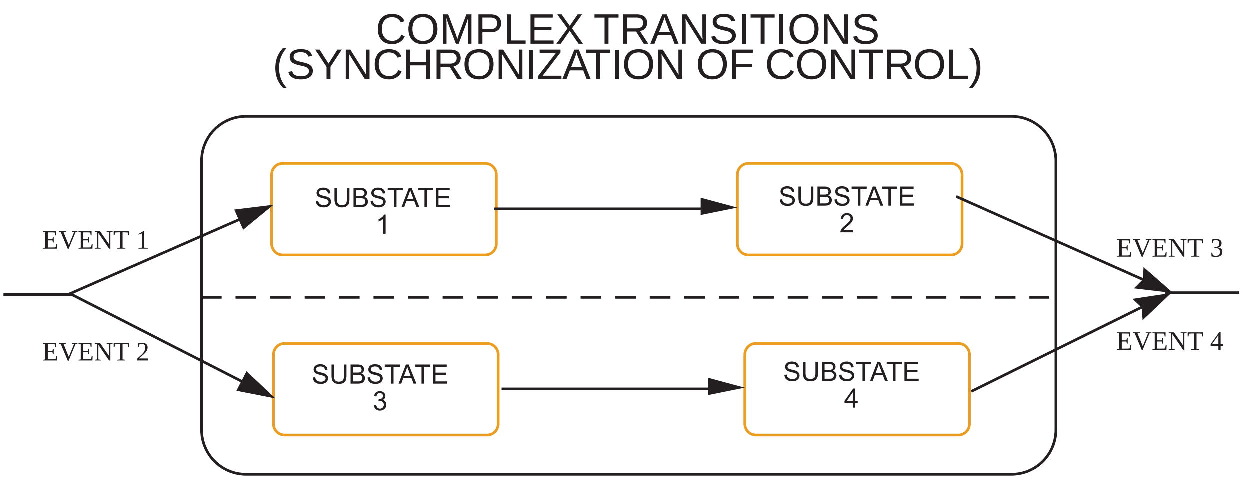


Figure 4-35c. System State Transition Description (SV-10b) —
Complex Transition Template

Figure 4-36 illustrates a Harel State Chart for a telephone. This example models the behavior of a telephone as a closed loop activity and thus does not show any initial or terminal states at the top level. There are a variety of other, logically equivalent forms of State Transition Diagram, although the Harel State Chart is the easiest to use for describing potentially complex, real-world situations.

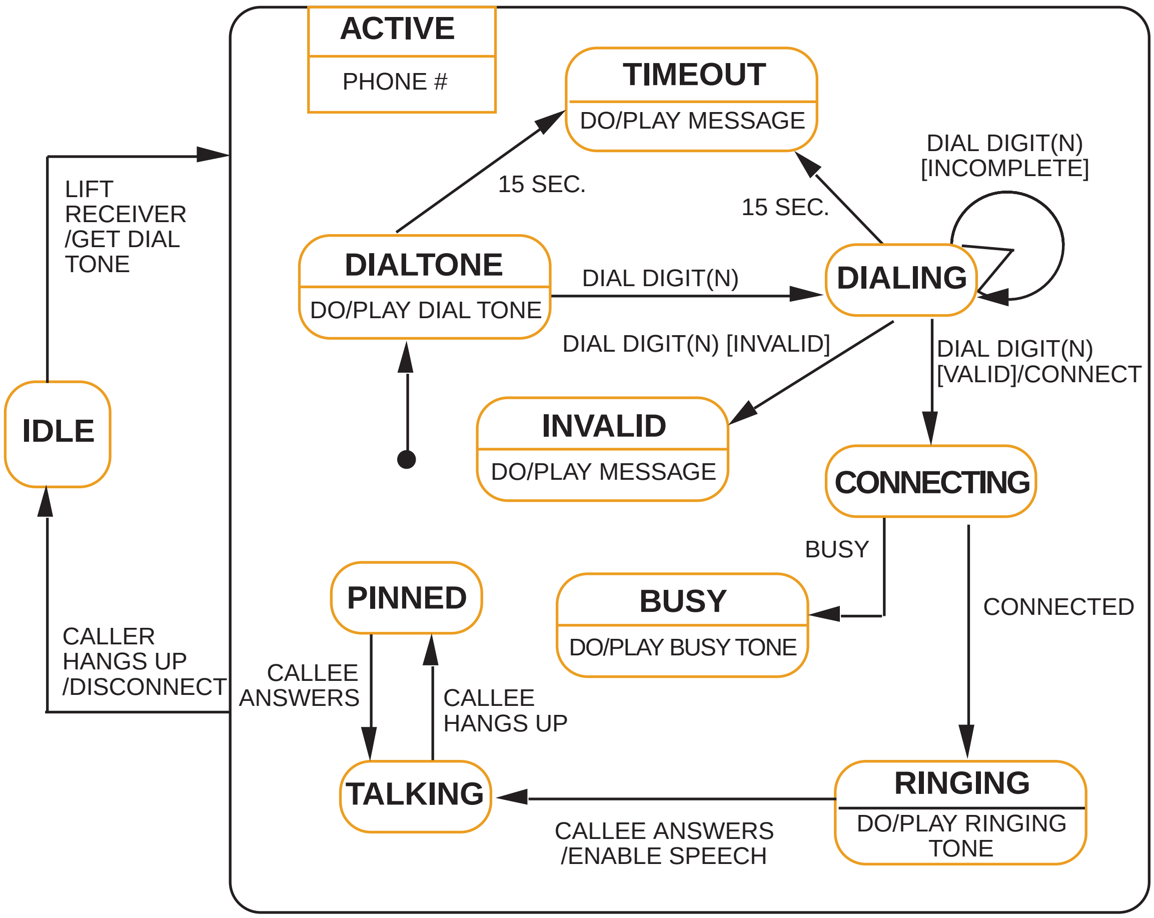


Figure 4-36. Systems State Transition Description (SV-10b) — Telephone Example

4.2.2.13.3 Systems Activity Sequence and Timing Descriptions — Systems Event/Trace Description (SV-10c)

Systems View	**Supporting Product**

Systems Event/Trace Descriptions, sometimes called Sequence Diagrams, Event Scenarios, and Timing Diagrams, allow the tracing of actions in a scenario or critical sequence of events. Systems Event/Trace Descriptions can be used by themselves or in conjunction with Systems State Transition

Descriptions to describe dynamic behavior. The Systems Event/Trace Descriptions in the systems architecture view may reflect system-specific aspects or refinements of critical sequences of events described in the operational architecture view.

Figure 4-37 provides a template for a Systems Event/Trace Description. The items across the top of the diagram are nodes, usually operational facilities where action must be taken based on certain types of events. Each node has a timeline associated with it which runs vertically. Specific points in time can be labeled running down the left hand side of the diagram. Directed lines between the node time lines represent events, and the points at which they intersect the timelines represent the times at which the nodes become aware of the events. The direction of the event lines represents the flow of control from one node to another based on the event.

Figure 4-38 provides an example of a Systems Event/Trace Description for a phone switching system. The sequence of events diagrammed represents the initiation of a call through the network. The example diagram contains formulas on the left hand side that relate the timing of certain events (e.g., that routing the call takes less than 5 seconds). (Note: This type of timing information can also be added to Systems State Transition Description, if desired.)

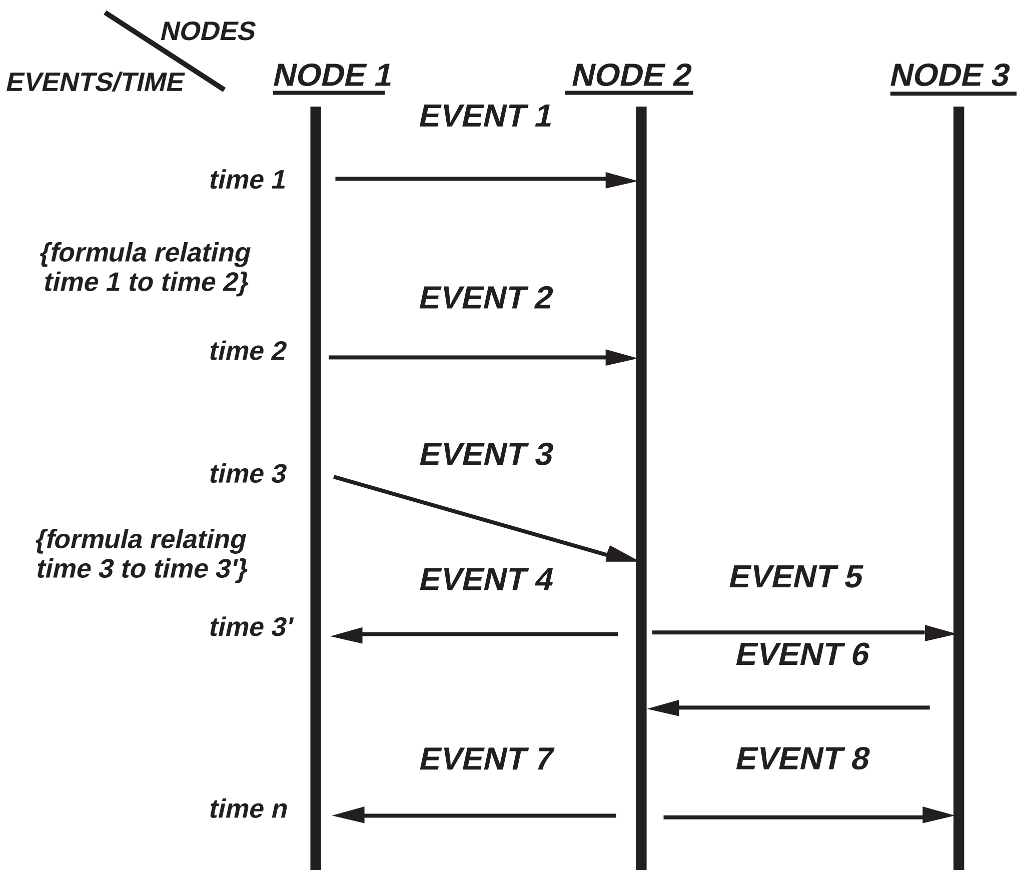


Figure 4-37. Systems Event/Trace Description (SV-10c) — Template

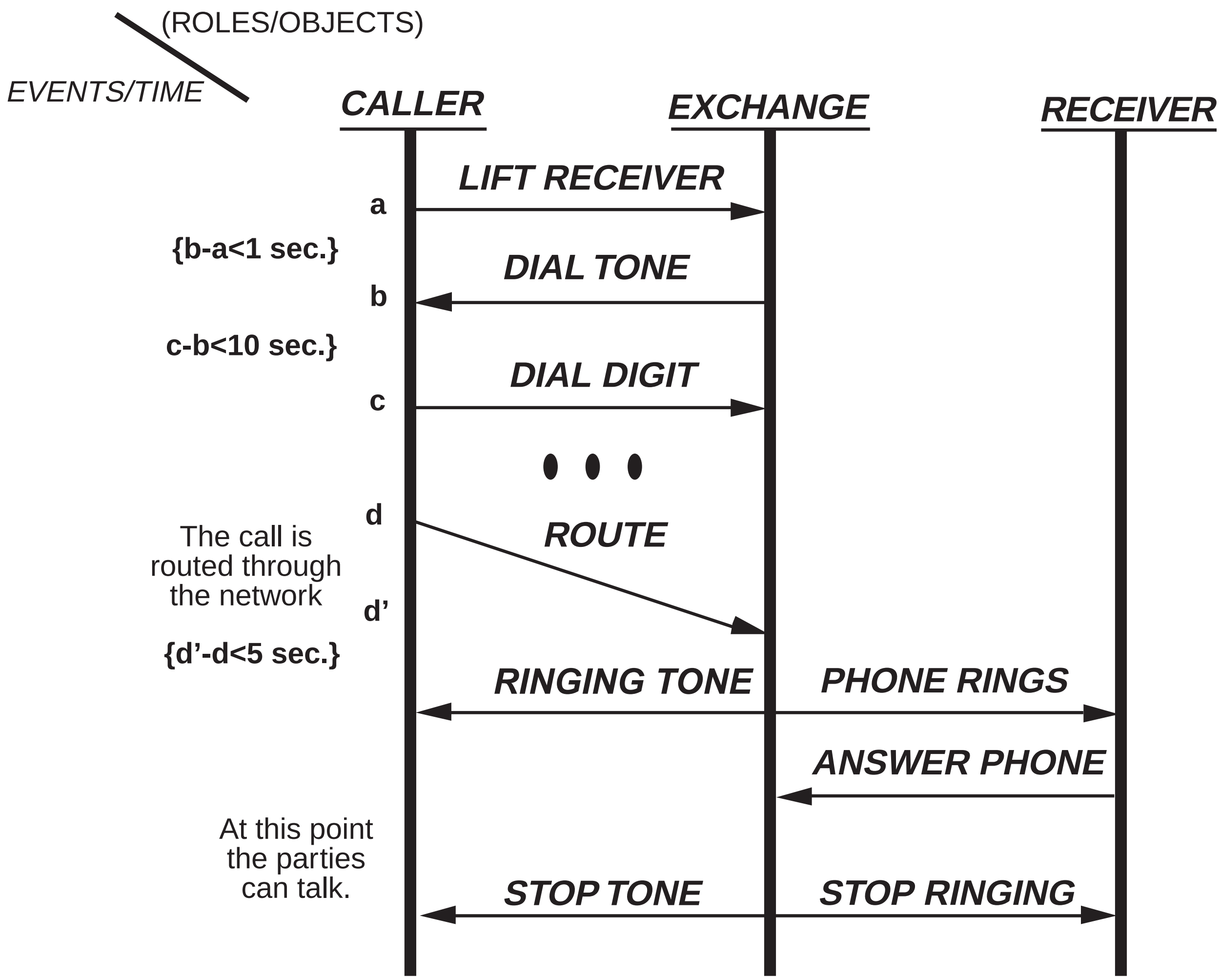


Figure 4-38. Systems Event/Trace Description (SV-10c) — Telephone Switching Example

4.2.2.14 Physical Data Model (SV-11)

Systems View	Supporting Product

The Physical Data Model (PDM) is used to describe how the information represented in the Logical Data Model is actually implemented in the systems architecture view. The Physical Data Model shows how the information-exchange requirements are actually implemented. The Physical Data Model shows how both data entities and their relationships are maintained.

There should be a mapping from a given Logical Data Model to the Physical Data Model if both models are used. The form of the Physical Data Model can vary greatly, as shown in figure 4-39. For some purposes, an additional entity-relationship style diagram will suffice. Data Definition Language may also be used in the cases where shared databases are used to integrate systems. References to message format standards (which identify message types and options to be used) may suffice for message-oriented implementations. Descriptions of file formats may be used when file passing is the mode used to exchange information. Interoperating systems may use a variety of techniques to exchange data, and thus have several distinct partitions in their Physical Data Model with each partition using a different form.

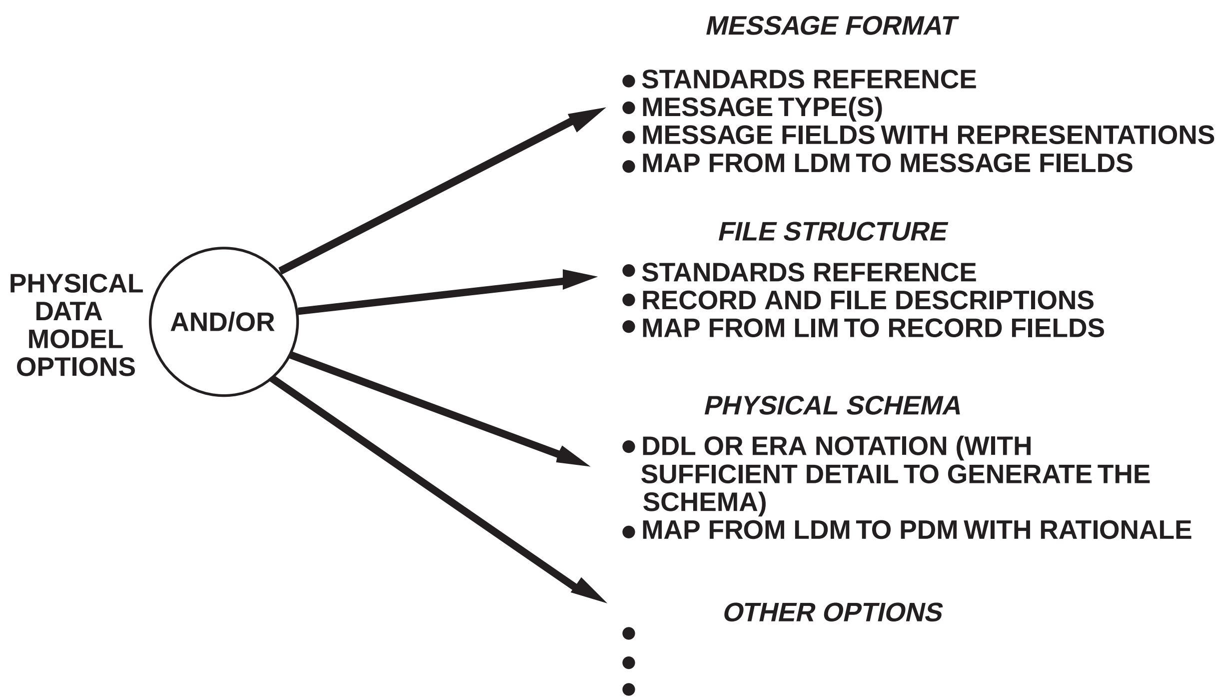


Figure 4-39. Physical Data Model (SV-11) — Representation Options

4.2.2.15 Standards Technology Forecast (TV-2)

Technical View	Supporting Product

A Standards Technology Forecast is a detailed description of emerging technology standards relevant to the systems and business processes covered by the architecture. It contains predictions about the availability of emerging standards and the likely obsolescence of existing standards in specific timeframes (e.g., 6-month, 12-month, 18-month intervals), and confidence factors for the predictions. It also contains matching predictions for market acceptance of each standard and an overall risk assessment associated with using the standard. The forecast includes potential standards impacts on current architectures, and thus influences the development of transition and objective architectures. The forecast should be tailored to focus on technology areas that are related to the purpose for which a given architecture description is being built, and should identify issues that will affect the architecture.

Figure 4-40 provides an example of a Standards Technology Forecast focused on the area of data production and management, as it might have been developed in 1993.

Service Areas	Service	Status	As of 6/93	Expected by 12/93	Expected by 12/94	Expected by 12/94	Comments
Operating System	Kernel	Now	FIPS PUB 151-1	FIPS PUB 151-2			
	Shell & Utilities	Now	IEEE 1003.2	FIPS Addition			
	Real Time Extension	Future	IEEE 1003.4	FIPS Addition			
Program-ming	Program-ming Language	Now	FIPS PUB 119 - Ada		FIPS PUB 119-1 Ada9X		
	CASE Tools & Environ-ment	Now	ECMA Spec 149 - PCTE				
User Interface	. . .						
Data Manage-ment	Data-Diction-ary/Direct -ory	Now	FIPS PUB 156 - IRDS				
	Data Manage-ment	Now	FIPS PUB 127-1-SQL		FIPS PUB 127-2-SQL+	FIPS PUB 127-3-SQL++	
. . .							

Figure 4-40. Standards Technology Forecast (TV-2) —
Data Production and Management Example (c. 1993)

4.3 UNIVERSAL REFERENCE RESOURCES

A number of reference models and information standards exist which serve as sources for guidelines and attributes that must be consulted while building architecture products. Each of these resources is defined and described in its own document (see Sources); however, some of these references are listed in table 4-2 and are briefly described here.

Table 4-2. Universal Reference Resources

Applicable Architecture Views	Universal Reference Resource	General Nature
All Views	*C4ISR Core Architecture Data Model (CADM)*	Logical data model of information used to describe and build architectures
All Views	*Defense Data Dictionary System (DDDS)*	Repository of standard data definitions, formats, usage, and structures
All Views	*Levels of Information Systems Interoperability (LISI)*	Reference Model of interoperability levels and operational, systems, and technical architecture associations
Operational	*Universal Joint Task List (UJTL)*	Hierarchical listing of the tasks that can be performed by a Joint military force
Operational	*Joint Operational Architecture (JOA)*	(In development) -- High-level, evolving architecture depicting Joint and multi-national operational relationships
System Technical	*Technical Reference Model (TRM)*	Common conceptual framework and vocabulary encompassing a representation of the information system domain
System Technical	*DII Common Operating Environment (COE)*	Framework for systems development encompassing systems architecture standards, software reuse, sharable data, interoperability and automated integration
Technical	*Shared Data Environment (SHADE)*	Strategy and mechanism for data-sharing in the context of DII COE-compliant systems
Technical	*Joint Technical Architecture (JTA)*	IT standards and guidelines

4.3.1 C4ISR Core Architecture Data Model (CADM)

All Views **Universal Reference Resource**

The *C4ISR Core Architecture Data Model (CADM)* was designed to provide a common approach for organizing and portraying the structure of architecture information. By facilitating the exchange, integration, and comparison of architecture information throughout the DoD, this common approach should help improve Joint C4ISR interoperability. (The current, initial version of the CADM focuses on C4ISR; later versions will have a broader focus.) The CADM is a logical rather than a physical data model. Thus, it provides a conceptual view of how information is organized, rather than a description of how the data is actually stored in a real database implementation. The model's design was patterned after architecture data models, refined by comparison with the information structure of architectures, and validated using Framework products.

It is important to understand that the CADM models the structure of architecture information in general, not the data of a particular C4ISR problem domain. The CADM also does not include features, such as logistics or fiscal entities, unique to the architecture processes and requirements of a particular user community or functional area. But users who require these features should be able to extend the core with little effort.

4.3.1.1 Overview of the CADM

The C4ISR Core Architecture Data Model (CADM) is designed to provide a common approach for organizing and portraying the structure of architecture information. By facilitating the exchange, integration, and comparison of architecture information throughout DoD, this common approach should help improve joint C4ISR interoperability.

The CADM was initially developed by selecting a from the most important and useful features of existing architecture data models, including the Standard Data Element-Based Automated Architecture Support Tool Environment (SAASE), the forthcoming Joint C4ISR Architecture Planning System (JCAPS), and architecture data models of the Military Services and Agencies (see figure 4-41). The resulting draft CADM was then subjected to several months of scrutiny and refinement by a panel consisting of representatives from each of the military services, as well as representatives from several key agencies. Finally, the information requirements of key Architecture Framework products were traced to the CADM to ensure that the model was sufficient and complete. In short, the model's design was patterned after architecture data models, refined by comparison with the information structure of architectures, and validated using Framework products. This development approach should make the CADM relatively stable in that it is primarily built around real world entities and relationships, since such real world objects are largely unaffected by changes in architecture processes and the Architecture Framework products that support those processes.

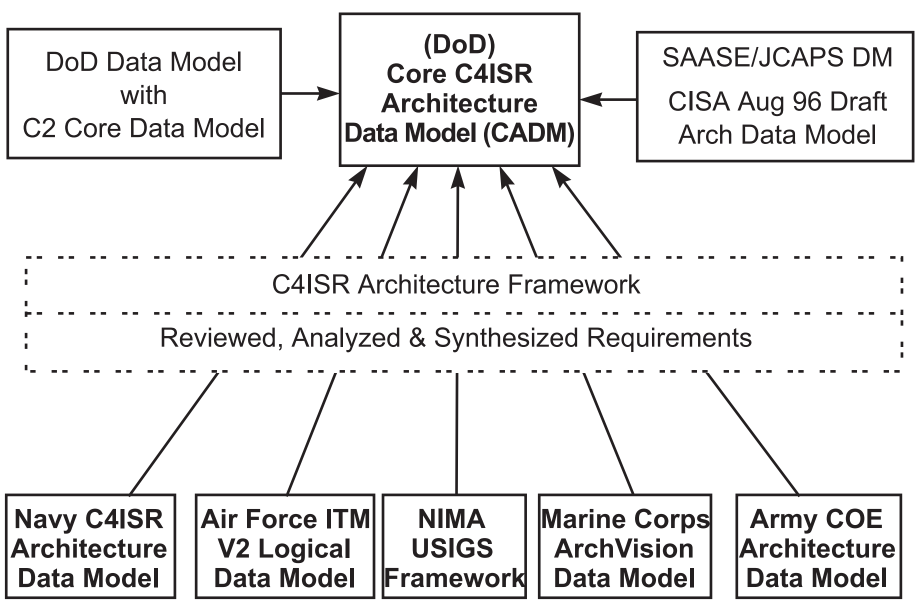


Figure 4-41. Sources for CADM Development

It is important to understand that the CADM models the structure of architecture information in general, not the data of a particular C4ISR problem domain. For example, the data model for a fire support architecture could be stored in a CADM database (as an instance of DOCUMENT or CONCEPTUAL-DATA-MODEL). The CADM itself does not include fire support entities such as "MISSILE-BATTERY" or "FORWARD-OBSERVER."[1] The CADM also does not include features, such as logistics or fiscal entities, unique to the architecture processes and requirements of a particular user community or functional area. However, users who require these features can typically extend the core with little effort. This "core model" approach offers several advantages:

- A core data model is easier to understand and maintain
- Legacy data is more easily mapped into a core data model[2]
- A core data model is more resistant to change because it contains only the most fundamental entities and relationships, and these entities and relationships are expected to be the most stable
- It is easier to gain and maintain consensus on a core data model

4.3.1.2 Model Overview

The CADM is a logical rather than a physical data model. Thus, it provides a conceptual view of how information is organized, rather than a description of how the data is actually stored in a real database implementation. Figure 4-42 is a high-level entity-relationship diagram depicting only 25 top-level entities (10 percent of the CADM entities) and none of the CADM attributes. Each entity (rectangular box) in figure 4-42 can be thought of as representing a table (a collection of like-structured records) in a traditional relational database, in which each column would provide values for an attribute. Relationships between entities are denoted with lines containing one or two bold dots (at the "many") end. For example, there is a many-to-many relationship between the high-level entities GUIDANCE and AGREEMENT—each instance of GUIDANCE corresponds to zero, one, or many instances of AGREEMENT, and each instance of AGREEMENT corresponds to zero, one, or many instances of GUIDANCE. The CADM

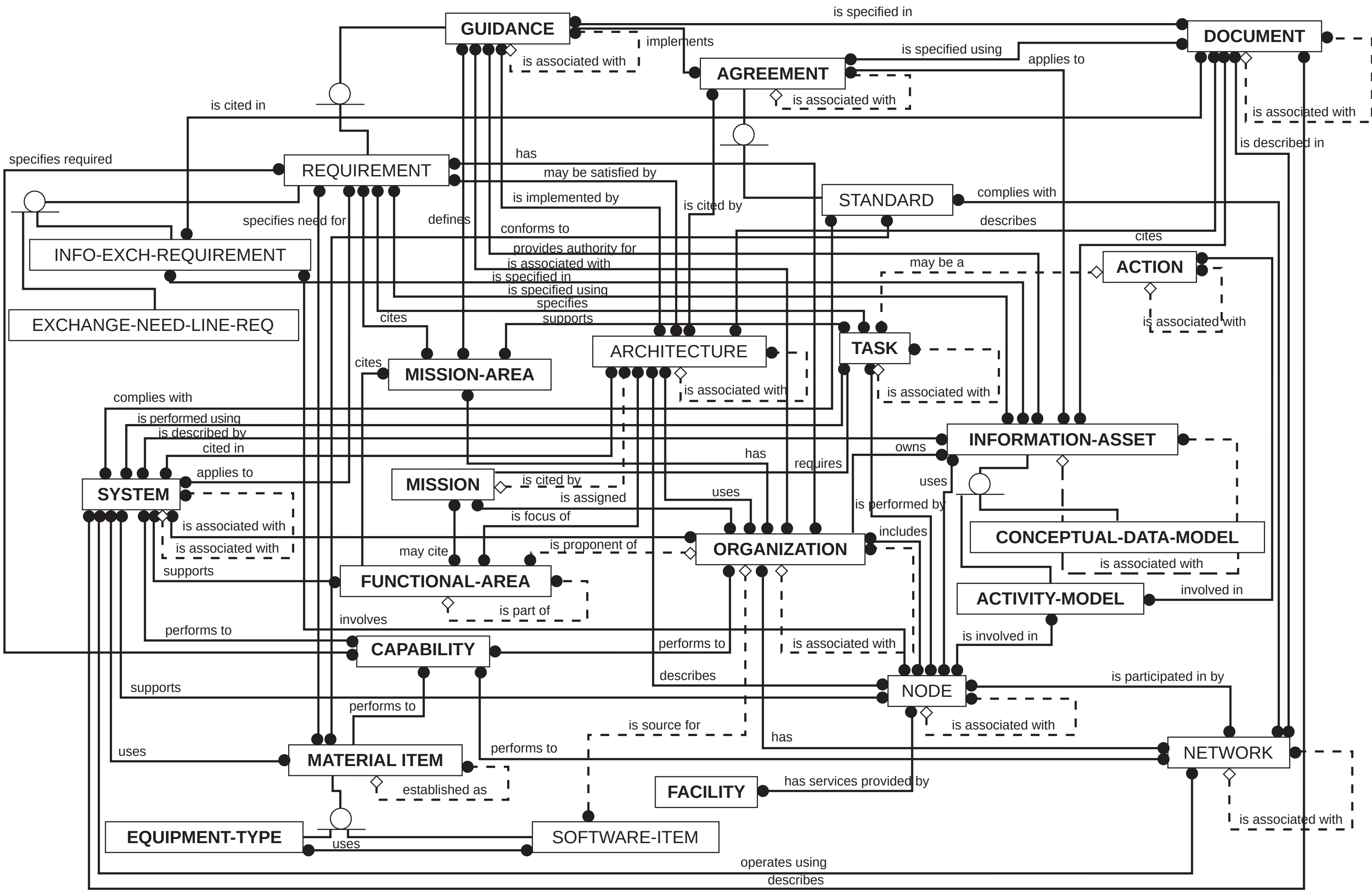
---

[1] The CADM is intended as a "core" architecture data model containing data requirements common across functional areas. This means that specifics that pertain to individual Commands, Services, or Agencies are not made part of the "core," but can be readily added to the "core" in order to satisfy those unique requirements identified by the user. Thus, although not part of the CADM, an entity such as MISSILE BATTERY could be added, since the CADM already contains the corresponding entity MATERIEL ITEM which can be viewed as the super-type of all different kinds of materiel.

[2] The reader should note that where no "core" is present to which and from which the multiple architectures can translate in order to interoperate, the number of needed pairwise "translations" scales as N^2-N, where N is the number of architectures exchanging information. However, this number would scale only as 2N if there were an agreed "core" to which and from which implementations could translate in order to share data. Even for small numbers of architectures (e.g., 50), the difference can be staggering. "Translation" is greatly simplified if the Commands, Services, and Agencies adopt the CADM as an integral part of the specification of architecture databases.

uses an “associative” entity AGREEMENT-GUIDANCE to record attributes of such relationships.

Table 4-3 lists informal definitions for the top-level entities depicted in figure 4-42. A fully attributed IDEF1X data model, a complete data dictionary, and additional CADM documents are provided in the CADM document.



NOTE: DoD standard entities are shown in bold font.

Figure 4-42. Overview of the Key Entities and Relationships for the C4ISR Core C4ISR Architecture Data Model (CADM)

Table 4-3. Descriptions of Key Entities of the CADM

	Entity	Definition and Remarks
1	ACTION	***An activity***, such as an IDEF0 activity or a war fighting task.
2	ACTIVITY-MODEL	***A representation of the interrelated functions of a system.*** Usually an IDEF0 Activity Model.
3	AGREEMENT	***An arrangement between parties***, such as an IEEE standard or memorandum of agreement.
4	ARCHITECTURE	*The structure of components, their relationships, and the principles and guidelines governing their design and evolution over time.* [IEEE STD 610.12; C4ISR Architecture Framework, June 1996] Architectures can be operational, systems, technical, organizational, functional, AS-IS, TO-BE, or any other architecture.
5	CAPABILITY	***An ability to achieve an objective***. Examples include MOEs, MOPs, and technical performance parameters.
6	CONCEPTUAL-DATA-MODEL	***A structured graphical and/or textual representation of concepts and knowledge within an activity.*** A description of how data are organized and how that organization reflects the information structure of a problem domain. Can describe complex (such as a database) or simple (such as a packet) data structures.
7	DOCUMENT	***Recorded information regardless of physical form***. Can include text, bit-mapped images, and spreadsheets. Also includes (electronic versions of) Architecture Framework products.
8	EQUIPMENT-TYPE	***A category of MATERIEL-ITEM that provides capability through repeated use.*** Includes hardware and software.
9	EXCHANGE-NEED-LINE-REQUIREMENT	*A REQUIREMENT that is the logical expression of the need to transfer information (whose content is specified by reference toINFORMATION-EXCHANGE-REQUIREMENT) among nodes (e.g., operational elements, system elements).*
10	FACILITY	***Real property, having a specified use, that is built or maintained by people***. A computing mega-center would be an example.
11	FUNCTIONAL-AREA	***A major area of related activity***, such as Ballistic Missile Defense, Logistics, or C2 support.
12	GUIDANCE	***A statement of direction***. This definition is broader (and more directive) than the definition used in some contexts. It includes doctrine, laws, and directives.
13	INFORMATION-ASSET	***An information resource***. Includes various data specifications and information models, such as activity, conceptual data, internal data, user presentation, and process models.
14	INFORMATION-EXCHANGE-REQUIREMENT	*A REQUIREMENT for the content of an information flow. Associated with an IER are such performance attributes as information size, throughput, timeliness, quality, and quantity values.* May be many-to-many in relation to EXCHANGE-NEED-LINE-REQUIREMENT.
15	MATERIEL-ITEM	***A characterization of a materiel asset.***
16	MISSION	*An objective together with the purpose of the intended action.*
17	MISSION-AREA	***The general class to which an operational mission belongs.***
18	NETWORK	*The joining of two or more components for a specific purpose.* Can be transportation, power, communications or other network.
19	NODE	*A primitive that is a component of a network.* Use is not limited to a node in a communications network. Can be combined with arcs to represent virtually any network or graph structure. Topologically, a NODE is zero dimensional. In the Framework, a representation of an element of architecture that produces, consumes, or processes data.
20	ORGANIZATION	***An administrative structure with a mission***. Organization is used here in a very broad sense. Includes military organizations, agencies, units, OPFACs, and even governments.
21	REQUIREMENT	***A need or demand***. A subtype of guidance. May be specified in other guidance or derived from necessity and circumstances.
22	SOFTWARE-ITEM	*A set of instructions that govern the operation of data processing equipment.* Includes firmware, software applications, operating systems, and embedded software.
23	STANDARD	*An agreement for a procedure, product, or relationship.*
24	SYSTEM	*A collection of components organized to accomplish a specific function or set of functions.* May itself be composed of systems.
25	TASK	*A discrete unit of work, not specific to a single organization, weapon system, or individual, that enables missions or functions to be accomplished.* May be explicitly or implicitly directed, as by doctrine or demands of the situation.

Note: Italic font identifies the formal definition used in the CADM. Bold font identifies approved DoD data standard definitions.

Several aspects of the CADM entity-relationship diagram are worth noting:

- Architecture information for a Framework product can often be specified using several different structures of the CADM. For example, a data model may be described in a specific DOCUMENT, while its technical composition is captured as a CONCEPTUAL-DATA-MODEL (a subtype of INFORMATION-ASSET). In the latter case, the data model is actually decomposed into its component parts (DATA-ENTITY, DATA-ATTRIBUTE, and DATA-ENTITY-RELATIONSHIP) and these parts are associated with a parent instance of CONCEPTUAL-DATA-MODEL. This allows the user to perform sophisticated queries not only on portions of the explanatory DOCUMENT but also on the technical details of the data model.

- Much of the data in a CADM database would be associated with specific ARCHITECTURE instances. For example, a particular SYSTEM might be associated with the "USCENTCOM AS-IS Theater Missile Defense Systems Architecture." This allows a single data base to simultaneously hold multiple architectures, usually distinguished by parent organization, supported function, or applicable time frame.

- Many entities are related to themselves in several ways. This is indicated in figure 4-42 by a dashed line from an entity back to the same entity. For example, a NODE might be "composed of" or "linked to" one or more other nodes. Similarly, SYSTEMs and ORGANIZATIONs might be "part of" other SYSTEMs and ORGANIZATIONs. In these cases, the dashed lines in figure 4-42 actually represent associative entities with attributes that specify the nature of the relationship (e.g., "part of" or "is linked to").

- Subtyping has been used to reduce the model's complexity and make it more resistant to change. For example, all of the Architecture Framework's paper products are subtypes of DOCUMENT. This allows all of the subtypes to inherit the relationships enjoyed by DOCUMENT. Also, changes to the product set will have minimal impact on the model because subtypes are easily added or removed.

4.3.1.3 Relationship Between the CADM and Framework Products

The CADM and the Architecture Framework's products are complementary, not alternatives. Thus, both the CADM and the Framework's products will remain important to DoD architecture processes. In essence, the CADM defines a common approach for organizing and sharing the information that is contained in the Framework products. The CADM offers flexible and automated queries while the Framework offers standardized views to facilitate comparison and integration. A database implementing the CADM can store information used to produce Framework products. It can also store the Framework products themselves.

4.3.1.4 Potential Uses for the CADM

Figure 4-43 depicts the role of the CADM as a logical basis for a (physical) DoD-wide architecture data repository. As a core of common architecture data structures, the CADM captures a set of top-down architecture data requirements and integrating common bottom-up architecture data requirements from Command/Service/Agency (C/S/A) architecture data models. As depicted at the bottom of the figure, C/S/A database systems based on the CADM also provide a mechanism for storing and sharing the information underlying common architecture products. Each C/S/A database stores data extracts from C/S/A-developed architecture products and constitutes sources for future products.

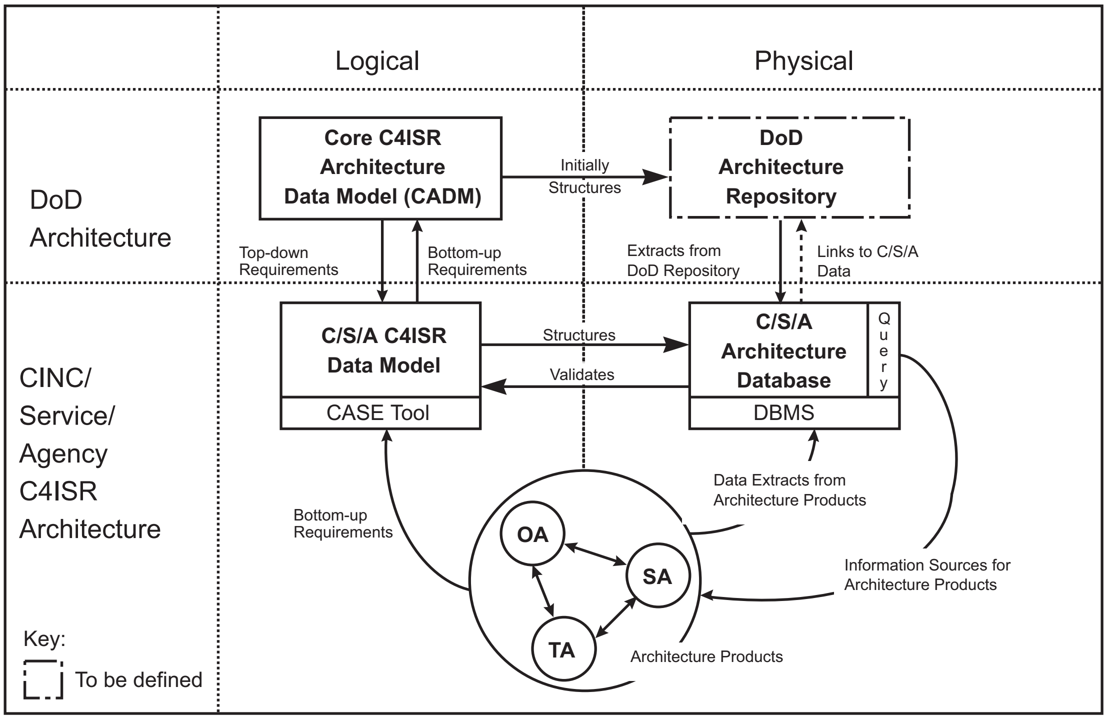


Figure 4-43. Potential Uses of the CADM

4.3.1.5 Conclusions

The CADM is truly intended to be a *core* data model that focuses on a small set of common architecture data. Individual Military Services, Commands, and Agencies will undoubtedly develop extensions to this model to meet their unique requirements. The CADM can be expected to evolve as the Architecture Framework's products, tools, and processes mature. A core architecture data model will remain a key reference for the Architecture Framework by providing a point of mediation between and among products, databases, and other logical data models.

The CADM is a conceptual, not a physical, data model. This means that its primary purpose is to specify atomic data requirements, formalizing both meaning and relationships of data. The CADM does not select the technology or other features of an physical implementation. Thus, implementers are free to choose relational, object-oriented, or other forms of a database and to develop specialized tools to create and manage architectural data and to produce the needed forms and types of architecture products. Further, implementers are free to denormalize data structures (e.g., combine tables of subtypes or make use of joined tables) for reasons such as improved performance. By designing physical databases in logical conformance to the CADM, developers and managers can improve interoperability of architecture tools, increase the exchange of architecture data, and enhance the possibility of reuse of architecture data from project to project and year to year.

The CADM captures all the data requirements specified in Version 1 and the early draft (prior to September 1997) of the C4ISR Architecture Framework. Specifically, it captures the attributes initially specified (June 1997) for the Version 2 Framework that are understood as the attributes of the Integrated Dictionary. Thus, one of the views of the CADM represents a unified schema for that Integrated Dictionary.

The CADM captures the core data requirements of both SAASE and the C4ISR Architectures Requirements Information System (ICARIS). It therefore has the capability to serve as a core data model for the JCAPS being developed by CISA for the Commands to replace ICARIS. Further, it has the potential to serve as the core data model for standardization of DoD data elements for C4ISR architecture development (one of the original purposes of SAASE).

A DoD Architecture Repository is needed. Such a repository would provide for recording and making available for review and reuse instances of architectures and their architecture descriptions. As a CADM-conformant database, the Repository would highlight the focus on data rather than form for architecture products. The Repository would support common lists of instances of TASKs, REQUIREMENTs, SYSTEMs, and ORGANIZATIONs to enhance architecture comparison and integration. Starting points for the DoD Architecture Repository would be the Integrated Data Dictionaries for the Joint Technical Architecture and the forthcoming Joint Operational Architecture.

The CADM describes the information structure of architectures. The following tasks and actions need to be accomplished before DoD architects and system builders can easily exchange architecture data:

- Use the CADM as the database model to support architecture description. Use of the CADM as the data model for implementation of tools and databases promotes interoperability for architecture data exchange. The real value of the CADM will become apparent when an initial implementation exists. The CADM simplifies sharing and reuse of architecture information. The CADM supports interoperability by providing common meanings of, and relationships among, data that are subject to exchange.

- Assign responsibility for stewardship and configuration management of the CADM. Stewardship and configuration management of the CADM is needed to support the evolution of architecture data and products. This must be resolved quickly to exploit (a) the initial consensus for the rationale behind the details of the CADM and (b) the current interest in using the CADM for architecture development activities and tools.

- Plan for and support development of an updated version of the CADM. In view of the tasks remaining to be done, the CADM's initial release cannot be considered an end-state. Instead, the CADM will continue to evolve, as will the architecture processes it supports. Thus, Version 1.0 CADM is the beginning of a new and more effective way of doing business.

- Establish a DoD Architecture Repository, together with policy and procedures to populate and maintain the Repository. A DoD Architecture Repository could be established in conjunction with the JCAPS effort. This would eliminate expensive and omission-prone data hunts that have long burdened architects and developers of joint systems. Responsibilities must be assigned for development, maintenance, and configuration management of this Repository. This requires important decisions about what architectures to include, which data elements to make mandatory (perhaps driven by the Framework essential product list), whom to assign to populate and maintain which data, and how to pay for each of these continuing tasks.

The CADM is available at http://www.cisa.osd.mil

4.3.2 Defense Data Dictionary System (DDDS)

All Views | **Universal Reference Resource**

In the DoD vision of data administration, as described in the 8000 series of Directives, data is viewed as a valuable corporate asset which must be properly managed to support the full range of the Department's needs. Pivotal to this process is a centrally-managed repository that has information about data needed by the data administration community, technical development activities, and functional activities throughout the Department. This mechanism was originally called the DoD Information Resource Dictionary System (IRDS) in DoD Directive 8320.1. Today, it is usually called the Defense Data Repository System or Suite (DDRS). The Defense Data Dictionary System (DDDS) is one currently implemented component of the DDRS.

This centrally controlled, DoD-wide data repository will be the place to receive, store, support access to, and manage standard data definitions, data formats, usage and structures (e.g., architecture, subject area models, other data model products). To facilitate data sharing and integrated systems operations, it will provide the information needed to manage and store data in physical structures that are based on logically constructed data models and related business rules. This will significantly improve the accessing, sharing and reconciling of information.

The repository is being developed under the purview of the DoD Data Administrator by devising a model based on functional, technical, operational, and personnel requirements inputs received from all functional areas. The model includes the capability to accommodate new information and new requirements. The repository, developed from this model, thus can be incrementally implemented, then maintained and updated to reflect current circumstances.

Various forms of documentation and user support services are available regarding the repository's operation, as well as all the DoD metadata and other reusable information available on which future applications and databases should be based to be in compliance with the data administration directives.

For additional information, reference DoD 8320.1-M, "DoD Data Administration Procedures," March 1994, or visit the web site at http://ssed1.ncr.disa.mil/datadmn.html

4.3.3 Levels of Information Systems Interoperability (LISI)

All Views	Universal Reference Resource

When developing, interrelating, and assessing the operational, systems, and technical views of an architecture or when comparing multiple architectures, standard disciplines and measurement criteria are needed to capture the required or postulated degrees of information-exchange interactions between and among the various architecture elements.

The products that describe the operational view must articulate the specific nature of each node-to-node needline's required information exchange(s). This articulation must be in detail sufficient to ascertain what specific "level" of information-exchange interoperability is needed on each needline to support the target mission/operation(s).

The products that describe the systems view of the architecture need to translate each needline's required operational level of interoperability into the set of system capabilities and characteristics needed to enable the requisite information exchange to be conducted effectively and interoperably. In other words, one of the first steps in transitioning from architecture products that reflect the <u>operational</u> view to products that reflect the <u>systems</u> view is to translate the operational interoperability requirements into systems interoperability requirements. This translation then provides the architect with the basis for assessing the adequacy of existing or postulated information system capabilities.

Finally, the products that describe the technical view of the architecture must complete the "view-to-view" interoperability audit trail by describing, for each system, the profile of technical standards/criteria required to implement the prescribed system capabilities to ensure that the requisite levels of interoperability are achieved across the scope of the architecture. LISI, one of the universal reference resources, provides a construct and a reference for enabling the interoperability descriptions and audit trail described above to be conducted across the spectrum of operational, systems, and technical architecture views.

LISI provides: (a) a reference model that discriminates among incremental levels of information-exchange complexity and interoperability; (b) a systems capabilities construct that associates the requisite and candidate system capabilities (including procedures, applications, infrastructure, and data) to each level; (c) cross-links from the capabilities construct to other universal reference resources (e.g., DII COE, JTA, TRM, ...) to identify the appropriate technical implementation for interoperability to be achieved; and (d) an automated process for dynamically determining and assessing operational and systems interoperability requirements, postures, and solution alternatives.

Appendix D provides a brief description of the LISI Reference Model. For further details on LISI, see the *AWG Interoperability Panel Final Report*.

4.3.4 Universal Joint Task List (UJTL)

Operational View | **Universal Reference Resource**

The *Universal Joint Task List (UJTL)* contains a comprehensive hierarchical listing of the tasks that can be performed by a Joint military force. As a common language and reference system for Joint force commanders, combat developers and training, the UJTL also is useful to planners who are describing Joint requirements, capabilities and combat activities; staff and field organizations who must relate Joint force needs to combatant command missions; and analysts who are trying to understand and integrate Joint architecture products.

Just as an English dictionary provides words and definitions that help one construct logical sentences, the UJTL provides tasks and task definitions that help commanders construct operational threads.

UJTL terms are segregated into four separate parts according to levels of war: strategic national military tasks; strategic theater tasks; operational tasks; and tactical tasks. Tasks and subtasks are indexed to reflect their placement in the hierarchical structure. An extract from one such breakdown is shown below in figure 4-44; note the "TA" label indicates a tactical task.

TA 1 CONDUCT MANEUVER
- TA 1.1 Position/Reposition Tactical Forces
  - TA 1.1.1 Prepare Forces for Movement
  - TA 1.1.2 Move Forces
  - TA 1.1.3 Close into Tactical Position
- TA 1.2 Negotiate Tactical Area of Operations
- TA 1.3 Navigate
- TA 1.4 Control or Dominate Combat Area
  - TA 1.4.1 Control or Dominate Combat Area through Fires or Fire Potential
  - TA 1.4.2 Occupy Combat Area
- TA 1.5 Coordinate Maneuver and Integrate with Firepower

Figure 4-44. Extract from the UJTL

Each of the levels of war, tasks and subtasks in the standard, along with relevant associated terms such as *mission, essential* and *Joint mission capability requirement*, is rigorously named and defined by the UJTL in accordance with Joint doctrine, tactics, techniques, procedures, and primary source documentation. The UJTL also provides for vertical and horizontal linkages between tasks within and across the levels of war. Vertical linkages connect related tasks between distinct levels of war; horizontal (or end-to-end) linkages connect fundamentally different tasks at the same level of war which must be synchronized for a military operation to succeed. The complete specification of the UJTL is available as CJCSM 3500.04, which can be consulted for further details.

4.3.5 Joint Operational Architecture (JOA)

Operational View	Universal Reference Resource

The objective of the Joint Operational Architecture (JOA) initiative is to provide focus for investments and systems lay-downs to achieve Joint interoperability in warfighting in accordance with Joint Vision 2010 Operational Concepts.

The planned approach is to first decompose UJTL tasks from strategic national through strategic theater, through operational to tactical levels, to produce generic Joint force views of functions. For each function supporting each mission area, Joint Force Activity models will be built and analyzed to produce Joint information exchange matrices and required capabilities matrices. Further details on the JOA are available by contacting the Joint Staff or in documents posted on the C4ISR Architecture Working Group homepage at http://www.cisa.osd.mil

4.3.6 DoD Technical Reference Model (TRM)

Operational View	Technical View	Universal Reference Resource

Under the purview of the DoD's Information Management initiative, the purpose of the *Technical Reference Model (TRM)* is to provide a common conceptual framework, and to define a common vocabulary so that diverse components with the DoD can better coordinate acquisition, development and support of DoD information systems. The TRM also provides a high-level representation of the information system domain showing major service areas and is to be used to increase commonality and interoperability across DoD. It is to be used as a guideline for selecting appropriate standards for implementation and systems planning.

The model is not a specific system architecture; rather, it defines a set of services and interfaces common to DoD information systems. The TRM includes a set of concepts, entities, interfaces and diagrams that provides a basis for the specification of standards. Its basic elements are those identified in the POSIX Open System Reference Model (POSIX.0). Services are partitioned into the following categories: application software entity (for mission area or support); application program interface; application platform entity; external environment interface; and external environment.

A primary objective of the TRM is to establish a context for understanding how to relate the disparate technologies needed to implement information management. The model also acts as a mechanism for identifying the key issues associated with applications portability, scalability and interoperability, with an eye towards an open systems environment. The reference model and standards profile included in the TRM define a target technical environment for the acquisition, development and support of DoD information systems. Thus the profile does not represent a final position, but is an evolutionary target to which standards and refinements will be added based on emergent technology advances. The TRM identifies classes of standards which can be referenced while constructing products that include profiling information.

Further details on the TRM are available in Volume 2 of the TAFIM. A current draft may be downloaded from the DISA Information Technology Standards Information web site at http://www.itsi.disa.mil

4.3.7 Defense Information Infrastructure Common Operating Environment (DII COE)

Operational View	**Technical View**	**Universal Reference Resource**

The *Defense Information Infrastructure Common Operating Environment (DII COE)* encompasses architecture, standards, software reuse, shareable data, interoperability, and automated integration in a cohesive framework for systems development. It is a superset of "plug and play" capabilities, from which some subset can be installed on a single workstation or at a specific operational site. *Infrastructure services* provide low-level tools for data exchange (e.g., TCP/IP, CDE, CORBA), which comprise the architectural framework for managing and distributing data flow throughout the system. *Common Support Applications* provide the architectural framework for managing and disseminating information flow throughout the system, and for sharing information among applications (e.g., common data format processing, display, information integration, visualization).

In COE-based systems, all software and data, except the operating system and basic windowing software, is packaged in self-contained units called *segments*. Segments thus are the basic COE building blocks. Each segment contains "self-descriptive" information accessible to the rest of the COE. Segments are defined in terms of the functionality they provide from the perspective of the end user, not in terms of modules that the developer might see. There are two types of segments: *COE component segments* are those which are part of the COE; whereas *mission application segments* are built on top of the COE to provide capabilities specific to a particular mission domain. The principles controlling how segments are loaded, removed or interact with one another are the same for all segments, although COE component segments are treated more strictly.

The COE offers considerable flexibility to customize an environment so that only the segments required to meet specific mission-application needs are present at runtime. This approach helps minimize the hardware resources needed to support a COE-based system. In other words, the COE is like a software "backplane" into which segments "plug," just as circuit cards plug into the hardware backplane of a computer platform. The selection of the actual components to populate a COE creates a COE *reference implementation*. The components which constitute a COE instantiation determine the specific problem domain that a COE can address (e.g., C4I for GCCS, logistics for GCSS, finance for ECPN), and how broadly defined the problem domain can be. The COE defines hardware and software infrastructure from which platform details can be drawn while constructing relevant system products.

Further details on COE are available by visiting the website at http://spider.osfl.disa.mil/dii

4.3.8 Shared Data Environment (SHADE)

Technical View	**Universal Reference Resource**

The Shared Data Environment (SHADE) is an extension of the principles of the DII COE; it is a strategy and mechanism for data sharing in the context of DII COE compliant systems. SHADE includes the necessary data access architectures, data sharing approaches, reusable software and data components, together with guidelines and standards for the development and migration of systems that meet the user's requirements for timely, accurate, and reliable data. SHADE applies to the entire requirements, build, and operational system lifecycle. SHADE focuses on facilitating interoperability by capturing and exposing systems' data assets, their metadata and data exchange requirements. The SHADE provides guidance per the layout of data onto specific platforms (servers) which can be relevant to the construction of system products, and additionally it may provide some inputs to technology forecasting.

The initial emphasis of SHADE has been on the development of reference database segments and shared databases/servers as a means of quickly providing a basic level of data access infrastructure and for reducing the number of point-to-point system interfaces. A database segment represents a standardized, configuration-managed packaging of a physical database (subset) for incorporation into the DII COE. This approach enables multiple databases to coexist on a single server and to be accessed from appropriate applications using common APIs and tools. Segments come in three varieties: unique (domain- and sponsor-specific); shared (Joint, functionally-oriented and applicable to multiple applications); and universal (widespread, "static" reference data such as look-up tables and country codes). Over 70 reference data sets have been composed to date.

Shared data servers (SDSs) and *Joint shared servers (JSSs)* are DII COE-compliant data servers which host segment collections for use by multiple systems. An SDS is presumed to be mission-specific and locally controlled and accessed. A JSS, on the other hand, is domain-specific and accepted as a Joint standard with central control and global access. The notion of the SDS plays prominently in at least one incremental migration scenario supported by SHADE, in which a legacy database is decomposed into one or more segments and moved to an SDS. Legacy applications are reengineered so they can use this new data source. Subsequently, data then resident on an SDS and/or applications modified to use this data can be reengineered to higher levels of SHADE compliance as appropriate for shared and/or Joint use.

SHADE details are available at websites http://diides.ncr.disa.mil/shade/shade.html and at http://spider.osfl.disa.mil/dii/shade/shade_page.html . A "capstone" document provides an overview of fundamental concepts, requirements, policies, architectural components, data sharing approaches, processes and procedures. An architecture document is currently in draft revision.

4.3.9 Joint Technical Architecture (JTA)

Technical View | **Universal Reference Resource**

The *Joint Technical Architecture (JTA)* draws on the *Technical Architecture Framework for Information Management (TAFIM)* to identify a common set of mandatory information technology standards and guidelines to be used in all new and upgraded C4I acquisitions across DoD. When implemented, the JTA "building codes" should facilitate the quick and seamless flow of information in support of the Warfighter. JTA standards cover: information transfer (e.g., transmit/receive protocols); information content and format (e.g., data elements or image interpretation standards); information processing; common human-computer interface (HCI); and information system security. Specific guidance and strategies for implementing the JTA are being formulated and discussed now and will be provided separately.

A fundamental principle underlying the JTA is that the responsibility for specific implementation details, enforcement decisions and mechanisms will be determined by each of the *Services and Agencies Acquisition Executives (SAE's)*. Yet at the same time the JTA applies to all other significant areas of the system lifecycle. Operational requirements developers will be guided by the JTA when developing requirements and functional descriptions that ensure interoperability. System developers will use the JTA to ensure that systems and their interfaces meet those interoperability requirements. System integrators will use the JTA to facilitate the integration of existing and new systems. And the Science and Technology community will use the JTA whenever possible to provide appropriate interfaces to *Advanced Technology Demonstrations (ATDs)* so that resulting capabilities will integrate readily into existing DoD systems. The JTA can act as a source from which standards can be drawn while constructing products which include profiling information.

The authors of the JTA are making every effort to produce a forward-looking document, which not only defines the standards to which DoD will build new and upgraded systems, but which also clearly indicates migration directions to accomplish smooth transitions towards common interoperability goals. Future versions of the JTA will extend its scope from C4I systems to include their interfaces with other key assets critical to Joint Warfighter interoperability (e.g., weapon systems, sensors, models and simulations).

The JTA is Joint configuration managed by the CINCs, Services and Agencies. *The Joint Technical Architecture, Version 1.0,* is available on disk (http://www.ntis.gov/fcpc/cpn7799.htm). The draft Version 2.0 is available at http://www-jta.itsi.disa.mil/index_nf.html

4.3.10 Pick List References

The development of architecture products drawn from a common pool of standardized architecture data is central to compliance with the Framework. The importance of providing a common language for use during architecture product creation, analysis, comparison and integration cannot be overemphasized. The control of vocabulary helps to minimize potential misrepresentations and misunderstandings of shared information, as well as assisting with data consistency and validation. This “pick list” approach may be particularly applicable in providing agreed choices for attribute entries in the Integrated Dictionary.

A well-known information interoperability problem can be described as follows. The success of a Joint operation obviously depends on the successful translation of a concept of operations into assigned tasks to various commands. This process combines doctrinal concepts (e.g., attack, destroy) and situational variables (e.g., specific location or type of enemy force), all of which must be *unambiguously understood* by the participants. One response to this problem is vocabulary standardization, such as the promulgation of pick list references which provide communities of users with agreed terms and/or definitions, usually within specific subject areas. Subscribers to such standards agree to compose the information they share using appropriate pick list selections.

The UJTL referenced above is an example of a pick list. Other well-known examples of pick lists include the “coded” data elements and acronyms used in tactical message standards such as *United States Message Text Formats (USMTF)* and the Navy’s *Over-the-Horizon Targeting Gold (OTG)* messages. Additionally, common reference sets are being implemented in SHADE (e.g., country code, security classification codes).

4.3.11 Other References

It should be noted that other existing or emerging standards, models, and descriptions may be relevant to the Framework, such as the C2 Core Data Model. The list of universal reference sources provided here is not intended to be exhaustive; it will expand and become more definitive both in content and in application as the Framework matures.

4.4 ARCHITECTURE PRODUCT INTERRELATIONSHIPS

No matter what the specific purpose is for building a particular architecture, a consistent and cohesive description needs to be developed across the operational, systems, and technical views and associated products of the architecture. Furthermore, the architecture products must reflect the prevailing DoD doctrine, policies, and direction that are appropriate for the architecture's scope and purpose.

Figure 4-45 provides a graphic that is intended to capture some of the general relationships and "threads" that logically interconnect the Framework products from one view to another. The architect needs to be continuously aware of these necessary relationships to produce an architecture that is consistent across the three views, and that provides clear traceability and connections from one view to another.

Systems and system attributes clearly need to be addressed in context with the operations they support or are intended to support, and the operational requirements that they must satisfy. System implementations must address the requisite suite of capabilities needed to satisfy the operational needs — and — they must be implemented in accordance with current DoD technical criteria. In addition, the details needed to address interoperability adequately, from operational, information-exchange requirements to system capabilities and standards needed in response to those requirements, must be well articulated.

There are many other cross-view relationships in addition to the ones shown. The architect should proactively seek opportunities to link together the various architecture products he or she builds through the creation and conscious articulation of logical threads that will make the important cross-view relationships clear.

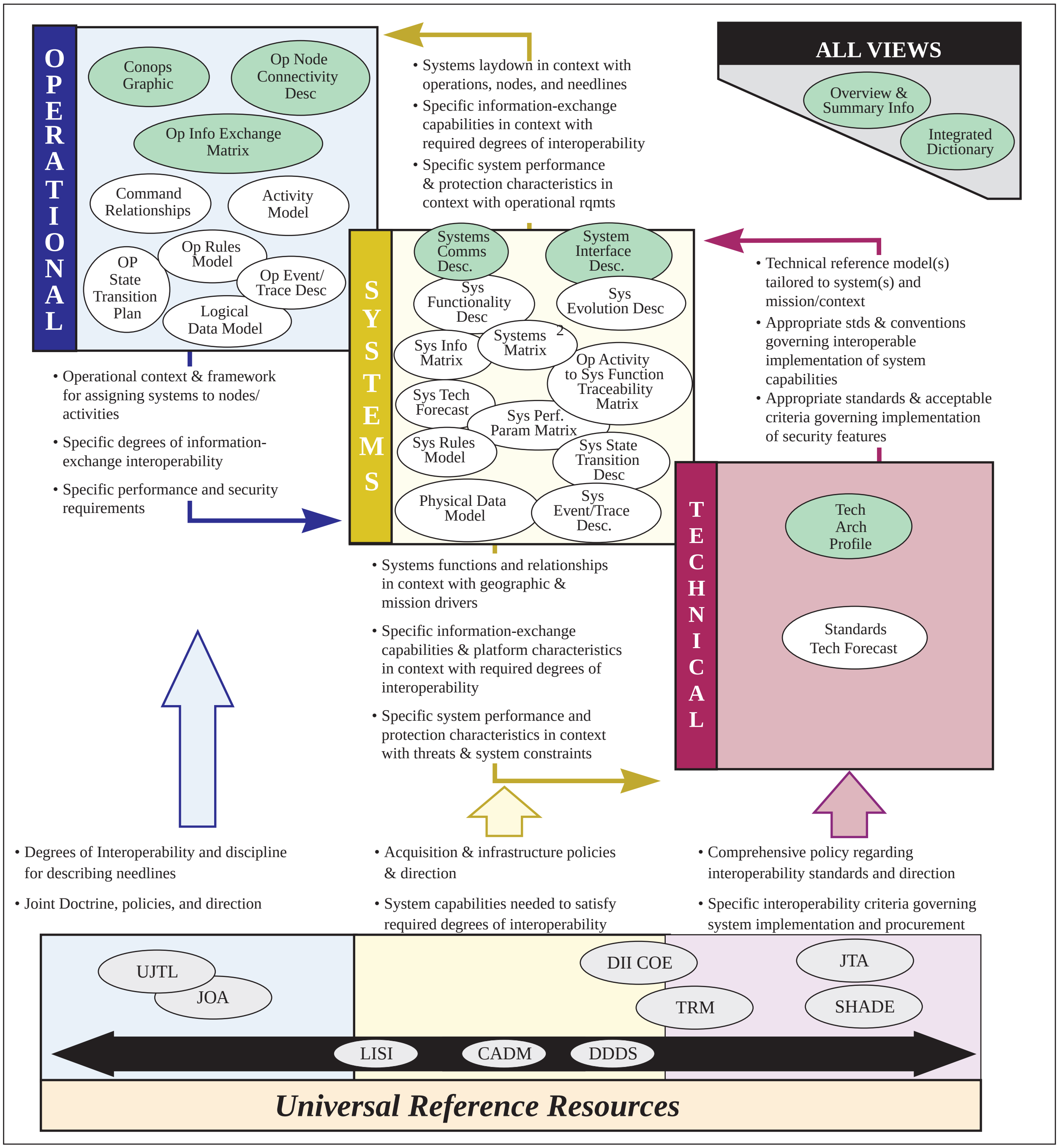


Figure 4-45. Interrelationships Among Architecture Views and Products

Intentionally Blank

SOURCES

Architecture Working Group, February 1997, *C4ISR Architecture Framework, V1.0, Lessons-Learned and Issues for Consideration.*

U.S. Army Systems Architecture Branch, *Army Systems Architecture Framework.* (For information, see the U.S. Army Systems Architecture Branch, Directorate of Combat Development, Systems Architecture Home Page posting to http://www.sysarch.gordon.army.mil.

Assistant Secretary of Defense for Command, Control, Communications and Intelligence (ASD[C3I]), March 1994, *Data Administration Procedures*, DoD 8320.1-M, Washington, DC.

C4I Integration Support Activity (formerly Intelligence and Communications Architectures), 31 October 1994, *Final Report of the Intelligence Broadcast Receive Working Group to the Intelligence Systems Board* (SECRET).

C4ISR Integration Task Force, 7 June 1996, *C4ISR Architecture Framework*, Version 1.0.

C4ISR Integration Task Force, 3 July 1996, *Integrated Architecture Panel Final Report*.

Chairman, Joint Chiefs of Staff, 23 March 1994 and 1 December 1989, *DoD Dictionary of Military and Associated Terms,* Joint Publication 1-02.

Chairman, Joint Chiefs of Staff, 13 September 1996, *Universal Joint Task List*, Version 3.0, CJCSM 3500.04A.

CISA Architectures Directorate, August 1995, *Unifying Guidance for C4I Architecture Development and Representation*, Coordination Draft.

CISA Architectures Directorate, 16 May 1997, *C4ISR Mission Assessment (CMA) Support Team Final Report.*

Data Model and Analysis Tools Panel / Architecture Working Group, 15 September 1997, *C4ISR Core Architecture Data Model*, Version 1.0.

Defense Information Systems Agency, 30 April 1996, *DoD Technical Architecture Framework for Information Management (TAFIM),* Version 3.0.

Defense Information Systems Agency, 11 July 1996, *Defense Information Infrastructure Shared Data Environment Capstone Document*, Version 1.0.

Defense Information Systems Agency, 1 July 1997, *Defense Information Infrastructure Common Operating Environment Integration & Runtime Specifications,* Version 3.0, CM-400-01-04.

Demarco, 1979, *Structured Analysis and Systems Specification*, New York: Prentice Hall.

DoD Joint Technical Architecture, 18 July 1997, Version 1.0 (updated).

DoD Data Administration Procedures, March 1994, DoD Manual 8320.1-M.

Greene, G. B, M. Medek, D. Morton, and C. Wales, *Battle Damage Assessment (BDA) - Related General Military Intelligence (GMI) Production, Dissemination and Use Functional Process Improvement (FPI) Case Study*, February 1995, MITRE MTR 95W0000026, The MITRE Corporation, McLean VA, Contract Sponsor: AF 497th Intelligence Group,

GUIDE Business Rules Project Final Report, 7 November 1995, Chicago, IL: GUIDE International Corporation.

Government Performance and Results Act of 1993

IEEE Guide to the POSIX Open System Environment, *1995*, *IEEE 1003.0,* Piscataway, NJ: Institute of Electrical and Electronics Engineers.

IEEE Standard Glossary of Data Management Terminology, *1990*, *IEEE 610.5,* Piscataway, NJ: Institute of Electrical and Electronics Engineers.

IEEE Standard Glossary of Software Engineering Terminology, 1990, IEEE STD 610.12-1990, Piscataway, NJ: Institute of Electrical and Electronics Engineers, Inc.

Information Technology Management Report Act (Clinger-Cohen Act of 1996)

Integrated Definition for Function Modeling (IDEF0), FIPS PUB 183, Gaithersburg, MD: National Institute of Standards and Technology.

Integration Definition for Information Modeling (IDEF1X), FIPS PUB 184, Gaithersburg, MD: National Institute of Standards and Technology.

Lee, LTC Robert E., *Strawman Joint Operational Architecture*, 22 June 1997, Briefing presented to the C4ISR Architecture Working Group.

The MITRE Corporation, September 1997, *LISI 97 Reference and Capabilities Maturity Model*, Draft.

Rational Software Corporation, 1997, *Unified Modeling Language Notation Guide*, Version 1.0, Santa Clara, CA: Rational Software Corporation.

Rumbaugh, Blaha, Premerlani, Eddy, and Lorensen, 1991, Object-Oriented Modeling and Design, New York: Prentice Hall.

Space and Electronic Warfare Systems Command, Deputy Chief Engineer for Architecture and Standards (SPAWAR 051-2), 17 January 1997, *Naval C4ISR Architecture Primer*, Final Draft.

Thompson, Bruce and Ariel Anderson, September 1997, “Achieving ‘Jointness’ Through Improved Levels of Information Systems Interoperability (LISI) — The Data Attribute,” Paper presented at DoD 1997 Database Colloquium, San Diego, CA.

GLOSSARY

(Dictionary of Terms)

The terms included here are terms that are used in some restrictive or special sense in this document. Certain terms are not defined (e.g., activity, event, function) since they have been left as primitives, and the ordinary dictionary usage should be assumed. Where the source for a definition is known, the reference has been provided in parentheses following the definition. Terms that are being used by both the Framework and the C4ISR Core Architecture Data Model (CADM) are marked with an asterisk.

Attribute*	A property or characteristic. (Derived from DATA-ATTRIBUTE, DDDS 4363 (A))
Communications Medium*	A means of data transmission.
Data	A representation of individual facts, concepts, or instructions in a manner suitable for communication, interpretation, or processing by humans or by automatic means (IEEE 610.12)
Data Element	A basic unit of data having a meaning and distinct units and values. (Derived from 8320.1) A uniquely named and defined component of a data definition; a data "cell" into which data items (actual values) can be placed; the lowest level of physical representation of data. (Derived from IEEE 610.5)
Data-Entity*	The representation of a set of people, objects, places, events or ideas, that share the same characteristic relationships. (DDDS 4362 (A))
Format	The arrangement, order, or layout of data/information. (Derived from IEEE 610.5)
Functional Area*	A major area of related activity (e.g., Ballistic Missile Defense, Logistics, or C2 support.) (DDDS 4198(A))
Information	The refinement of data through known conventions and context for purposes of imparting knowledge.
Information Exchange Requirement*	A requirement for the content of an information flow. Associated with an IER are such performance attributes as information size, throughput, timeliness, quality, and quantity values.
Link	The physical realization of connectivity between system nodes.

Mission*	An objective together with the purpose of the intended action. (Extension of DDDS 1(A)) Note: Multiple tasks accomplish a mission. (SPAWAR)
Mission Area*	The general class to which an operational mission belongs. (DDDS 2305(A)) Note: Within a class, the mission have common objectives.
Needline*	A requirement that is the logical expression of the need to transfer information among nodes (e.g., operational elements, system elements). (The content of the transfer[s] is specified by reference to IER[s].)
Network*	The joining of two or more nodes for a specific purpose.
Node*	A representation of an element of architecture that produces, consumes or processes data.
Operational Element	An organization or a portion of an organization or a type of organization. Note: Operational Architectures typically represent an operational element within an operational node.
Operational Node	A node that performs a role or mission.
Organization*	An administrative structure with a mission. (DDDS 345 (A))
Platform*	A system that is a physical structure that hosts systems or systems components. Note: A kind of system element in the CADM.
Process	A group of logically related activities required to execute a specific task or group of tasks. (Army Systems Architecture Framework) Note: Multiple activities make up a process. (SPAWAR)
Requirement*	A need or demand. (DDDS 12451/1 (D))
Role	A function or position (Webster's)
Service	A distinct part of the functionality that is provided a system element on one side of an interface to a system element on the other side of an interface. (Derived from IEEE 1003.0)
System	A collection of components organized to accomplish a specific function or set of functions. (IEEE 610.12)

System Element	Subset of a system that maintains a separate identity and performs a specific function.
System Function*	A data transform that supports the automation of activities or exchange requirements.
Systems Node	A node with the identification and allocation of resources (e.g., people, platforms, facilities, or systems) required to implement specific roles and missions.
Rule	Statement that defines or constrains some aspect of the enterprise.
Task	A discrete unit of work, not specific to a single organization, weapon system, or individual, that enables missions or functions to be accomplished. (Extension from UJTL, JCSM 3500.04A, 1996) Note: Multiple processes accomplish a task; a single process may support multiple tasks. (SPAWAR)

* Definitions shared between the Framework and CADM documents

ACRONYM LIST

ABCCC	Airborne Command and Control Center
ACC	Architecture Coordination Council
ACDS	Advanced Combat Direction System
ACE	Analysis and Control Element
ACOM	Atlantic Command
AEGIS	Advanced Electronics Guidance and Intercept System
AIS	Air Intelligence Squadron
AMC	Army Materiel Command
AMHS	Automatic Message Handling System
AO	Area of Operations
AOC	Air Operations Center
ASD(C3I)	Assistant Secretary of Defense (Command, Control, Communications, and Intelligence)
A&T	Acquisition and Technology
ATD	Advanced Technology Demonstration
ATO	Air Tasking Order
AV	All Views
AWACS	Airborne Warning and Control System
AWG	Architecture Working Group
BCL	Battlefield Coordination Line
BDA	Battle Damage Assessment
BDE	Brigade
BMDO	Ballistic Missile Defense Organization
C2	Command and Control
C3I	Command, Control, Communications, and Intelligence
C4I	Command, Control, Communications, Computers, and Intelligence
C4ISR	Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance
CADM	C4ISR Core Architecture Data Model
CE	Combat Element
CENTCOM	Central Command
CFF	Call For Fire
CFMCC	Combined Force Maritime Component Command
CIAD	Command Intelligence Architecture Document
CINC	Commander In Chief
CIO	Chief Information Officer
CISA	C4I Integration Support Activity
CJCS	Chairman Joint Chiefs of Staff
CJCSM	Chairman, Joint Chiefs of Staff Memorandum
CJTF	Combined Joint Task Force
COE	Common Operating Environment
COMINT	Communications Intelligence
CO/TAO	Commanding Officer/Tactical Air Officer

CORBA	Common Object Request Broker Architecture
CTT	Commander's Tactical Terminal
C/S/As	Command, Services, and Agencies
DARO	Defense Airborne Reconnaissance Office
DASD	Deputy Assistant Secretary of Defense
DDDS	Defense Data Dictionary System
DDG	Guided Missile Destroyer
DDL	Data Definition Language
DDRS	Defense Data Repository System (or Suite)
DEPSECDEF	Deputy Secretary of Defense
DIA	Defense Intelligence Agency
DISA	Defense Information Systems Agency
DIVARTY	Division Artillery
DJFLCC	Deputy Joint Forces Land Component Commander
DMSO	Defense Modeling and Simulation Office
DOCC	Deep Operations Coordination Cell
DoD	Department of Defense
DODIIS	Department of Defense Information Infrastructure System
DSNET1	Defense Secure Network 1
DSP	Defense Support Program
DSSCS	Defense Special Security Communications System
ECPN	Electronic Commerce Processing Nodes
EEI	External Environment Interface
EM/ESM	Electro-Magnetic/ Electronic Security Measures
ELINT	Electronic Intelligence
EO/IR	Electro-Optical/ Infrared Radar
ER	Entity Relationship
ERD	Entity Relationship Diagrams
F2C2	Friendly Forces Coordination Center
FCO	Fire Coordination Officer
FCT	Fire Control Team
FK	Function Key
FLTCINC	Fleet Commander-In-Chief
FPI	Functional Process Improvement
FS	Fire Support
FSCL	Fire Support Coordination Line
FSE	Fire Support Element
FSO	Fire Support Officer
FSSG	Forward Service Support Group
GCCS	Global Command and Control System
GCSS	Global Combat Support System
GENSER	General Service(s) Traffic
GFCP	Generic Front End Communications Processor
GMI	General Military Intelligence
GPRA	Government Performance and Results Act of 1993
GPS	Global Positioning System

GW	Gateway
HCI	Human-Computer Interface
HQ	Headquarters
IAP	Integrated Architectures Panel
ICARIS	Intelligence C4ISR Architectures Requirements Information System
ICOM	Inputs/Controls/Outputs/Mechanisms
ID	Identify
ID	Integrated Dictionary
IDEF	Integrated Definition language
IDHS	Intelligence Data Handling System
IEEE	Institute of Electrical and Electronics Engineers, Inc.
IER	Information Exchange Requirement
IEWCS	Intelligence and Electronic Warfare Common Sensor
IFF	Identification, Friend or Foe
INFOSEC	Information Security
IRDS	Infrastructure Resource Dictionary System
ISS	Intelligence Systems Secretariat
ITF	Integration Task Force
ITMRA	Information Technology Management Reform Act - Clinger-Cohen Act of 1996
JAF	Joint Architecture Framework
JCAPS	Joint C4ISR Architecture Planning System
JBC	Joint Battle Center
JCS	Joint Chiefs of Staff
JFACC	Joint Forces Air Component Commander
JFC	Joint Force Commander
JFLCC	Joint Force Land Component Commander
JFMCC	Joint Force Maritime Component Commander
JFSOCC	Joint Forces Special Operations Component Commander
JIC	Joint Intelligence Center
JICCENT	Joint Intelligence Center Central Command
JMCIS	Joint Maritime Command Information System
JOA	Joint Operational Architecture
JOC	Joint Operations Center
JOTS	Joint Operational Tactical System
JS	Joint Staff
JSIPS	Joint Services Imagery Processing System
JSS	Joint Shared Servers
JSTARS	Joint Surveillance Target Attack Radar System
JTA	Joint Technical Architecture
JTAMDO	Joint Theater Air and Missile Defense Organization
JWICS	Joint Worldwide Intelligence Communications System
LAN	Local Area Network
LDM	Logical Data Model
LOS	Line of Sight
LISI	Levels of Information System Interoperability
LTG	Lieutenant General (Army)

MAGTF	Marine Air-Ground Task Force
MEA	Munitions Effects Assessment
MEF	Marine Expeditionary Force
METOC	Meteorological Oceanographic
MIDB	Modernized Integrated Data Base
MOA	Memorandum of Agreement
MOE	Measure of Effectiveness
MOP	Measure of Performance
MSC	Military Sealift Command
MTBF	Mean Time Between Failures
MTMC	Military Traffic Management Command
MTTR	Mean Time to Repair
MVR	Maneuver
NAS	Network Access Switch
NCA	National Command Authorities
NCD	Node Connectivity Diagram
NIMA	National Imagery Management Agency
NM	Nautical Mile
NSA	National Security Agency
OASD	Office of the Assistant Secretary of Defense
OOB	Order of Battle
OPFAC	Operational Facility
OPLAN	Operations Plan
OTG	Over-the-Horizon Targeting Gold
OUSD	Office of the Undersecretary of Defense
OV	Operational View
PAID	Procedures, Applications, Infrastructures And Data
PDASD	Principal Deputy Assistant Secretary of Defense
PDM	Physical Data Model
POSIX	Portable Operating System Interface Standard for UNIX
PSN	Packet Switched Network
RCVR	Receiver
ROE	Rules of Engagement
SAASE	Standard Data Element-Based Automated Architecture Support Environment
SACC	Supporting Arms Coordination Center
SAE	Services and Agencies Acquisition Executives
SALT	Supporting Arms Liaison Team
SATCOM	Satellite Communications
SCI	Sensitive Compartmented Information
SDS	Shared Data Server
SECDEF	Secretary of Defense
SHADE	Shared Data Environment
SHF	Super High Frequency

SIGINT	Signals Intelligence
SIM	Systems Integration Management
SIMO	Systems Integration Management Office
SIPRNET	Secret Internet Protocol Router Network
SOFA	Status Of Forces Agreement
SSI	Single Source Integration
STD	Standard
S-TRED	Standard TRE Display
SUCCESS	Synthesized UHF Computer Controlled Equipment Subsystem
SV	Systems View
TACINTEL	Tactical Intelligence
TACMS	Tactical Missile System
TADIX	Tactical Data Information Exchange System
TAFIM	Technical Architecture Framework for Information Management
TBMD	Theater Ballistic Missile Defense
TCC	Tactical Command Center
TCN	Telecommunications Network
TCP/IP	Transport Control Protocol/Internet Protocol
TDDS	TRE/TRAP Data Dissemination System
TERPES	Tactical Electronic Reconnaissance Processing And Evaluation System
TOC	Tactical Operations Center
TRAP	TRE-Related Application
TRE	Tactical Receive Equipment
TRM	Technical Reference Model
TV	Technical View
TWCS	Tomahawk Weapons Control Systems
UAV	Unmanned Aerial Vehicle
UHF	Ultra High Frequency
UJTL	Universal Joint Task List
U.S.	United States
USARCENT	United States Army Central Command
USAREUR	United States Army Europe
USCENTCOM	United States Central Command
USCINCTRANS	United States Commander In Chief Transportation Command
USD(A&T)	Under Secretary of Defense (Acquisition & Technology)
USEUCOM	United States European Command
USMARCENT	United States Marines Central Command
USMTF	United States Message Text Format
USTRANSCOM	United States Transportation Command
VDS	Variable Depth Sonar
VHF	Very High Frequency
WCS	Weapon Control System
WOC	Wing Operations Center

APPENDIX A: PRODUCT ATTRIBUTE TABLES

APPENDIX A

PRODUCT ATTRIBUTE TABLES

A.1 Introduction

The purpose of appendix A is to provide more detailed information on the contents and characteristics (called "attributes" here for convenience) of the Framework products. In section 4 of the Framework, the products are introduced, examples and templates are provided, and the major characteristics of the products are reviewed. For each product, appendix A contains a table presenting details of the product attributes or characteristics. Each product attribute represents a piece of information about a given architecture that should be captured in the product and stored in the Integrated Dictionary. The collection of information in the Integrated Dictionary will allow the set of Framework products developed by an architecture project to be read and understood with minimal reference to outside resources. Note that not all attributes will be applicable to all architecture projects, and that not all attribute values may be available at the same time as the products are being constructed.

As the Framework is used and lessons-learned are compiled, a better understanding of all the information needed to describe architectures will emerge. As noted in the body of this document, it is envisioned that future architecture descriptions will be built using an information-focused approach rather than the current approach focused on standard products. With an information-focused approach, specified information is collected (in the Integrated Dictionary) and then user-defined products, tailored to the user's specific needs, can be generated from that information. In the future, the product attributes will merge with the C4ISR Core Architecture Data Model (CADM), discussed in sections 3.3 and 4.3.1. The CADM also supplies a pointer from each attribute definition to an applicable term in the DoD Defense Data Dictionary or DoD Enterprise Data Model, if one exists.

A.2 Attribute Tables

In the following tables, the products are presented in the order in which they are described in section 4. It should be noted that, in addition to the attributes listed, every product should have a title and an identification of the time frame for which the product is valid (e.g., "As-Is" or "To-Be," together with the relevant date).

For each product, entities, attributes, and relationships specified or implied in the product are listed in the corresponding table. For graphical products, the entities, attributes, and relationships expressed by the icons (i.e., "graphical boxes") and lines ("graphical arrows") of the graphic are addressed first, followed by "implied" entities, attributes, and relationships. These "implied" entities, attributes, and relationships are not explicit in the graphic but are indicated through the physical arrangement or juxtaposition of the icons and lines in the graphic. For example, some icons may be placed *inside* other icons to indicate containment or subordinate relationships. Also, some entities are included by implication when their attributes are used as labels or annotations to graphical features. For example, the names of information or data items may be used to label graphical lines indicating the physical communications channels used to transmit the information or data. By convention, all entities, attributes, and relationships of non-graphical products, such as matrices, are considered to be implied.

A.2.1 Attribute Tables For Essential Products

A.2.1.1 Overview and Summary Information (AV-1)

The Overview and Summary Information product provides overview and summary information in a consistent form that allows quick reference and comparison among architecture descriptions. This information includes scope, purpose and intended users, environment, and findings (i.e., analyses and decisions, if any, that used the architecture). Table A-1 describes the Integrated Dictionary entries related to the Overview and Summary Information.

Table A-1. Integrated Dictionary Attributes for Overview and Summary Information

Implied Entities, Attributes, & Relationships	Example Values/Explanation
Entities & Attributes	
•Architecture Project	
Project Name	Name/identifier of project that involves development or documentation of an architecture
Architect Name/Organization	Name of chief architect or organization charged with development or documentation on the architecture
Project Purpose	Text description of purpose of architecture development/documentation
Assumptions and Constraints	Text description, including budget and schedule constraints
•Architecture	
Architecture Name	Name of architecture being described (e.g., Naval Strike Warfare)
Date Completed	Date on which architecture description completed
•Architecture View	
Name	Name/identifier of architecture view
Type	One of: Operational, Systems, or Technical
Timeframe	As-Is, To-Be together with relevant dates (e.g., As-Is as of November 1996; To-Be for 2010)
•Architecture Product	
Name	Product name/title/identifier
Product Type	Architecture product type name (e.g., Operational Node Connectivity Description)
Timeframe	As-Is, To-Be together with relevant dates

Table A-1. Integrated Dictionary Attributes for Overview and Summary Information (Continued)

Hardcopy Location	Reference to the hardcopy document (i.e., name, date, etc.) in which product is included
Softcopy Location	Reference to softcopy database or file name
•Organization	See OV-1 Attribute Table
•Mission	
Name	Mission name/identifier
Description	Description of mission
•Geographic Configuration	
Name	Geographical context generic name
Description	Geographical context description
•Political Situation	
Name	Name/identifier for political context (e.g., coalition peace enforcement during civil war/ internal conflict)
Description	Text description of political situation
•Doctrine, Goals, and Vision	
Name	Name/identifier of document that contains doctrine, goals, or vision
Type	Doctrine, goals, or vision
Description	Text summary description of contents or relevance of doctrine, goals or vision to architecture
•Tasking	
Name	Name/identifier of tasking
Source	Source of the tasking (e.g., organization, directive, order)
Description	Text summary of tasking
•Rules, Criteria, or Conventions	
Name	Name/identifier of document that contains rules, criteria, or conventions
Type	One of: rules, criteria, or conventions
Description	Text summary description of contents or applicability of rules, criteria or conventions to architecture description development
•Analysis	
Name	Name/identifier of analysis process
Description	Description of analysis process
•Analysis Results	
Identifier	Name/identifier of analysis process instance

Table A-1. Integrated Dictionary Attributes for Overview and Summary Information (Continued)

Date Analysis Performed	Date on which analysis was performed or completed
Technique Used	Name and description of analysis technique used
Description	Text summary of results
Location	Reference to hardcopy or softcopy location of full results
•Recommendation	
Identifier	Name/identifier of recommendation or recommendation set
Description	Description of recommendations
Date Made	Date on which recommendations were made
•Tool	
Tool Name	Full name of tool, including version number and platform used
Tool Vendor	Name and context information for vendor
Tool Description	Text description of tool, including tool functions used
Tool Output Formats	File formats for tool output, or database access/report conventions for database-based tools
Relationships	
•Architecture Project Develops Architecture	
Project Name	Architecture project name/identifier
Architecture Name	Name of architecture whose description is a product of the project
•Architecture Contains Views	
Architecture Name	Architecture name/identifier
View Name	Name of view included in the architecture description (e.g., Joint Air Strike Operational Architecture)
•View Contains Products	
Architecture View	Name/identifier of architectural view
Architecture Product	Name/identifier of architecture product contained in the view
•Analysis Requires Architecture View	
Analysis Name	Name/identifier of analysis process
Architectural View Name	Name/identifier of Architectural View needed for analysis input
•Analysis Uses Architecture Product	
Analysis Name	Name/identifier of analysis process
Architecture Product Name	Name/identifier of product analyzed

Table A-1. Integrated Dictionary Attributes for Overview and Summary Information (Continued)

•Architecture Project Supports Analysis	
Architecture Project Name	Architecture Project name/identifier
Analysis Name	Name/identifier of analysis process required by project purpose
•Analysis Yields Results	
Analysis Name	Name/identifier of analysis process
Analysis Results Identifier	Identifier for results set associated with a specific execution of the analysis process
•Results Drive Recommendations	
Analysis Results Identifier	Identifier for results set associated with a specific execution of the analysis process
Recommendations Identifier	Identifier of recommendation set that was based on this specific set of results
•Results Obtained Using Tool	
Analysis Results Identifier	Identifier for results set associated with a specific execution of the analysis process
Tool Name	Full name of tool (including version number and platform) used to help produce results for this particular execution of the analysis process
•Architecture Product Developed Using Tool	
Architecture Product Name	Name/identifier of a specific architecture product
Tool Name	Full name of tool (including version number and platform) used to develop this architecture product
•Architecture Project Results in Recommendations	
Architecture Project Name	Name/identifier of Architecture Project
Recommendation Identifier	Identifier of recommendation set produced using results of analyses based on architecture views and products developed by this project
•Architecture Project Has Context Tasking	
Architecture Project Name	Name/identifier of Architecture Project
Tasking Name	Name/identifier of tasking that generated the Architecture Project
•Architecture Project Has Context Conventions	
Architecture Project Name	Name/identifier of Architecture Project
Rules, Criteria, & Conventions Name	Name/identifier of rules, criteria, or conventions that apply to this Architecture Project
•Architecture Has Context Mission	

Table A-1. Integrated Dictionary Attributes for Overview and Summary Information (Concluded)

Architecture Name	Name/identifier of architecture description
Mission Name	Name/identifier of mission associated with this architecture
•Architecture Has Context Geographic Configuration	
Architecture Name	Name/identifier of architecture description
Geographic Configuration Name	Name/identifier of geographic configuration associated with this architecture
•Architecture Has Context Political Situation	
Architecture Name	Name/identifier of architecture description
Political Situation Name	Name/identifier of political situation associated with this architecture
•Architecture Has Context Doctrine	
Architecture Name	Name/identifier of architecture description
Doctrine, Goals, & Vision Name	Name/identifier of doctrine, goals, or vision document relevant to this architecture
•Architecture Has Context Architecture	
Architecture Name	Name/identifier of architecture description
Related Architecture Name	Name/identifier of another architecture whose views or products are referenced by this architecture
•Architecture Project Involves Organizations	
Architecture Project Name	Name/identifier of an Architecture Project
Organization Name	Name/identifier of an organization involved in this Architecture Project
Organization Role	Text description of the role this organization plays in this Architecture Project

A.2.1.2 Integrated Dictionary (AV-2)

As indicated above, all the tables in this appendix describe information to be captured in the Integrated Dictionary on a product-by-product basis. Each table in the appendix lists characteristics of the related product, although not all characteristics will be relevant for all architecture projects. In the longer term, the C4ISR Core Architecture Data Model (CADM) will provide a uniform view of the overall organization for Integrated Dictionary.

A.2.1.3 High-Level Operational Concept Graphic (OV-1)

The High-Level Operational Concept Graphic product provides a graphical representation of operations in terms of such things as missions, functions, organizations, and/or asset distribution, suitable for presentation to high-level decision makers and as a means for orienting and focusing detailed discussions. Table A-2 describes the Integrated Dictionary entries related to the High-Level Operational Concept Graphic.

Table A-2. Integrated Dictionary Attributes for the High-Level Operational Concept Graphic

Entities, Attributes & Relationships	Example Values/Explanation
Graphical Box Types	
•Asset Icon	
Name	Generic asset name that appears on graphic (e.g., AWACS, fighter squadron, carrier battle group)
Representation Type	Type represented by the icon: platform, sensor, or weapon; organization; asset; mission; or task (e.g., aircraft type; air organization; air assets; air mission or task)
Description	Textual description of representation
Generic Location	Location with respect to geographic configuration on graphic
•Organization	
Name	Name of organization that appears on the graphic
Description	Text description of the organization's purpose, including the spelling out of all acronyms
(Military) Service	Army, Navy, Air Force, Marine Corps, Joint
Code/Symbol	Service office code or symbol
Role/Responsibility	Text description of the role played in the described operation
•Target Area	
Identifier	Label on graphic or other assigned identifier
Type	Type of target represented (e.g., land based installation, troops, satellite, aircraft, ships)
Description	Text description of target importance or role
Generic Location	Location with respect to geographic configuration on graphic
Graphical Arrow Types	
•Connectivity	
Name/Label	Name/identifier

Table A-2. Integrated Dictionary Attributes for the High-Level Operational Concept Graphic (Concluded)

Description	General description
Type	Logical and/or physical
Operational Information Element	For logical connections - see Attribute Table for OV-3
Media/Communication Type	For physical connections (e.g., digital, voice, image)
"From" Box	Name of source box for arrow on graphic
"To" Box	Name of destination box for arrow on graphic
•Trajectory	
Identifier	Label on graphic or other assigned identifier
Type	Class of fire represented (e.g., air-to-air, air-to-ground)
Description	Text description of trajectory, including weapons, if known (e.g., missile type, bomb type)
"From" Asset Icon Name	Name of asset icon from which trajectory begins
"To" Target Area Identifier	Identifier of target area where trajectory ends
Annotations	
•Mission	
Generic Mission Name	Name/identifier
Mode	Peace, Crisis, War, Operations Other Than War
Type	Joint, Coalition, Combined, Service-Specific
•Geographic Configuration	
Map Segment Name/ID	Name/identifier of map segment referenced (if applicable)
Type	Real or notional geography
Other	Mapping, Charting, and Geodesy (MCG) Metadata
Implied Relationships	
•Organization Has Assets	
Organization Name	Name of organization or role
Asset Icon Name	Name of asset icon, representing asset or asset type, that is associated with this organization or role

A.2.1.4 Operational Node Connectivity Description (OV-2)

The Operational Node Connectivity Description focuses on the operational nodes, the needlines between them, and the characteristics of the information exchanged. Associated activities may also be noted. Table A-3 describes the Integrated Dictionary entries for the Operational Node Connectivity Description.

Table A-3. Integrated Dictionary Attributes for the Operational Node Connectivity Description

Entities, Attributes & Relationships	Example Values/Explanation
Graphical Box Types	
•Operational Node	
Name	Name or label of node box on diagram
Description	Text description of mission or role being performed by the node
Graphical Arrow Types	
•Needline	
Name	Name/identifier of needline represented
Description	Text description of needline
"From" Operational Node	Name of node box that is the source of the node connector on the diagram
"To" Operational Node	Name of the node box that is the destination of the node connector on the diagram.
Implied Entities & Attributes	
•Operational Information Element	See OV-3 Attribute Table
•Activity	See OV-5 Attribute Table
Implied Relationships	
•Needline Is Associated With Operational Information Element	
Needline Name	Name/identifier of needline
Operational Information Exchange Name	Name/identifier of associated operational information exchange requirement
•Operational Node Has Associated Activity	
Operational Node Name	Name/identifier of operational node
Activity Name	Name/identifier of activity associated with operational node

A.2.1.5 Operational Information Exchange Matrix (OV-3)

The Operational Information Exchange Matrix product captures requirements for information exchanges between operational nodes by describing, in tabular format, the logical and operational aspects of the information exchanges called for in Operational Node Connectivity Descriptions; that is, the information and its quality requirements, along with the information source, destination, and supported activity. An Operational Information Exchange Matrix shows such characteristics as substantive content, format, and security classification, and requirements such as volume, timeliness, and required interoperability level for the information exchanges. Table A-4 describes the Integrated Dictionary entries for the Operational Information Exchange Matrix.

Table A-4. Integrated Dictionary Attributes for the Operational Information Exchange Matrix

Implied Entities, Attributes, & Relationships	Example Values/Explanation
Entities & Attributes	
•Operational Information Element	
Name	Name/identifier for the information flow associated with an Information Exchange Requirement
Description	Definition of the information element in terms of warfighter information
Media	Digital, voice, text, etc.
Size	Value range or size (i.e., number of characters or digits) of permissible data (if applicable)
Units	Feet, inches, liters, etc. (if applicable)
•Information Exchange Requirement (IER)	
Name	Name/identifier for IER
Quality Requirements	Including frequency of exchange, timeliness, and throughput
Security Requirements	Classification or other security related categorization
Interoperability Requirements	LISI or other interoperability measure
•Needline	See OV-2 Attribute Table
•Operational Node	See OV-2 Attribute Table
•Operational Element	
Name	Name/identifier of operational element
Description	Text description spelling out any acronyms in name and describing the function, role, or mission of the operational element
•Activity	See OV-5 Attribute Table
Relationships	
•Information Exchange Requirement Contains Operational Information Element	

Table A-4. Integrated Dictionary Attributes for the Operational Information Exchange Matrix (Concluded)

IER Name	Name/identifier of IER
Operational Information Element Name	Name of information specified in the IER
•Operational Node Represents Operational Element	
Operational Node Name	Name/identifier of operational node
Operational Element Name represented by the operational node	Name/identifier of operational organization or element assigned the mission or role
•Needline Involves Operational Elements	
Needline Name	Name/identifier of a needline
Producing Operational Element Name	Name of the operational element with the requirement to send information
Consuming Operational Element Name	Name of the operational element with the requirement to receive information
•Activity Is Performed By Operational Element	
Activity Name	Name/identifier of an activity
Operational Element Name	Name/identifier of the operational element performing the activity
•Needline Is Associated with Operational Information Exchange Requirement (OIER)	
Needline Name	Name/identifier of a needline
OIER Name	Name/identifier of the OIER that describes the contents of the information flow associated with the needline

A.2.1.6 System Interface Description (SV-1)

The System Interface Description product helps to link together the operational and systems architecture views by depicting the assignments of specific systems and their interfaces to the nodes and needlines described in the Operational Node Connectivity Description. Table A-5 describes the Integrated Dictionary entries associated with System Interface Descriptions.

Table A-5. Integrated Dictionary Attributes for System Interface Description

Entities, Attributes & Relationships	Example Values/Explanation
Graphical Box Types	
•Systems Node	
Name	Name or label of systems node box on diagram
Description	Text summary description of systems node role or mission and associated resources (e.g., people, platforms, facilities, systems) that perform these roles or missions
•System	
Name	Name/identifier of system
Description	Text summary of function or set of functions performed and components contained
•System Element	
Name	Name/identifier of system subset (that has separate identity and performs specific function)
Description	Description of function performed by system element
•Communications Node	See SV-2 Attribute Table
•System Component	
Name	Name/identifier of system component, including model/version number
Type	For example: hardware component; platform component (i.e., combined hardware and system software); system software; or application (i.e., mission unique) software
Description	Text description of function(s) or service(s) supported by system component
Vendor/Source	Source of system component
Graphical Arrow Types	
•Link	
Name	Name/identifier of communications link
Description	Text description of link; includes communications nodes or communications systems elements involved as well as indications as to whether link is two-way or one-way only
Protocols Supported	For example, TCP/IP; Link-11
Capacity	Throughput; channel capacity, bandwidth

Table A-5. Integrated Dictionary Attributes for System Interface Description (Continued)

Infrastructure Technology	Infrastructure technology supporting this link (e.g., radio plus frequency, encryption (if any))
Endpoint 1 Systems Node/System Element/System Component Name	Name of graphic box that is at one end of the link on the diagram; in case of one-way connections, this endpoint is the source endpoint. The endpoint of a link may also be listed as "External" if the endpoint is outside the scope of the architecture or diagram. (In other diagrams, links may be able to connect combinations including systems and communications nodes as well as systems nodes, system elements, and system components.)
Endpoint 2 Systems Node/System Element/System Component Name	Name of the graphic box that is at the other end of the link on the diagram; in case of one-way connections, this endpoint is the target endpoint. The endpoint of a link may also be listed as "External" if the endpoint is outside the scope of the architecture or diagram. (In other diagrams, links may be able to connect combinations including systems and communications nodes as well as systems nodes, system elements, and system components.)
•Component Interface	
Name	Name/identifier of component interface (these are interfaces that do not involve communications systems; they may be Application Programming Interfaces internal to a Description Text description of interface, including any API or other interface standards supported
Endpoint 1 System Component Name	Name of system component graphic box that is at one end of the component interface
Endpoint 2 System Component Name	Name of the system component graphic box that is at the other end of the component interface
Implied Entities & Attributes	
•System Function	
Name	Name/identifier of system function
Description	Text summary description of system function
•Needline	See OV-2 Attribute Table

Table A-5. Integrated Dictionary Attributes for System Interface Description (Concluded)

Implied Relationships	
•Systems Node Contains System	
Systems Node Name	Name/identifier of systems node
System Name	Name/identifier of contained system node
•System Contains System Element	
System Name	Name/identifier of system
System Element Name	Name/identifier of contained system element
•System Element Contains System Component	
System Element Name	Name/identifier of system element
System Component Name	Name/identifier of contained system component
•System Performs System Function	
System Name	Name/identifier of system
System Function Name	Name/identifier of system function performed by system
•System Element Performs System Function	
System Element Name	Name/identifier of system element
System Function Name	Name/identifier of system function performed by system element
Operational Node Maps to Systems Node	
Operational Node Name	Name/identifier of operational node
Systems Node Name	Name/identifier of systems node that performs operational role or mission
•Link Implements Needline	
Link Name	Name/identifier of link
Needline Name	Name/identifier of needline
•Link Transmits System Information Element	
Link Name	Name/identifier of link
System Information Element Name	Name/identifier of System Information Element transmitted using the link

A.2.1.7 Technical Architecture Profile (TV-1)

The Technical Architecture Profile product provides a time-phased enumeration of the relevant subset of technical standards that apply to the architecture and how they have been or are to be implemented. Table A-6 describes the Integrated Dictionary entries for the Technical Architecture Profile.

Table A-6. Integrated Dictionary Attributes for the Technical Architecture Profile

Implied Entities, Attributes, & Relationships	Example Values/Explanation
Entities	
•Standards Profile	
Name	Name/identifier of profile
Description	Text summary description covering the content of the profile, including reference to any parent profile
Applicable Date	Start date for use of the profile
•Reference Model	
Name	Name/identifier of reference model used to select services and organize standards
Description	Text summary description of technical domain addressed by the reference model
Source	Reference to the source documentation and organization supporting the reference model
• Service Area	
Name	Name/identifier for service area included in profile or forecast
Description	Textual description of service area and included services, including issues for and impacts on system architecture
Version/Date	Date or version number for the service area forecast (for use in forecast products)
•Service	
Name	Name/identifier for service
Description	Text summary description of the service
Status	Applicability of some standard for this service: for example, “now” or “future,” meaning there are current standards for this service or interface to the service; or there are expected to be some in the future
•Standard	
Standard Name	Name and ID number for standard, including maintaining organization and relevant revision dates
Description	Text summary description of content of standard
Options	Selected standard options
Parameters	Selected standard parameters
Start Date	Initial date on which the standard is applicable
End Date	Date after which the standard is no longer applicable

Table A-6. Integrated Dictionary Attributes for the Technical Architecture Profile (Continued)

•Standard Data Element	
Name	Name of identified standard data element
Reference	Source and reference number for standard definition
Version(s)	Version number for standard definitions
•Standard Data Model	
Name	Name of identified standard data model (logical or physical)
Description	Text summary description of domain covered by standard data model
Reference	Source and reference number for standard models
Version(s)	Version number for standard models
•Project-Specific Standard	
Name	Name of local, company, proprietary, or methodology-based standards that don't correspond with reference models (e.g., coding standards, design standards, test format standards) or that cover services for which other standards are not mandated
Description	Text summary description of applicability and content of project-specific standard
Options	Selected standard options
Parameters	Selected standard parameters
Relationships	
•Standards Profile Is Refinement Of Standards Profile	
Standards Profile Name	Name/identifier of a standards profile
Standards Profile Name	Name/identifier of a standards profile which is a refinement of the other profile (i.e., has more of the parameters and options selected, has selected fewer service areas, or has selected specific standards for a service out of a set of potential standards for that service offered in the more general profile)
•Standards Profile Is Based On Reference Model	
Standards Profile Name	Name/identifier of standards profile
Reference Model Name	Name of a reference model used to organize the profile's standards
•Reference Model Includes Service Area	
Reference Model Name	Name of a reference model
Service Area Name	Name of a service described in the reference model

Table A-6. Integrated Dictionary Attributes for the Technical Architecture Profile (Concluded)

•Service Area Includes Service	
Service Area Name	Name/identifier of a service area
Service Name	Name/identifier of a service included in that service area and for which standards forecasts will be performed
•Standards Profile Include Service Area	
Standards Profile Name	Name/identifier of a standards profile
Service Area Name	Name/identifier of a service area contained in the standards profile
•Standard Addresses Service	
Standard Name	Name/identifier of a standard
Service Name	Name of the service to which the standard is applicable
•Standards Profile Contains Standard	
Standards Profile Name	Name/identifier of a standards profile
Standard Name	Name/identifier of a standard contained in the profile
•Standards Profile References Standard Data Element	
Standards Profile Name	Name/identifier of standards profile
Standard Data Element	Name/identifier of a standard data element referenced in the profile
•Standards Profile References Standard Data Model	
Standards Profile Name	Name/identifier of standards profile
Standard Data Model Name	Name/identifier of a standard data model referenced in the profile
•Standard Data Model Contains Standard Data Element	
Standard Data Model Name	Name/identifier of standard data model
Standard Data Element Name	Name/identifier of standard data element used in the model
•Standards Profile Contains Project-Specific Standard	
Standards Profile Name	Name/identifier of standards profile
Project Specific Standard Name	Name of a project-specific standard contained in the profile

A.2.2 Attribute Tables For Supporting Products

A.2.2.1 Command Relationships Chart (OV-4)

The Command Relationships Chart product illustrates the hierarchy of organizations or resources in an architecture and the relationships among them (e.g., command, control, coordination). Table A-7 describes the Integrated Dictionary entries for the Command Relationship Chart.

Table A-7. Integrated Dictionary Attributes for the Command Relationships Chart

Entities, Attributes & Relationships	Example Values/Explanation
Graphical Box Types	
•Organization	See OV-1 Attribute Table
Graphical Arrow Types	
•Organizational Relationship	
Name/Label	Relationship label used on graphic
Description	Textual description of relationship
Type	For example: Direct/Command, Indirect, Situation Dependent; Coordination; Backup
Organization Name 1	Name of source organization for relationship
Organization Name 2	Name of destination organization for relationship

A.2.2.2 Activity Model (Including Overlays) (OV-5)

The Activity Model describes the applicable activities associated with the architecture, the data and/or information exchanged between activities, and the data and/or information exchanged with other activities outside the scope of the model (i.e., external interfaces). Annotations to Activity Models can further the purposes of the description with minimal additional effort by adding supplemental information onto the basic diagrams, such as indicating activity costs and specific attributes of exchanged information. Table A-8 describes the Integrated Dictionary entries for the Activity Model.

Table A-8. Integrated Dictionary Attributes for the Activity Model

Entities, Attributes & Relationships	Example Values/Explanation
Graphical Box Types	
•Activity	
Name	Name/identifier of mission/business activity
Description	Description of the activity (e.g., IDEF0 Glossary entry)

Table A-8. Integrated Dictionary Attributes for the Activity Model (Continued)

References	Any policy or doctrine references that provide further explanation of the activity
Level identifier	For leveled families of diagrams
Activity Cost	Cost for activity derived from or used in activity based costing analysis
•Operational Node	See OV-2 Attribute Table
Graphical Arrow Types	
•ICOM	
Name	Name or label of ICOM on graphic
Description	Textual description (e.g., IDEF0 Glossary entry)
Type	One of: input, output, control, mechanism
For subtype Input	
Source	Name of source activity box or "External"
Destination	Name of destination activity box
Information Element Name	Name/identifier of the Operational Information Element exchanged
For subtype Output	
Source	Name of source activity box
Destination	Name of destination activity box or "External"
Information Element Name	Name/identifier of the Operational Information Element exchanged
For subtype Control	
Source	Name of source activity box or "External"
Destination	Name of destination activity box
Information Element Name	Name/identifier of the Operational Information Element exchanged
For subtype Mechanism	
Source	Name of source activity box or "External"
Destination	Name of destination activity box
Resource type	Type of resource represented: role or system
For subtype role	
Organization	Organization name or personnel skill type
For subtype system	
System	System name or generic identifier
•Node Tree Connector	(For Activity Hierarchy Chart)
Parent Activity	Name/identifier of an activity that has a decomposition
Child Activity	Name/identifier of child (i.e., subordinate) activity
Implied Entities & Attributes	

Table A-8. Integrated Dictionary Attributes for the Activity Model (Concluded)

•Model	
Name	Name /identifier of activity model
Type	IDEF0-style model or other type of model
Purpose	Purpose of model
Viewpoint	Viewpoint of model
•Diagram	
Title	Title of diagram/graphic
Diagram Number	Level number of diagram (for leveled families of diagrams)
•Operational Information Element	See OV-3 Attribute Table
•Facing Page Text	
Identifier	Identifier/title of a page of text
Text	Text description of a diagram and its component parts
Implied Relationships	
•Diagram Belongs To Model	
Diagram Title	Title of a diagram
Model Name	Name of the model to which the diagram belongs
•Facing Page Text References Diagram	
Facing Page Text Identifier	Identifier/title for a page of text
Diagram Title	Title of the diagram which the text describes
•Activity Box Is Contained in Diagram	
Activity Name	Name/identifier of an activity
Diagram Title	Title of the diagram on which the activity box occurs.
•ICOM Is Contained in Diagram	
ICOM Name	Name/label of ICOM
Diagram Title	Title of diagram on which the ICOM appears
•Activity Is Performed At Node	
Activity Name	Name/identifier of an activity
Operational Node Name	Name/identifier of the operational node where that activity is performed.
•ICOM Corresponds To ICOM	
ICOM Name	Name of boundary ICOM on child diagram
ICOM Name	Name of activity ICOM on parent diagram
•Activity Is Parent To Activity	
Activity Name	Name of activity in parent diagram
Activity Name	Name of child activity in child diagram (i.e., diagram with larger number)

A.2.2.3 Operational Activity Sequence and Timing Descriptions (OV-16a, 6b, 6c)

Operational Activity Sequence and Timing Descriptions products include a set of three types of models needed to refine and extend the operational view, to adequately describe the dynamic behavior and performance characteristics of the business processes critical to an architecture.

The ***Operational Rules Model*** (OV-6a) extends the representation of business requirements and concept of operations by capturing, in the form of operational rules expressed in a formal language, both action assertions (constraints on the results that actions produce, such as "if-then" and integrity constraints) and derivations (algorithmically derived facts based on other terms, facts, derivations and/or action assertions). Table A-9 describes the Integrated Dictionary entries for the Operational Rules Model.

Table A-9. Integrated Dictionary Attributes for the Operational Rules Model

Implied Entities, Attributes, & Relationships	Example Values/Explanation
Entities & Attributes	
•Action Assertion	
Name	Assertion name/identifier
Description	Textual discussion on assertion
Text	Text of assertion in selected formal language
•Derivation	
Name	Assertion name/identifier
Description	Textual discussion on assertion
Text	Text of assertion in selected formal language

The ***Operational State Transition Description*** (OV-6b) describes the detailed time sequencing of activities or work flow in the business process, depicting how the current state of the system changes in response to external and internal events. Note that the *splitting* and *synchronizing* transitions mentioned below correspond to two halves of the *complex* transition illustrated in figure 4-20c. Table A-10 describes the Integrated Dictionary entries for the Operational State Transition Description.

Table A-10. Integrated Dictionary Attributes for the Operational State Transition Description

Entities, Attributes & Relationships	Example Values/Explanation
Graphical Box Types	
•State	
Name	State name
Description	Textual description as necessary
Type	One of: Simple, Nesting, Concurrent Superstate
For Concurrent Superstates	
Number of Partitions	Number of contained state charts
Graphical Arrow Types	
•Transition	
Label	Identifier or event that triggers the transition
Description	Textual description of transition
Type	One of: Simple, Splitting, Synchronizing
For Simple Transitions	
Source State Name	Name of state where transition begins
Target State Name	Name of state where transition ends
For Splitting Transitions	
Source State Name	Name of state where transition begins
Number of Target States	Number of states where transition ends
For Synchronizing Transitions	
Number of Source States	Number of state where transition begins
Target State Name	Name of state where transition ends
Implied Entities & Attributes	
•State Chart	
Name	Name/identifier of state chart
Description	Textual description of what the state chart represents
Start State Name	Name of start state for state chart
•State Activity	
Name	Name/identifier of an activity that takes place while the system is in a given state
Description	Pseudo-English or code for activity function
•Event	
Name	Name of event
Description	Textual description of the event
•Event Qualifier Attribute	
Name	Name of attribute associated with an event or transition
Definition	Textual definition of attribute
•Event Qualifier Action	

Table A-10. Integrated Dictionary Attributes for the Operational State Transition Description (Continued)

Name	Name/identifier of action associated with an event or transition
Description	Pseudo-English or code for action function
•Event Qualifier Guard	
Name	Name/identifier for a Boolean expression that must be true for the associated transition to trigger
Definition	Expression that defines the guard
•Event Qualifier Export Event	
Name	Name of an event that will be exported beyond the scope of the generating state chart
Description	Textual description of the event represented
Implied Relationships	
•Event Triggers Transition	
Transition Name	Name/identifier of a transition
Event Name	Name of the event that triggers the transition
•Transition Has Event Qualifier Attribute	
Transition Name	Name/identifier for a transition
Event Qualifier Attribute Name	Name of attribute that characterizes the transition
•Transition Has Event Qualifier Action	
Transition Name	Name/identifier for a transition
Event Qualifier Action Name	Name of action performed as a result of triggering the transition
•Transition Has Event Qualifier Guard	
Transition Name	Name/identifier for a transition
Event Qualifier Guard Name	Name of associated expression that must be true before transition can be triggered
•Transition Has Event Qualifier Export Event	
Transition Name	Name/identifier for a transition
Event Qualifier Export Event Name	Name of event that will be exported beyond the scope of the containing state chart as a result of triggering the transition
•State Has Associated Activity	
State Name	Name of a state
State Activity Name	Name of the activity performed while the system is in the given state
•Splitting Transition Has Ending State	
Transition Name	Name/identifier of a splitting transition

Table A-10. Integrated Dictionary Attributes for the Operational State Transition Description (Concluded)

State Name	Name of one of the target states of the splitting transition
•Synchronizing Transition Has Starting State	
Transition Name	Name/identifier of a synchronizing transition
State Name	Name of one of the source states for the synchronizing transition
•Nesting State Has Contained State Chart	
State Name	Name of nesting state
State Chart Name	Name of the state chart that decomposes the nesting state
•Concurrent Superstate Has Partition	
State Chart State Name	Name of concurrent super state
State Chart Name	Name of the state chart in one of the partitions
•State Chart Has Terminal State	
State Chart Name	Name/identifier of a state chart
State Name	Name of a terminal state for that state chart
•Splitting Transition Has Matching Synchronizing Transition	
Splitting Start State Name	Name of a state that is the source for a splitting transition
Synchronizing End State Name	Name of the target state where a synchronizing transition brings together the separate threads of control started by the corresponding splitting transition. Splitting and synchronizing transitions must come in corresponding pairs; each pair makes up a complex transition.

The ***Operational Event/Trace Description*** (OV-6c) can be used alone or in conjunction with the Operational State Transition Description to depict the dynamic behavior of mission processes, tracing the actions which organizations or roles must perform in a scenario or critical sequence of events (e.g., sensor-to-shooter) along a given timeline. Table A-11 describes the Integrated Dictionary entries for Operational Event/Trace Diagram.

Table A-11. Integrated Dictionary Attributes for Operational Event/Trace Description

Entities, Attributes & Relationships	Example Values/Explanation
Graphical Box Types	
•Node Event Timeline	
Operational Node Name	Name of the operational node for which this represents a timeline
Description	Text description of any assumptions or scope constraints on the timeline
Graphical Arrow Types	
•Event Timeline Cross Link	
Name	Cross Link label or name of event
Description	Textual description of event
Originating Node Event Timeline Name	Name of node event timeline where cross link begins
Terminating Node Event Timeline Name	Name of node event timeline where cross link ends
Implied Entities & Attributes	
•Operational Node	See OV-2 Attribute Table
•Event Time	
Identifier	Identifier for time event stops or starts
Timeline Position	Relative position of event on timeline
Formula	Algebraic formula for calculating time of event occurrence (i.e., starting or stopping of event) relative to beginning of node event timeline
Implied Relationships	
•Event Starts At Time	
Event Timeline Cross Link Name	Name of the event that the cross link represents or label of the cross link
Starting Event Time Identifier	Identifier of the time at which the event occurs or starts; gives the relative position of the cross link on its starting timeline; may be identical to the ending time
•Event Ends At Time	
Event Timeline Cross Link Name	Name of the event that the cross link represents or label of the cross link
Ending Event Time Identifier	Identifier of the time at which the event ends; gives the relative position of the cross link on its ending timeline; value of time should be greater than or equal to the value of the starting time, in terms of timeline position.

A.2.2.4 Logical Data Model (OV-7)

The Logical Data Model describes the data and information that are associated with the information exchanges of the architecture, within the scope and to the level of detail required for the purposes of the architecture description. The Logical Data Model documents the data requirements and structural business process rules of the operational view. This "information- centric" perspective includes information items and/or data elements, their attributes or characteristics, and their interrelationships. Table A-12 describes the Integrated Dictionary entries for the Logical Data Model.

Table A-12. Integrated Dictionary Attributes for the Logical Data Model

Entities, Attributes & Relationships	Example Values/Explanation
Graphical Box Types	
•Entity Type	
Name	Name of the type of person, place, thing, or event of interest: the type of an information exchange item
Description	Textual description of the entity type
Graphical Arrow Types	
•Relationship Type	
Name	Name/identifier of the relationship type
Description	Textual description of the relationship represented
Source Entity Type Name	Name of the entity type at the source of the relationship
Target Entity Type Name	Name of the entity type at the target of the relationship
Cardinality Designation	Examples: one to one, one to many, etc.
• Category Relationship Type	
Name	Name of the subtyping relationship
Description	Textual description of the subtype relationship represented
Source Discriminated Entity Type Name	Name of the supertype that is the source of the relationship
Discriminant Attribute Type Name	Name of the attribute type that provides the discriminant for the entity type (must be an attribute associated with the entity)
Number of Discriminant Values	Number of different subtypes (if known)
Implied Entities & Attributes	
• Attribute Type	
Name	Name of attribute type
Definition	Definition of attribute
Reference	Reference to accepted definition of attribute, if one exists) (e.g., DDDS reference)

Table A-12. Integrated Dictionary Attributes for the Logical Data Model (Concluded)

•Rule	
Name	Name/identifier of rule
Type	Examples: Null rule; child delete rule, child update rule
Text	Text of rule
•Data Domain	
Name	Name of data domain
Description	Textual description of data domain
Range Constraint	Value range allowable for attributes in data domain
Size Constraint	Maximum number of characters in display representation
Implied Relationships	
•Entity Type Is Described By Attribute Type	
Entity Type Name	Name of entity type
Attribute Type Name	Name of associated attribute type
Role of attribute	For example: Key, Foreign Key, Non-Key
•Data Domain Constrains Values of Attribute Type	
Data Domain Name	Name of data domain
Attribute Type Name	Name of attribute type whose values are selected from the data domain
•Relationship Type Has Rule	
Relationship Type Name	Name of a relationship type
Rule Type Name	Name/identifier of a rule associated with that relationship type
•Category Relationship Type Has	
Destination Entity Type	
Category Relationship Type Name	Name of subtyping relationship
Destination Entity Type Name	Name of entity type that is a subtype
Discriminant Value	Value of the discriminant attribute that is associated with the entity subtype

A.2.2.5 Systems Communications Description (SV-2)

The Systems Communications Description represents the specific communications systems pathways or networks and the details of their configurations through which the physical nodes and systems interface. This product focuses on the physical aspects of the information needlines represented in the Operational Node Connectivity Description. Table A-13 describes the Integrated Dictionary entries associated with the Systems Communications Description product.

Table A-13. Integrated Dictionary Attributes for the Systems Communications Description

Entities, Attributes & Relationships	Example Values/Explanation
Graphical Box Types	
•Systems Node	See SV-1 Attribute Table
•System	See SV-1 Attribute Table
•Communications Node	
Name	Name/identifier of systems node whose primary function is to control the transfer and movement of data or information. Examples include network switches and routers and communications satellite
Description	Text summary description of communications functions of systems node
Graphical Arrow Types	
•Link	See SV-1 Attribute Table; with this product, links connect systems nodes, communications nodes, and systems
Implied Entities & Attributes	
•Needline	See OV-2 Attribute Table
•Operational Node	See OV-2 Attribute Table
•LAN	
Name	Name/identifier of local area network
Description	Textual description of LAN, including purpose, size, and capability
•Communications Path	
Name	Name/identifier of multiple link communications pathway that describes a single way (i.e., with no options) to communicate from one systems node/system to another
Description	Textual description of path, including whether the path is one-way only or two-way
Endpoint 1 Systems Node/System Name	Name of systems node or system at one end of path; if path is one-way, this endpoint should be the source endpoint. May be listed as "External"
Endpoint 2 Systems Node/System Name	Name of systems node or system at the other end of path; if path is one-way, this endpoint should be the destination endpoint. May be listed as "External"
Number of Links	Number of links or steps in the path
•Network	
Name	Name/identifier for a Wide Area Network or Metropolitan Area Network

Table A-13. Integrated Dictionary Attributes for the Systems Communications Description (Concluded)

Description	Textual description of network purpose, size, and capability
Security Classification	Classification of data that the network is allowed to carry
Implied Relationships	
•Systems Node Contains System	See SV-1 Attribute Table
•Operational Node Maps to Systems Node	See SV-1 Attribute Table
•Link Implements Needline	See SV-1 Attribute Table
•LAN Contains Link	
LAN Name	Name/identifier of a LAN
Link Name	Name/identifier of a link that makes up part of the LAN
•Systems Node Contains LAN	
Systems Node Name	Name/identifier of a systems node
LAN Name	Name/identifier of a LAN contained within the systems node
•Communications Path Contains Link	
Communications Path Name	Name/identifier of communications path
Link Name	Name/identifier of link within the path
Link Position In Path	Position of link in the path, given in terms of number of links from endpoint 1
•Network Contains LAN	
Network Name	Name/identifier of a network
LAN Name	Name/identifier of a LAN that is part of the network
•Network Contains Link	
Network Name	Name/identifier of a network
Link Name	Name/identifier of a link that is part of the network
•Network Contains Communications Node	
Network Name	Name/identifier of a network
Communications Node Name	Name/identifier of a communications node that is part of the network
•System Is Attached to Network	
System Name	Name/identifier of a system
Network Name	Name/identifier of a network to which the system is attached
•Systems Node Is Attached to Network	
Systems Node Name	Name/identifier of a systems node
Network Name	Name/identifier of a network that is attached to the node (i.e., a network to which all systems at the systems node are connected via a common service

A.2.2.6 Systems^2 Matrix (SV-3)

The Systems^2 Matrix is a description of the system-to-system relationships identified in the various types (e.g., internodal and intranodal) of System Interface Description products. The Systems^2 Matrix is similar to an "N^2"-type matrix where the systems are listed in the rows and the columns and each cell represents a system interface, if one exists. The system-to-system interfaces can be represented using different symbols and/or color codings to indicate various interface characteristics.

Table A-14. Integrated Dictionary Attributes for the Systems^2 Matrix

Implied Entities, Attributes, & Relationships	Example Values/Explanation
Implied Entities & Attributes	
•System	See SV-1 Attribute Table
•Interface	
Name	Name/identifier of interface; may be similar to a link, network, or communications path name
Description	Textual summary description of the interface
Status	For example: existing, planned, potential, de-activated
Purpose	Category of military operations supported, such as intelligence, C2, logistics
Security Classification	Classification of the data that flows through the interface
Code Legend	Textual description of any symbol or color codings used in the matrix to represent interface characteristics
•System Information Element	See SV-6 Attribute Table
Implied Relationships	
•System Is Source of Interface	
System Name	Name/identifier of system
Interface Name	Name/identifier of a system interface for which the named system is the data/ information source (assuming the interface is one-way)
•System Is Target of Interface	
System Name	Name/identifier of system
Interface Name	Name/identifier of a system interface for which the named system is the data/ information sink (assuming the interface is one-way)
•Communications Path Enables Interface	

Table A-14. Integrated Dictionary Attributes for the Systems[2] Matrix (Concluded)

Communications Path Name	Name/identifier of communications path
Interface Name	Name/identifier of interface that used that communications path to pass data/ information
•Network Enables Interface	
Network Name	Name/identifier of a network
Interface Name	Name/identifier of an interface that uses the network to pass data/information
•Interface Transmits System Information Element	
Interface Name	Name/identifier of an interface
System Information Element Name	Name/identifier of a system information element whose information flow is implemented (in whole or in part) by the interface

A.2.2.7 Systems Functionality Description (SV-4)

The Systems Functionality Description product describes the flow of data among system functions, and the relationships between systems or system functions and activities at nodes. Variations may focus on intranode data flow, internode data flow, data flow without node considerations, and function-to-node allocations using overlays and/or annotations. Table A-15 describes the Integrated Dictionary entries associated with Systems Functionality Description.

Table A-15. Integrated Dictionary Attributes for the Systems Functionality Description

Entities, Attributes & Relationships	Example Values/Explanation
Graphical Box Types	
•System Function	See SV-1 Attribute Table
•Systems Node	See SV-1 Attribute Table
•External Data Source/Sink	
Name	Name/identifier for a data source or sink (e.g., system, node, or user) outside the scope of current diagram product
Description source or sink	Textual description of the external data
•Data Repository	
Name	Name/identifier of data store
Description	Textual summary description of data store
Graphical Arrow Types	
•Data Flow	
Name	Name/identifier of data flow (may be the same as the system information element name)

Table A-15. Integrated Dictionary Attributes for the Systems Functionality Description (Concluded)

Description	Textual description of the data flow
System Information Element Name	Name of system information element which is contained in the data flow
From System Function/External Data Source/Data Repository	Name of box entity from which the arrow originates
To System Function/External Data Sink/Data Repository	Name of box entity at which the arrow terminates
•Function Decomposition Connector	
Super Function	Name/Identifier of function that is being decomposed
Sub-Function	Name/Identifier of system sub-function into which the super-function decomposes
Implied Entities & Attributes	
•System Information Element	See SV-6 Attribute Table
Implied Relationships	
•Data Repository Is Sink For System Information Element	
Data Repository Name	Name/identifier of a data store
System Information Element Name	Name/identifier of a system information element that is input to the data store
•Data Repository Is Source For System Information Element	
Data Repository Name	Name/identifier of a data store
System Information Element Name	Name/identifier of a system information element that is output from the data store
•System Function Produces System Information Element	
System Function Name	Name/identifier of system function
System Information Element Name	Name/identifier of system information element that is output from the system function
•System Function Processes System Information Element	
System Function Name	Name/identifier of system function
System Information Element Name	Name/identifier of system information element that is input to the system function
•System Function Is Allocated To Systems Node	
System Function Name	Name/identifier of system function
Systems Node Name	Name/identifier of systems node to which the function has been allocated

A.2.2.8 Operational Activity to System Function Traceability Matrix (SV-5)

The Operational Activity to System Function Traceability Matrix helps to link the operational and systems architecture views by depicting the "many-to-many" mappings of operational activities to system functions. Table A-16 describes the Integrated Dictionary entries associated with Operational Activity to System Function Matrices.

Table A-16. Integrated Dictionary Attributes for the Operational Activity to System Function Matrix

Implied Entities, Attributes, & Relationships	Example Values/Explanation
Entities & Attributes	
•System Function	See SV-1 Attribute Table
•Operational Activity	See OV-5 Attribute Table
Relationships	
•Operational Activity Is Supported By	
System Function	
Operational Activity Name	Name/identifier of operational activity
System Function Name supports the operational activity •System Function Implements Operational	Name/identifier of system function that
Activity	
System Function Name	Name/identifier of system function
Operational Activity Name	Name/identifier of operational activity (at least partially) implemented by the system function

A.2.2.9 System Information Exchange Matrix (SV-6)

The System Information Exchange Matrix describes, in tabular format, the physical aspects of how the information exchanges called for in Operational Node Connectivity Descriptions actually are (or will be) implemented, in terms of protocols, data formats, etc. This is particularly useful for understanding the potential for overhead and constraints introduced by these choices. Table A-17 describes the Integrated Dictionary entries for the Systems Information Exchange Matrix.

Table A-17. Integrated Dictionary Attributes for the Systems Information Exchange Matrix

Implied Entities, Attributes, & Relationships	Example Values/Explanation
Entities & Attributes	
•System	See SV-1 Attribute Table
•System Element	See SV-1 Attribute Table
•Application Software	See SV-1 Attribute Table; note that Application Software is a specific type of System Component
•System Information Element	
Name	Name/identifier of system information element
Content	Definition of information element
Media	Such as digital transmission; hardcopy; voice message.
Data/Media Format	Message type (with parameters & options used); file format; digital voice transmission; etc.
Security	Security of system information element (which may be maximum classification of aggregate of operational information elements implemented)
Frequency	Frequency, timeliness, and throughput, as appropriate, including overhead for format/ protocol and transmission media used
•System Function	See SV-1 Attribute Table
Relationships	
•System Performs System Function	See SV-1 Attribute Table
•System Element Performs System Function	See SV-1 Attribute Table
•Application Software Performs System Function	
Application Software	Application software name/identifier; any system or system element that contains this component should also perform the given system function
System Function	System function name/identifier
•System Information Element Is Input To System Function	See SV-4 Attribute Table
•System Information Element Is Output From System Function	See SV-4 Attribute Table
•System Is Source of System Information Element	

Table A-17. Integrated Dictionary Attributes for the Systems Information Exchange Matrix (Concluded)

System Name	Name/identifier of system that produces the system information element as output
System Information Element Name element	Name/identifier of system information
•System Element Is Source of System Information Element	
System Element Name	Name/identifier of system element that produces the system information element as output
System Information Element Name element	Name/identifier of system information
•System Is Destination of System Information Element	
System Name	Name/identifier of system that takes the system information element as input
System Information Element Name element	Name/identifier of system information
•System Element Is Destination of System Information Element	
System Element Name	Name/identifier of system element that takes the system information element as input
System Information Element Name element	Name/identifier of system information
•Systems Information Element Implements Operational Information Element	
Systems Information Element Name element	Name/identifier of system information
Operational Information Element Name	Name/identifier of operational information element (at least partially) implemented by the system information element

A.2.2.10 System Performance Parameters Matrix (SV-7)

The System Performance Parameters Matrix builds on the System Element Interface Description by portraying the current hardware and software performance characteristics of each system, and the expected or required performance characteristics at specified times in the future, geared towards the Standards Technology Forecasts of the technical view. Table A-18 describes the Integrated Dictionary entries for the System Performance Matrix.

Table A-18. Integrated Dictionary Attributes for the System Performance Matrix

Implied Entities, Attributes, & Relationships	Example Values/Explanation
Entities & Attributes	
•System	See SV-1 Attribute Table
•System Element	See SV-1 Attribute Table
•Platform	See SV-1 Attribute Table; note that platform is a specific type of system component
•Software Application	See SV-1 Attribute Table; note that application software is a specific type of system component
•Performance Parameter Set	
Name	Name/identifier of parameter set
Number of parameters in set	Number of different performance characteristics for which measures will be taken
•Parameter Type	
Name	Name/identifier of performance characteristic (e.g., mean time between failures, maintainability, availability, system initialization time, data transfer rate, program restart time for platforms; and data throughput/capacity; response time, effectiveness, mean time between software failures for application software)
Description	Textual description of the performance characteristic and what measurements mean
Relationships	
•System Contains System Element	See SV-1 Attribute Table
•System Element Contains System Component	See SV-1 Attribute Table
•Parameter Set Includes Parameter Type	
Parameter Set Name	Name/identifier of parameter set
Parameter Type Name	Name/identifier of parameter to be included in parameter set
•System Component Has Parameter Set	
System Component Name	Name/identifier for system component such as platform or application software
Parameter Set Name	Name/identifier for the matching parameter set indicating desired set of performance characteristics
•Parameter Type Has Baseline Value	

Table A-18. Integrated Dictionary Attributes for the System Performance Matrix (Concluded)

Parameter Type Name	Name/identifier of performance characteristic (i.e., parameter)being measured
Value	Value of performance characteristic at baseline time
Timestamp	Date and time of baseline
•Parameter Type Has Intermediate Value	
Parameter Type Name	Name/identifier of performance characteristic (i.e., parameter)being measured
Value	Value of performance characteristic at a selected point in time after the baseline time
Timestamp	Date and time of measurement
•Parameter Type Has Objective Value	
Parameter Type Name	Name/identifier of performance characteristic (i.e., parameter)being measured
Value	Projected or goal value of performance characteristic at a selected time in the future
Timestamp	Date and time for projected measurement

A.2.2.11 System Evolution Description (SV-8)

System Evolution Description depicts how a suite of systems will be "modernized" over time, including evolution and/or migration steps to accommodate the specific information requirements, performance parameters and technology forecasts provided in other products. Table A-19 describes the Integrated Dictionary entries for the System Evolution Descriptions.

Table A-19. Integrated Dictionary Attributes for the System Evolution Description

Entities, Attributes & Relationships	Example Values/Explanation
Graphical Box Types	
•System	See SV-1 Attribute Table
•System Element	See SV-1 Attribute Table
•System Component	See SV-1 Attribute Table
•Migration/Integration Timeline	
Name	Name of timeline
Description	Textual description of purpose of timeline
•Milestone	
Name	Name/identifier for milestone

Table A-19. Integrated Dictionary Attributes for the System Evolution Description (Continued)

Date	Date for achieving milestone in terms of month and year or number of months from baseline date
Description	Goals to be achieved at milestone
Version	Version number for system configuration at completion of milestone
Graphical Arrow Types	
•Grouping Link	
Milestone Name	Name/identifier of the milestone when this grouping should be integrated
Group Name	Name/identifier for a set of systems, system elements, or system components
Number of Constituent Systems/System Elements/System Components	Number of systems, system elements, or system components grouped together
Implied Relationships	
•Group Contains Constituent System/System Element/System Component	
Group Name	Name/identifier for a set of systems, system elements, or system components
System/System Element/System Component Name	Name of existing systems/system elements/ system components whose migrated functionality will make up the new version at the milestone or the name/identifier of the builds/upgrades/new functionality of the evolving system that will be included in the new version at the milestone
Version number	Version number for the constituent system/ system element/system component
•Timeline Has Beginning Point	
Timeline Name	Name/identifier of timeline
Beginning Time	Date of beginning of timeline
System Name	Name of initial system configuration (for system evolution timelines)
•Timeline Has Ending Point	
Timeline Name	Name/identifier of timeline
Ending Time	Date of ending of timeline
System Name	Name of new system available at end of timeline

Table A-19. Integrated Dictionary Attributes for the System Evolution Description (Concluded)

•Timeline Contains Milestone	
Timeline Name	Name/identifier of timeline
Milestone Name	Name/identifier of milestone
Relative Position of Milestone	Position of milestone on timeline relative to beginning of timeline (e.g., first, fifteenth)

A.2.2.12 System Technology Forecasts (SV-9)

System Technology Forecasts contain predictions about the availability of emerging technologies, specific hardware/software products, and industry trends in short-, mid-, and long-term intervals (e.g., 6-, 12- and 18-month intervals), focused on technology areas relevant to the architecture's purpose. These forecasts include confidence factors for the predictions, along with issues that may affect the architecture, such as potential technology impacts. Table A-20 describes the Integrated Dictionary entries for the System Technology Forecasts.

Table A-20. Integrated Dictionary Attributes for the System Technology Forecasts

Implied Entities, Attributes, & Relationships	Example Values/Explanation
Entities & Attributes	
•Technology Forecast	
Name	Name/identifier of technology forecast
Description	Textual description of purpose of forecast
System/System Element/System Component Name	Name/identifier of system, system element, or system component for which the forecast is being performed
•Technology Area	
Name	Name/identifier for technology area included in forecast
Description	Textual description of technology area and included capabilities, including issues for and impacts on system architecture
Version/Date	Date or version number for the technology area forecast
•Technical Capability	
Name	Name of specific technical capability for which a forecast can be made
Description	Definition of the capability

Table A-20. Integrated Dictionary Attributes for the System Technology Forecasts (Concluded)

•Timed Forecast	
Name	Name/identifier of specific forecast (e.g., short term forecast for GUI trends)
Timeframe	Timeframe for which forecast is valid; usually expressed in terms of a (future) date or months from baseline
Forecast	Text of forecast
Confidence Factor	Textual description of confidence level in forecast
Implied Relationships	
•Technology Forecast Covers Technology Area	
Technology Forecast Name	Name/identifier of technology forecast document
Technology Area Name	Name/identifier of a technology area covered by the forecast document
•Technology Area Covers Technical Capability	
Technology Area Name	Name/identifier of a technology area
Technical Capability Name	Name/identifier of a technical capability included in that technology area and for which forecasts will be performed
•Technical Capability Has Timed Forecast	
Technical Capability Name	Name/identifier of a technical capability
Timed Forecast Name	Name/identifier of a specific, time sensitive forecast for the technical capability

A.2.2.13 System Activity Sequence and Timing Descriptions (SV-10a, 10b, 10c)

System Activity Sequence and Timing Descriptions consists of a set of three types of models needed to refine and extend the systems view, to adequately describe the dynamic behavior and performance characteristics of an architecture.

The ***Systems Rules Model*** (SV-10a) focuses on constraints imposed on systems functionality due to some aspect of systems design or implementation by capturing, in the form of rules expressed in a formal language, both action assertions (constraints on the results that actions produce, such as “if-then” and integrity constraints) and derivations (algorithmically derived facts based on other terms, facts, derivations and/or action assertions). Table A-21 describes the Integrated Dictionary entries for the Systems Rules Model.

Table A-21. Integrated Dictionary Attributes for the Systems Rules Model

Implied Entities, Attributes, & Relationships	Example Values/Explanation
Entities & Attributes	
•Action Assertion	
Name	Assertion name/identifier
Description	Textual discussion on assertion
Text	Text of assertion in selected formal language
•Derivation	
Name	Assertion name/identifier
Description	Textual discussion on assertion
Text	Text of assertion in selected formal language

The ***Systems State Transition Description*** (SV-10b) describes the detailed sequencing of functions in a system by depicting how the current state of the system changes in response to external and internal events, resulting in time-sequenced activities. Note that the splitting and synchronizing transitions mentioned below correspond to two halves of the complex transition illustrated in figure 4-35c. Table A-22 describes the Integrated Dictionary entries for the Systems State Transition Description.

Table A-22. Integrated Dictionary Attributes for the Systems State Transition Description

Entities, Attributes & Relationships	Example Values/Explanation
Graphical Box Types	
•State	
Name	State name
Description	Textual description as necessary
Type	One of: Simple, Nesting, Concurrent Superstate
For Concurrent Superstates	
Number of Partitions	Number of contained state charts
Graphical Arrow Types	
•Transition	
Label	Identifier or event that triggers the transition
Description	Textual description of transition
Type	One of: Simple, Splitting, Synchronizing
For Simple Transitions	
Source State Name	Name of state where transition begins
Target State Name	Name of state where transition ends
For Splitting Transitions	

Table A-22. Integrated Dictionary Attributes for the Systems State Transition Description (Continued)

Source State Name	Name of state where transition begins
Number of Target States	Number of states where transition ends
For Synchronizing Transitions	
Number of Source States	Number of state where transition begins
Target State Name	Name of state where transition ends
Implied Entities & Attributes	
•State Chart	
Name	Name/identifier of state chart
Description	Textual description of what the state chart represents
Start State Name	Name of start state for state chart
•State Activity	
Name	Name/identifier of an activity that takes place while the system is in a given state
Description	Pseudo-English or code for activity function
•Event	
Name	Name of event
Description	Textual description of the event
•Event Qualifier Attribute	
Name	Name of attribute associated with an event or transition
Definition	Textual definition of attribute
•Event Qualifier Action	
Name	Name/identifier of action associated with an event or transition
Description	Pseudo-English or code for action function
•Event Qualifier Guard	
Name	Name/identifier for a Boolean expression that must be true for the associated transition to trigger
Definition	Expression that defines the guard
•Event Qualifier Export Event	
Name	Name of an event that will be exported beyond the scope of the generating state chart
Description	Textual description of the event represented
Implied Relationships	
•Event Triggers Transition	
Transition Name	Name/identifier of a transition
Event Name	Name of the event that triggers the transition

Table A-22. Integrated Dictionary Attributes for the Systems State Transition Description (Continued)

•Transition Has Event Qualifier Attribute	
Transition Name	Name/identifier for a transition
Event Qualifier Attribute Name	Name of attribute that characterizes the transition
•Transition Has Event Qualifier Action	
Transition Name	Name/identifier for a transition
Event Qualifier Action Name	Name of action performed as a result of triggering the transition
•Transition Has Event Qualifier Guard	
Transition Name	Name/identifier for a transition
Event Qualifier Guard Name	Name of associated expression that must be true before transition can be triggered
•Transition Has Event Qualifier Export Event	
Transition Name	Name/identifier for a transition
Event Qualifier Export Event Name	Name of event that will be exported beyond the scope of the containing state chart as a result of triggering the transition
•State Has Associated Activity	
State Name	Name of a state
State Activity Name	Name of the activity performed while the system is in the given state
•Splitting Transition Has Ending State	
Transition Name	Name/identifier of a splitting transition
State Name	Name of one of the target states of the splitting transition
•Synchronizing Transition Has Starting State	
Transition Name	Name/identifier of a synchronizing transition
State Name	Name of one of the source states for the synchronizing transition
•Nesting State Has Contained State Chart	
State Name	Name of nesting state
State Chart Name	Name of the state chart that decomposes the nesting state
•Concurrent Superstate Has Partition State Chart	
State Name	Name of concurrent super state
State Chart Name	Name of the state chart in one of the partitions
•State Chart Has Terminal State	
State Chart Name	Name/identifier of a state chart
State Name	Name of a terminal state for that state chart

Table A-22. Integrated Dictionary Attributes for the Systems State Transition Description (Concluded)

•Splitting Transition Has Corresponding Synchronizing Transition	
Splitting Start State Name	Name of a state that is the source for a splitting transition
Synchronizing End State Name	Name of the target state where a synchronizing transition brings together the separate threads of control started by the corresponding splitting transition. Splitting and synchronizing transitions must come in corresponding pairs; each pair makes up a complex transition.

The ***Systems Event/Trace Description*** (SV-10c) can be used alone or in conjunction with the System State Transition Description to describe dynamic behavior, tracing the actions in a scenario or critical sequence of events along a given timeline. This product may reflect system-specific aspects or refinements of critical sequences of events described in the operational view (e.g., performance-critical scenarios). Table A-23 describes the Integrated Dictionary entries for the Systems Event/Trace Description.

Table A-23. Integrated Dictionary Attributes for the Systems Event/Trace Description

Entities, Attributes & Relationships	Example Values/Explanation
Graphical Box Types	
•Node Event Timeline	
Systems Node Name	Name of the systems node for which this represents a timeline
Description	Text description of any assumptions or scope constraints on the timeline
Graphical Arrow Types	
•Event Timeline Cross Link	
Name	Cross Link label or name of event
Description	Textual description of event
Originating Node Event Timeline Name	Name of node event timeline where cross link begins
Terminating Node Event Timeline Name	Name of node event timeline where cross link ends
Implied Entities & Attributes	
•Systems Node	See SV-1 Attribute Table
•Event Time	
Identifier	Identifier for time event stops or starts
Timeline Position	Relative position of event on timeline

Table A-23. Integrated Dictionary Attributes for the Systems Event/Trace Description (Concluded)

Formula	Algebraic formula for calculating time of event occurrence (i.e., starting or stopping of event) relative to beginning of node event timeline
Implied Relationships	
•Event Starts At Time	
Event Timeline Cross Link Name	Name of the event that the cross link represents or label of the cross link
Starting Event Time Identifier	Identifier of the time at which the event occurs or starts; gives the relative position of the cross link on its starting timeline; may be identical to the ending time
•Event Ends At Time	
Event Timeline Cross Link Name	Name of the event that the cross link represents or label of the cross link
Ending Event Time Identifier	Identifier of the time at which the event ends; gives the relative position of the cross link on its ending timeline; value of time should be greater than or equal to the value of the starting time, in terms of timeline position.

A.2.2.14 Physical Data Model (SV-11)

The Physical Data Model describes how the information represented in the Logical Data Model is actually implemented; that is, how the information exchange requirements actually are implemented and how both data entities and their relationships are maintained in the Systems Architecture. Table A-24 describes the Integrated Dictionary entries for the Physical Information Model.

Table A-24. Integrated Dictionary Attributes for the Physical Data Model

Implied Entities, Attributes, & Relationships	Example Values/Explanation
Entities and Attributes	
•Physical Data Model	
Name	Name/identifier of physical data model
Description	Textual summary description of the mechanisms used to implement the logical data model; may include several different types of mechanisms and their associated models. For example, both messages and flat files may be used.

Table A-24. Integrated Dictionary Attributes for the Physical Data Model (Continued)

Number of Component Models	Number of other types of models that make up the physical data model
•Message Model	
Message Standard Name	Name/identifier of messaging standard to be used (e.g., USMTF; TADIL A, B, J)
Message Format Name	Name/identifier of message format used within the message standard
Message Type Parameters/Options	Parameter and option values necessary to completely identify message format to be used
•File Structure Model	
File Name	Name/identifier of file used to hold data/ information
File Structure Type	Type of file structure used; this will vary by platform type (e.g., UNIX file; VSAM or FTAM for IBM/MVS platforms)
Description	Textual or code description of record structure(s) within the file
•Entity Relationship Diagram (ERD) Model	
ERD Name	Name/identifier of specific entity-relationship model
ERD Type	Name of specific form of notation used; may be tool dependent (e.g., IDEF1X; System Architect)
Softcopy Reference	Location and file format for softcopy of the specific model
•Data Definition Language (DDL) Model	
DDL Name	Name/identifier of DDL schema or file
DDL Language Type	Name of language in which the DDL is written (.e.g., SQL)
Softcopy Reference	Location and file format for the softcopy of the DDL
Relationships	
•Physical Data Model Contains Model	
Physical Data Model Name	Name/identifier of physical data model
Message Model/File Structure Model/ ERD Model/DDL Model Name	Name/identifier of one of the types of models that makes up the physical data model

Table A-24. Integrated Dictionary Attributes for the Physical Data Model (Concluded)

•Logical Model Maps to Physical Model	
Logical Model Name	Name/Identifier of logical data model
Physical Data Model Name	Name/Identifier of corresponding physical data model
Reference to Mapping Document	Location of hardcopy or softcopy of document containing the detailed mapping between the logical and physical data models; there is no generic form for this mapping - it can be complex and varies based on the types of physical models used

A.2.2.15 Standards Technology Forecast (TV-2)

The Standards Technology Forecasts provide detailed descriptions of emerging technology standards and implementing products relevant to the systems and business processes covered by the architecture in short-, mid-, and long-term intervals, with confidence factors for the predictions, along with issues that may affect the architecture. Table A-25 describes the Integrated Dictionary entries for the Standards Technology Forecasts.

Table A-25. Integrated Dictionary Attributes for the Standards Technology Forecasts

Implied Entities, Attributes, & Relationships	Example Values/Explanation
Entities & Attributes	
•Standards Forecast	
Name	Name/identifier of standards forecast
Description	Textual description of purpose of forecast
•Reference Model	See TV-1 Attribute Table
•Service Area	See TV-1 Attribute Table
•Service	See TV-1 Attribute Table
•Timed Standards Forecast	
Name	Name/identifier of specific forecast (e.g., short term forecast for HCI API standards)
Timeframe	Timeframe for which forecast is valid; usually expressed in terms of a (future) date or months from baseline
Standard Name	Name/identifier of standard
Standard Status	Expected status based on forecast; for example: approved; updated; unchanged; replaced
Discussion	Textual notes regarding standard status
Confidence Factor	Textual description of confidence level in forecast

Table A-25. Integrated Dictionary Attributes for the Standards Technology Forecasts (Concluded)

Implied Relationships	
•Standards Forecast Based on Reference Model	
Standards Forecast Name	Name/identifier of standards forecast
Reference Model Name	Name/identifier of reference model used to organize the standards in the forecast
•Reference Model Includes Service Area	See TV-1 Attribute Table
•Standards Forecast Covers Service Area	
Standards Forecast Name	Name/identifier of standards forecast
Service Area Name	Name/identifier of a service area covered by the standards forecast
•Service Area Includes Service	See TV-1 Attribute Table
•Service Has Timed Standards Forecast	
Service Name	Name/identifier of a service
Timed Standards Forecast Name	Name/identifier of a specific, time sensitive forecast for the service

Intentionally Blank

APPENDIX B: C4ISR CORE ARCHITECTURE DATA MODEL (CADM) EXTRACT

APPENDIX B

C4ISR CORE ARCHITECTURE DATA MODEL (CADM) EXTRACT

B. SUPPORT FOR ACTIVITY MODELS

1. Activity Model Diagram

a. Characteristics of the Activity Model Diagram

The Activity Model Diagram (Figure 1) of the C4ISR Core Architecture Data Model has been extracted, with technical modifications, from the DoD Data Model. This view identifies activities and information flows through entities (PROCESS-ACTIVITY and ICOM, respectively) that are independent of any data model and therefore available for reuse in various activity models.

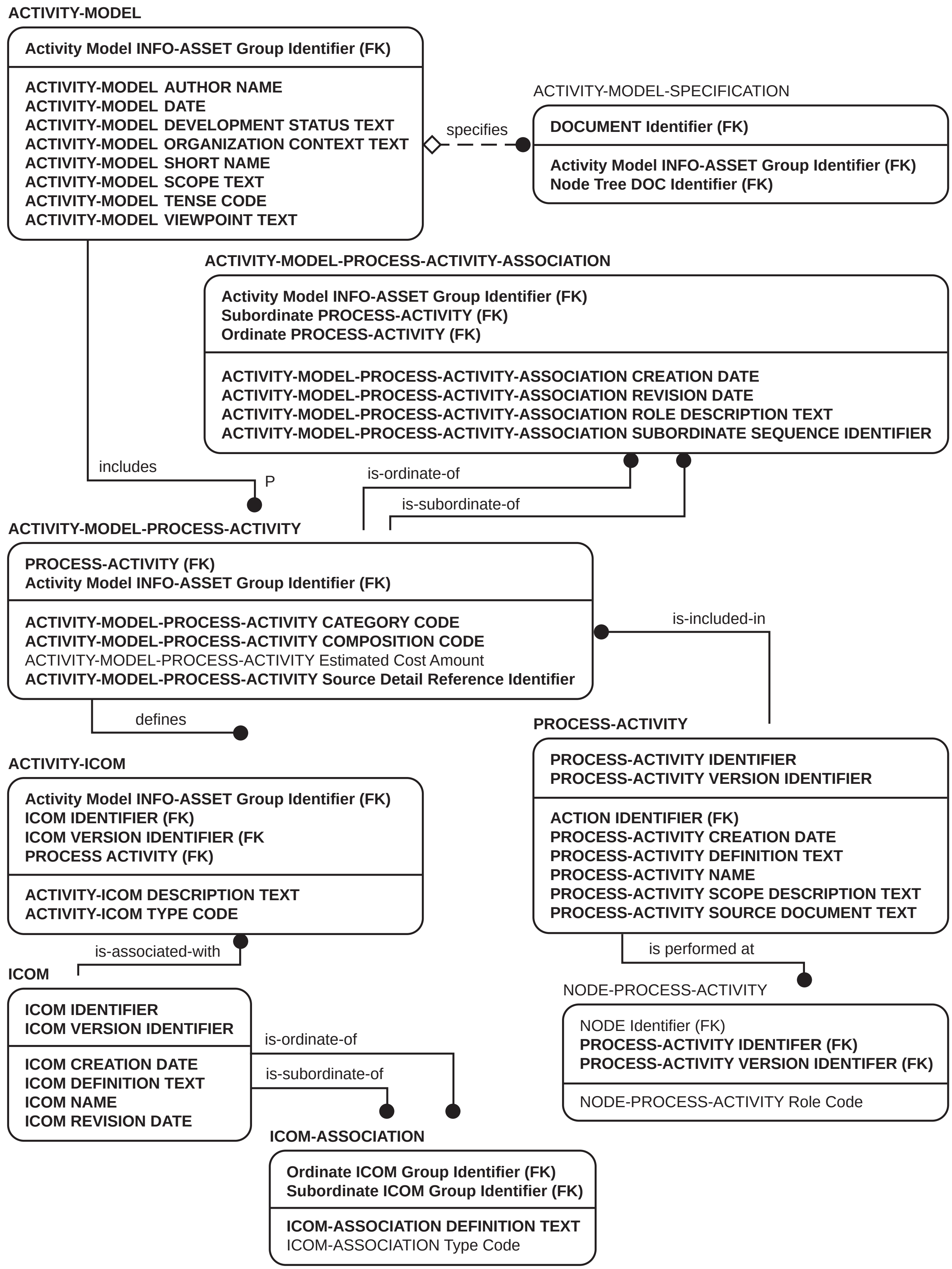
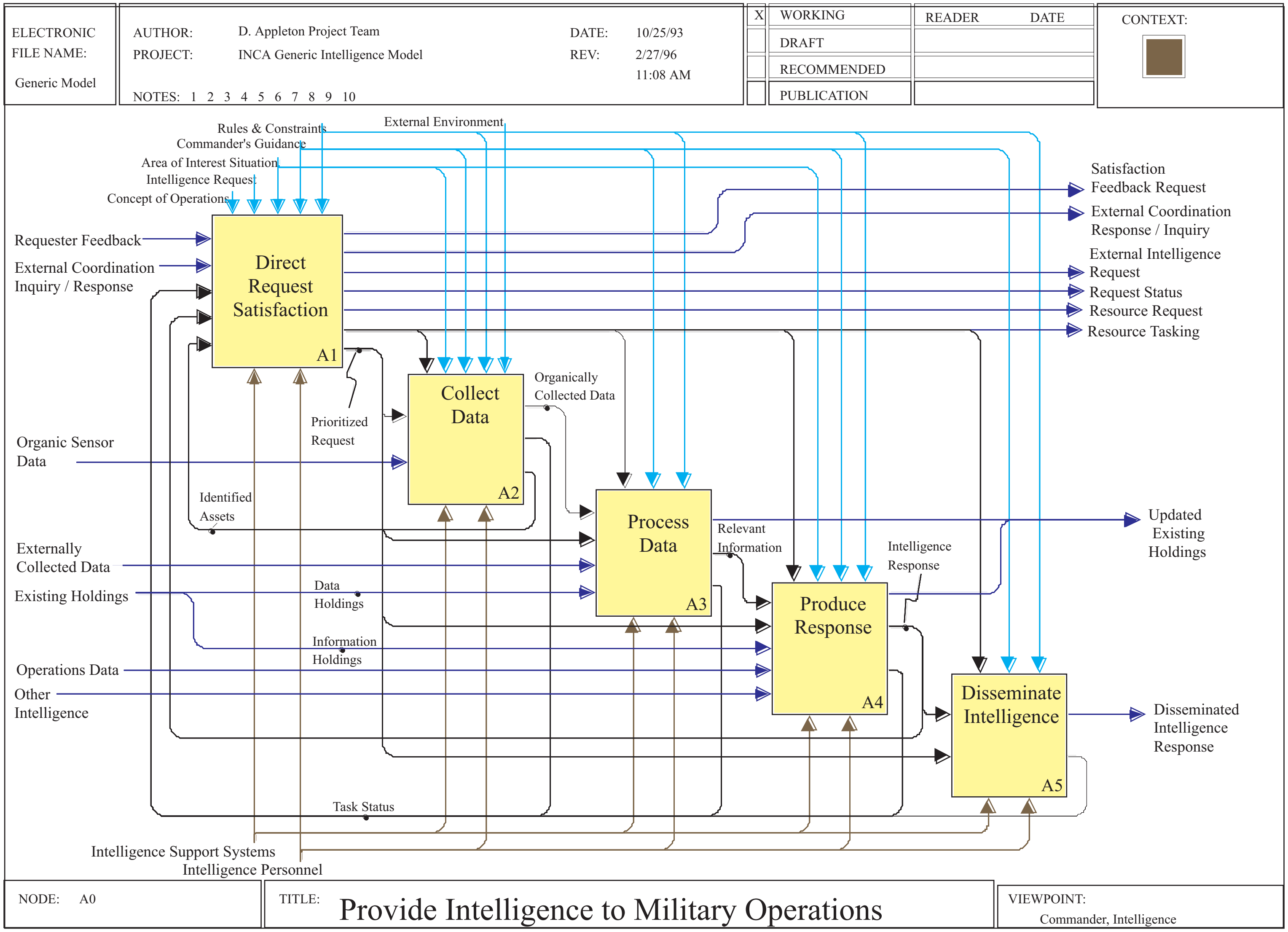


Figure 1. Entities of the CADM Supporting Activity Model Architecture Product

Each instance of an ACTIVITY-MODEL is specified in the DoD Data Model as an INFORMATION-ASSET. Thus, the connection of an ACTIVITY-MODEL to ARCHITECTURE can be made directly through a relationship ARCHITECTURE-INFORMATION-ASSET. For each architecture product (subtypes of DOCUMENT), an appropriate INFORMATION-ASSET can also be specified. For example, the DOCUMENT subtype ACTIVITY-MODEL-SPECIFICATION cites a specific ACTIVITY-MODEL that is being specified. The entity ACTIVITY-MODEL contains the details of the activities and information flows, whereas the ACTIVITY-MODEL-SPECIFICATION adds descriptive text and other information.

The data model specifies the activities in any activity model as instances of PROCESS-ACTIVITY and the activities in a *specific* activity model as instances of ACTIVITY-MODEL-PROCESS-ACTIVITY (all have the same three primary key attribute values that identify the ACTIVITY-MODEL). The associative entity ACTIVITY-MODEL-PROCESS-ACTIVITY-ASSOCIATION is used to specify which activities are components of another activity and in which order they occur. Thus, if the single entity in an A0 IDEF0 activity model is “Provide Intelligence to Military Operations” and it has five activities in its breakdown, there would be five instances of ACTIVITY-MODEL-PROCESS-ACTIVITY-ASSOCIATION each specifying “Provide Intelligence to Military Operations” as the Ordinate ACTIVITY-MODEL-PROCESS-ACTIVITY. The five subactivities would be specified as the Subordinate ACTIVITY-MODEL-PROCESS-ACTIVITY of the associative entity and be given sequence numbers 1, 2, 3, 4, and 5 (enabling one to decide which is A1, A2, A3, A4, and A5 in an IDEF0 Node Tree Diagram). This is illustrated in Figure 2, which is taken from Version 1 of the *C4ISR Architecture Framework* (July 1996). The Title Block for A0, “Provide Intelligence to Military Operations,” defines another PROCESS-ACTIVITY; thus, Figure 2 has six entities, the sixth being the entire diagram.

The data model often provides a common role name for a primary key that consists of two or more attributes. In such cases for this data model, the role name (usually) ends in the phrase “Group Identifier” (the exceptions occur when the role names are specified otherwise in the DoD Data Model). For example, in the discussion below, the three primary key attributes of INFORMATION-ASSET are given the role name INFO-ASSET Group Identifier, and the primary key (containing five attributes) of ACTIVITY-MODEL-PROCESS-ACTIVITY is given the role name ACTIVITY-MODEL-PROCESS-ACTIVITY Group Identifier.



Source: *C4ISR Architecture Framework*, Version 1.0 (Figure E-3).

Figure 2. Example Activity Diagram in IDEF0 Format

b. Discussion with Instance Tables

Table 1 provides instance tables for the entities INFORMATION-ASSET, ACTIVITY-MODEL, and ORGANIZATION—there is no meaning to the order of instance tables but all are needed to specify one instance of ACTIVITY-MODEL. These (and other examples of this section) are drawn where possible from Figure 2. Since ACTIVITY-MODEL is a subtype of INFORMATION-ASSET, it has exactly the same keys as its parent. The ORGANIZATION Identifier in INFORMATION-ASSET and ACTIVITY-MODEL identifies the organization that owns the asset.

Table 1. Instance Table for ACTIVITY-MODEL

INFORMATION-ASSET

INFO-ASSET Identifier	INFO-ASSET Version Identifier	ORGANI-ZATION Identifier	INFO-ASSET Name	INFO-ASSET Type Code	INFO-ASSET Definition Text	INFO-ASSET Comment Text	INFO ASSET Stnd Status Cd
IA2001	IAV0001	ORG0001	Generic Model	Activity Model	—	—	D

ACTIVITY-MODEL

INFO-ASSET Identifier	INFO-ASSET Version Identifier	ORGANI-ZATION Identifier	ACTIVITY MODEL Short Name	ACTIVITY MODEL Scope Text	ACTIVITY MODEL Tense Code	ACTIVITY MODEL Viewpoint Text	ACTIVITY MODEL Org Context Text
IA2001	IAV0001	ORG0001	Provide Intelligence	-	-	Commander, Intelligence	

ORGANIZATION

ORGANIZATION Identifier	ORG-ECHELON Code	ORG-TYPE Identifier (FK)	ORG Description Text	ORG Operational Element Indicator Code	ORG Category Code
ORG0001	—	JTF J2	—	Operational Element	HQ

In these and other instance tables to follow, a vertical double bar separates primary key attributes from descriptive attributes and a dotted vertical bar at the right-hand side of the table indicates that not all attributes are illustrated (there should be additional columns for a complete instance table). The term foreign key (FK) denotes those attributes whose values migrated from another (parent) entity. Thus, all three primary key attributes of ACTIVITY-MODEL are foreign key attributes originally specified in INFORMATION-ASSET, whereas in INFORMATION-ASSET only ORGANIZATION Identifier is a foreign key. The vertical bars show that ORGANIZATION has one primary key attribute and the others have three primary key attributes. For all three entities, the dotted vertical bar indicates that all three have attributes not shown in Table 1.

Table 2 identifies the six PROCESS-ACTIVITYs specified in Figure 2 (note that the overall process activity A0 should be named at the bottom of the IDEF0 diagram as “Provide Intelligence to Military Operations”). Table 2 includes an instance table for ACTIVITY-MODEL-PROCESS-ACTIVITY, which shows that each PROCESS-ACTIVITY cited in the table is a member of a single ACTIVITY-MODEL (cited in Table 1 above).

Table 2. Instance Table for PROCESS-ACTIVITY and ACTIVITY-MODEL-PROCESS-ACTIVITY

PROCESS-ACTIVITY

PROCESS-ACIVITY Indentifier	PROCESS-ACIVITY Version Identifier	PROCESS-ACIVITY Name	PROCESS-ACTIVITY Definition Text	PROCESS-ACIVITY Scope Text
PA0001	PAV0001	Direct Request Satisfaction	—	—
PA0002	PAV0001	Collect Data	—	—
PA0003	PAV0001	Process Data	—	—
PA0004	PAV0001	Produce Response	—	—
PA0005	PAV0001	Disseminate Intelligence	—	—
PA0011	PAV0001	Provide Intelligence to Military Operations	—	—

ACTIVITY-MODEL-PROCESS-ACTIVITY

Activity Model INFO-ASSET Group Identifier			PROCESS-ACTIVITY Group Identifier		ACTIVITY-MODEL-PROCESS-ACTIVITY Source Detail Ref Identifier	ACIIVITY-MODEL-PROCESS-ACTIVITY Category Code	ACTIVITY MODEL-PROCESS-ACTIVITY Com-position Code
INFO-ASSET Identifier	INFO-ASSET Version Identifier	ORGANI-ZATION Identifier	PROCESS-ACTIVITY Identifier	PROCESS-ACTIVITY Version Identifier			
IA2001	IAV0001	ORG0001	PA0001	PAV0001	—	—	—
IA2001	IAV0001	ORG0001	PA0002	PAV0001	—	—	—
IA2001	IAV0001	ORG0001	PA0003	PAV0001	—	—	—
IA2001	IAV0001	ORG0001	PA0004	PAV0001	—	—	—
IA2001	IAV0001	ORG0001	PA0005	PAV0001	—	—	—
IA2001	IAV0001	ORG0001	PA0011	PAV0001	—	—	—

Table 3 provides the five instances of ACTIVITY-MODEL-PROCESS-ACTIVITY-ASSOCIATION that, as noted, are required to specify that A1, A2, A3, A4, and A5 are all subactivities of A0 and to specify the order of occurrence. Also not shown in Table 2 (above) are instances that would identify the subactivities of A1 (A11, A12, etc.) or their order of occurrence. However, the labeling A1, A11, A12, …, A2, A21, …, etc., can be inferred entirely from the instances of ACTIVITY-MODEL-PROCESS-ACTIVITY-ASSOCIATION (using the Subordinate Sequence Number attribute shown at the right of Table 3) for the entire ACTIVITY-MODEL. These labels are often used as a shorthand for instances of PROCESS-ACTIVITY.

Table 3. Instance Table for ACTIVITY-MODEL-PROCESS-ACTIVITY-ASSOCIATION

ACTIVITY-MODEL-PROCESS-ACTIVITY-ASSOCIATION

Activity Model INFO-ASSET Group Identifier			Ordinate PROCESS-ACTIVITY Group Identifier		Subordinate PROCESS-ACTIVITY Group Identifier		ACTIVITY-MODEL-PROCESS-ACTIVITY-ASSOC Subordinate Sequence Identifier
INFO-ASSET Identifier	INFO-ASSET Version Identifier	ORGANI-ZATION Identifier	PROCESS-ACTIVITY Identifier	PROCESS-ACTIVITY Version Identifier	PROCESS-ACTIVITY Identifier	PROCESS-ACTIVITY Version Identifier	
IA2001	IAV0001	ORG0001	PA0011	PAV0001	PA0001	PAV0001	1
IA2001	IAV0001	ORG0001	PA0011	PAV0001	PA0002	PAV0001	2
IA2001	IAV0001	ORG0001	PA0011	PAV0001	PA0003	PAV0001	3
IA2001	IAV0001	ORG0001	PA0011	PAV0001	PA0004	PAV0001	4
IA2001	IAV0001	ORG0001	PA0011	PAV0001	PA0005	PAV0001	5

These instances of ACTIVITY-MODEL-PROCESS-ACTIVITY-ASSOCIATION state that PROCESS-ACTIVITY PA0011 has five sub-PROCESS-ACTIVITYs in this ACTIVITY-MODEL: PROCESS-ACTIVITYs PA0001, PA0002, PA0003, PA0004, and PA0005.

As noted, ICOM is an independent entity representing instances of an information flow. The ICOMs in a specific ACTIVITY-MODEL are identified by ACTIVITY-MODEL, which is an associative entity of ACTIVITY-MODEL-PROCESS-ACTIVITY (having five primary key attributes) and ICOM (having two additional primary key attributes).

Figure 2 identifies 31 distinct ICOMs (listed in full in Annex G of the CADM Version 1.0 Report), many of which are external to the A0 diagram (coming from or going to the edge of the diagram). Some of the ICOMs are internal to the A0 diagram, representing flows from one of its activities to another. In one case, an information flow (Existing Holdings) is split into two other information flows (Data Holdings and Information Holdings). Every information flow in Figure 2 is related to at least one activity named in the diagram as an ICOM. Some are related to more than one activity as an input (Prioritized Request is an input to A2, A3, A4, and A5); control (Rules & Constraints is a control for A1, A2, A3, A4, and A5); output (Task Status is the combined output of A2, A3, A4, and A5); and mechanism (Intelligence Support Systems is a mechanism for all five activities). Thus, there is no concept of a single “source” or a single “destination” of an ICOM. These concepts shown in Figure 2 are illustrated in the instance tables that follow (Table 4) and are completely specified by the unified set of instance tables in Annex G of the CADM Version 1.0 Report.

Table 4. Instance Table for ICOM (Partial List) and ICOM-ASSOCIATION

ICOM [Full list is provided in Annex G of the CADM Report]

ICOM Identifier	ICOM Version Identifier	ICOM Name	ICOM Definition Text	ICOM Creation Date	ICOM Revision Date
ICOM0001	ICOMV0001	Rules & Constraints	—	—	—
ICOM0002	ICOMV0001	Commander’s Guidance	—	—	—
ICOM0006	ICOMV0001	External Environment	—	—	—
ICOM0007	ICOMV0001	Requester Feedback	—	—	—
ICOM0012	ICOMV0001	Existing Holdings	—	—	—
ICOM0014	ICOMV0001	Other Intelligence	—	—	—
ICOM0015	ICOMV0001	Data Holdings	—	—	—
ICOM0016	ICOMV0001	Information Holdings	—	—	—
ICOM0019	ICOMV0001	Task Status	—	—	—
ICOM0020	ICOMV0001	Intelligence Support Systems	—	—	—
ICOM0029	ICOMV0001	Disseminated Intelligence Response	—	—	—
ICOM0030	ICOMV0001	Prioritized Request	—	—	—
ICOM0031	ICOMV0001	Organically Collected Data	—	—	—
ICOM0032	ICOMV0001	Relevant Information	—	—	—

ICOM-ASSOCIATION

Ordinate ICOM Group Identifier		Subordinate ICOM Group Identifier		
ICOM Identifier	ICOM Version Identifier	ICOM Identifier	ICOM Version Identifier	ICOM-ASSOCIATION Definition Text
ICOM0012	ICOMV0001	ICOM0015	ICOMV0001	—
ICOM0012	ICOMV0001	ICOM0016	ICOMV0001	—

These instances of ICOM-ASSOCIATION state that ICOM 12 (Existing Holdings) splits into two ICOMs: ICOM 15 (Data Holdings) and ICOM 16 (Information Holdings).

As might be expected, the instance table for ACTIVITY-ICOM is the most complex of the instance tables, primarily because there are seven primary key attributes: three attributes from the ACTIVITY-MODEL, two from the PROCESS-ACTIVITY, and two from the ICOM. The most important descriptive attribute is shown at the right of Table 5 stating whether the ICOM serves as an input, control, output, or mechanism for the cited PROCESS-ACTIVITY in the cited ACTIVITY-MODEL. Each ICOM is listed as many times as it serves in any of the four roles in the data model diagram. For example, in Table 5:

- Requester Feedback (ICOM Id 7) is listed only twice, one as an input for A1 (Direct Request Satisfaction, PROCESS-ACTIVITY Id 1) and once as an input for A0 (Provide Intelligence to Military Operations, PROCESS-ACTIVITY Id 11).
- Intelligence Support Systems (ICOM 20) is listed six times (always as a control), once for each PROCESS-ACTIVITY.
- Prioritized Request (ICOM Id 30) is listed as an output for A1 (PROCESS-ACTIVITY Id 1) and as an input to A2, A3, A4, and A5.
- Relevant Information (ICOM Id 32) is listed twice, once as an output of A3 and once as an input to A4.

Table 5. Instance Table for ACTIVITY-ICOM (Partial List)

ACTIVITY-ICOM [Full list is provided in Annex G of the CADM Report]

Activity Model INFO-ASSET Group Identifier			PROCESS-ACTIVITY Group Identifier		ICOM Group Identifier		
INFO-ASSET Identifier	INFO-ASSET Version Identifier	ORGANI-ZATION Identifier	PROCESS-ACTIVITY Identifier	PROCESS-ACIVITY Version Identifier	ICOM Identifier	ICOM Version Identifier	ACTIVITY-ICOM Category Code
IA2001	IAV0001	ORG0001	PA0011	PAV0001	ICOM0007	ICOMV0001	Input
IA2001	IAV0001	ORG0001	PA0001	PAV0001	ICOM0007	ICOMV0001	Input
IA2001	IAV0001	ORG0001	PA0011	PAV0001	ICOM0020	ICOMV0001	Mechanism
IA2001	IAV0001	ORG0001	PA0001	PAV0001	ICOM0020	ICOMV0001	Mechanism
IA2001	IAV0001	ORG0001	PA0002	PAV0001	ICOM0020	ICOMV0001	Mechanism
IA2001	IAV0001	ORG0001	PA0003	PAV0001	ICOM0020	ICOMV0001	Mechanism
IA2001	IAV0001	ORG0001	PA0004	PAV0001	ICOM0020	ICOMV0001	Mechanism
IA2001	IAV0001	ORG0001	PA0005	PAV0001	ICOM0020	ICOMV0001	Mechanism
IA2001	IAV0001	ORG0001	PA0002	PAV0001	ICOM0030	ICOMV0001	Input
IA2001	IAV0001	ORG0001	PA0003	PAV0001	ICOM0030	ICOMV0001	Input
IA2001	IAV0001	ORG0001	PA0004	PAV0001	ICOM0030	ICOMV0001	Input
IA2001	IAV0001	ORG0001	PA0005	PAV0001	ICOM0030	ICOMV0001	Input
IA2001	IAV0001	ORG0001	PA0001	PAV0001	ICOM0030	ICOMV0001	Output
IA2001	IAV0001	ORG0001	PA0004	PAV0001	ICOM0032	ICOMV0001	Input
IA2001	IAV0001	ORG0001	PA0003	PAV0001	ICOM0032	ICOMV0001	Output

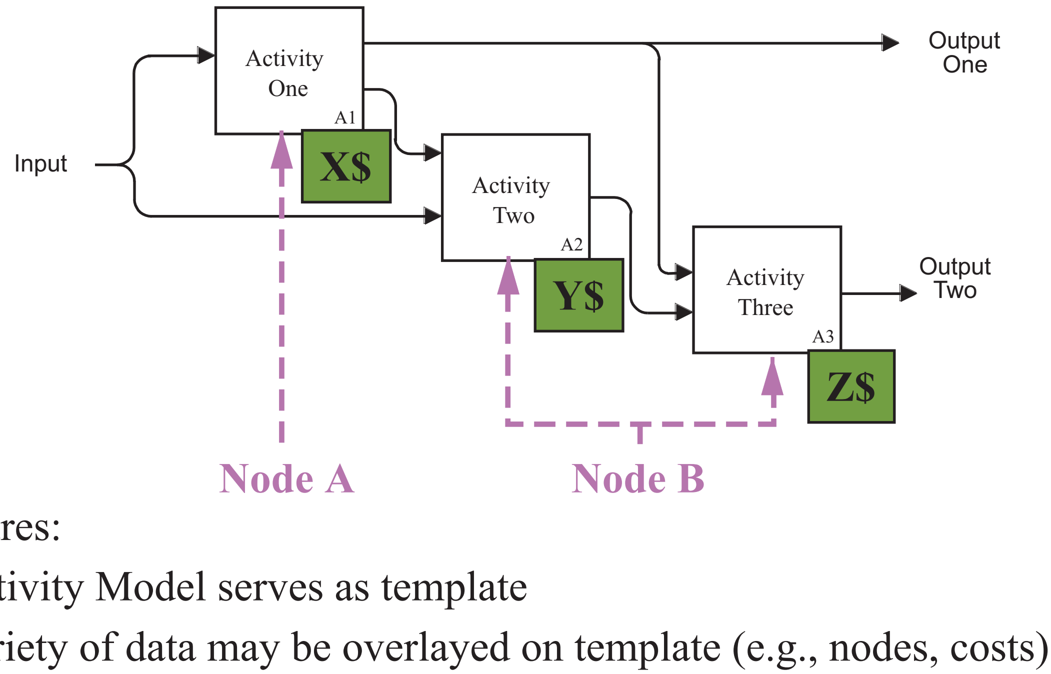
c. Key Entities and their Attributes

The key entities in the C4ISR Core Architecture Data Model for ACTIVITY-MODEL are defined as follows (the attributes are defined by entity in Section IV.C in the discussion of ACTIVITY-MODEL and chronologically in Annex D of the CADM Version 1.0 Report):

- ACTIVITY-ICOM—(4182) (A) An associative entity that identifies an ACTIVITY-MODEL-ACTIVITY with an ICOM.
- ACTIVITY-ICOM-ASSOCIATION—(4391) (A) The relationship between one ACTIVITY-ICOM and another.
- ACTIVITY-MODEL—(4187) (A) A representation of the interrelated functions of a system.
- ACTIVITY-MODEL-PROCESS-ACTIVITY—(4188) (A) The association of an ACTIVITY-MODEL with a PROCESS-ACTIVITY.
- ACTIVITY-MODEL-PROCESS-ACTIVITY-ASSOCIATION—(4192) (A) The association of one ACTIVITY-MODEL-PROCESS-ACTIVITY to another ACTIVITY-MODEL-PROCESS-ACTIVITY.
- ICOM—(4199) (A) Material related to one or more ACTIVITY-MODEL-ACTIVITYs.
- ICOM-ASSOCIATION—(4202) (A) The association of one ICOM to another ICOM.
- INFORMATION-ASSET—(4246) (A) An information resource.
- PROCESS-ACTIVITY—(4204) (A) The representation of a means by which a process acts on some input to produce a specific output.

2. Activity Model Overlay

This section will provide instances tables to show how the CADM can capture information represented in Figure 3.



Source: *C4ISR Architecture Framework*, Version 1.0 (Figure 4-4).

Figure 3. Example Activity Model Overlay

Table 6 specifies the three activities of Figure 3 in terms of the CADM. Each activity is an instance of PROCESS-ACTIVITY and further related to a single instance of ACTIVITY-MODEL by recording such associations in ACTIVITY-MODEL-PROCESS-ACTIVITY. Each PROCESS-ACTIVITY is related to NODE through NODE-PROCESS-ACTIVITY, as shown in the lower part of Table 6. The estimated costs are recorded as instances of ACTIVITY-MODEL-PROCESS-ACTIVITY, as shown in middle of Table 6.

Table 6. Specifying Data for the Activity Model Overlay in the CADM

ACTIVITY-MODEL

INFO-ASSET Identifier	INFO-ASSET Version Id	ORGANI-ZATION Identifier	ACTIVITY MODEL Short Name	MODEL Scope Text	ACTIVITY MODEL Tense Code	ACTIVITY MODEL Organizational Context Text
IA5001	IAV0001	ORG0001	Overlay Example	—	—	—

PROCESS-ACTIVITY

PROCESS-ACTIVITY Identifier	PROCESS-ACTIVITY Version Identifier	PROCESS-ACTIVITY Name	PROCESS-ACTIVITY Definition Text	PROCESS-ACTIVITY Scope Text
PA5001	PAV0001	Activity One	—	—
PA5002	PAV0001	Activity Two	—	—
PA5003	PAV0001	Activity Three	—	—

ACTIVITY-MODEL-PROCESS-ACTIVITY

Activity Model INFO-ASSET Group Identifier			PROCESS-ACTIVITY Group Identifier		ACTIVITY-MODEL-	ACTIVITY-MODEL-
INFO-ASSET Identifier	INFO-ASSET Version Identifier	ORGANI-ZATION Identifier	PROCESS-ACTIVITY Identifier	PROCESS-ACTIVITY Version Identifier	PROCESS-ACTIVITY Source Detail Ref Identifier	PROCESS-ACTIVITY Estimated Cost Amount
IA5001	IAV0001	ORG0001	PA5001 (Activity One)	PAV0001	—	X$
IA5001	IAV0001	ORG0001	PA5002 (Activity Two)	PAV0001	—	Y$
IA5001	IAV0001	ORG0001	PA5003 (Activity Three)	PAV0001	—	Z$

NODE

NODE Identifier	NODE Category Code	NODE Description Text	NODE Identifier Description Text	NODE Name	NODE Physical Idicator Code
NOD5001	—	—	—	Node A	—
NOD5002	—	—	—	Node B	—

NODE-PROCESS-ACTIVITY

	PROCESS-ACTIVITY Group Identifier (FK)		
NODE Identifier (FK)	PROCESS-ACTIVITY Identifier	PROCESS-ACTIVITY Version Identifier	NODE-PROCESS-ACIVITY Role Code
NOD5001 (Node A)	PA5001 (Activity One)	PAV0001	—
NOD5002 (Node B)	PA5002 (Activity Two)	PAV0001	—
NOD5002 (Node B)	PA5003 (Activity Three)	PAV0001	—

APPENDIX C: STANDARD WARFIGHTER INFORMATION

APPENDIX C
STANDARD WARFIGHTER INFORMATION

To date, there is no community-accepted, standard taxonomy of warfighter information, i.e., that information that is required by the warfighter to accomplish his missions, and that all Commands, Services, and DoD Agencies can use to describe the information categories and elements that are the subject of their information exchanges. Table C-1 presents a high-level example of the kind of information taxonomy that needs to be built.

TABLE C-1
EXAMPLE WARFIGHTER INFORMATION

Information Type	Information Category	Definition
1. Situational	1.1 Force Assessments	Data about an aggregation of military personnel, weapon systems, vehicles and necessary support, or combination thereof. Also a major subdivision of a fleet. (JCS 1-02)
	1.2 Platform/Unit (Track)	A physical tactical object.
	1.3 Areas and Points	Data about spatial areas including areas as defined in JCS 1-02 and civilian areas.
2. Physical Environment	2.1 Geography, Terrain, and Hydrography	Description of the static, physical characteristics of the theater of operations.
	2.2 Atmospheric and meteorological information	Climatological facts pertaining to the envelope of air surrounding the Earth, including its interfaces and interactions with the Earth's solid or liquid surface, such as wind, temperature, air density, and other phenomena which affect military operations
	2.3 Oceanographic and acoustic information	Data resulting from the study of the sea, embracing and integrating all knowledge pertaining to the sea and its physical boundaries, the chemistry and physics of seawater (including the propagation characteristics of sound), and marine biology.
	2.4 Weather	Standard descriptors of weather, such as temperature, barometric pressure, humidity, visibility, precipitation, and cloud cover.
	2.5 Acoustic Propagation Conditions	Acoustic propagation conditions. Information on conditions which affect the performance of acoustic sensors.
	2.6 EM, EO, IR Propagation Conditions	Information on conditions which affect the performance of sensors and communications systems using the atmosphere.
	2.7 Hazards to Surface and Air Navigation	Hazards to sea, air, land navigation. Traffic, natural features, obstacles, or environmental conditions, such as thunderstorms, which may threaten safe movement.
	2.8 Detectable Battlefield Phenomenon	Smoke and other temporary phenomena that are detectable by battlefield sensors

TABLE C-1
EXAMPLE WARFIGHTER INFORMATION (CONT)

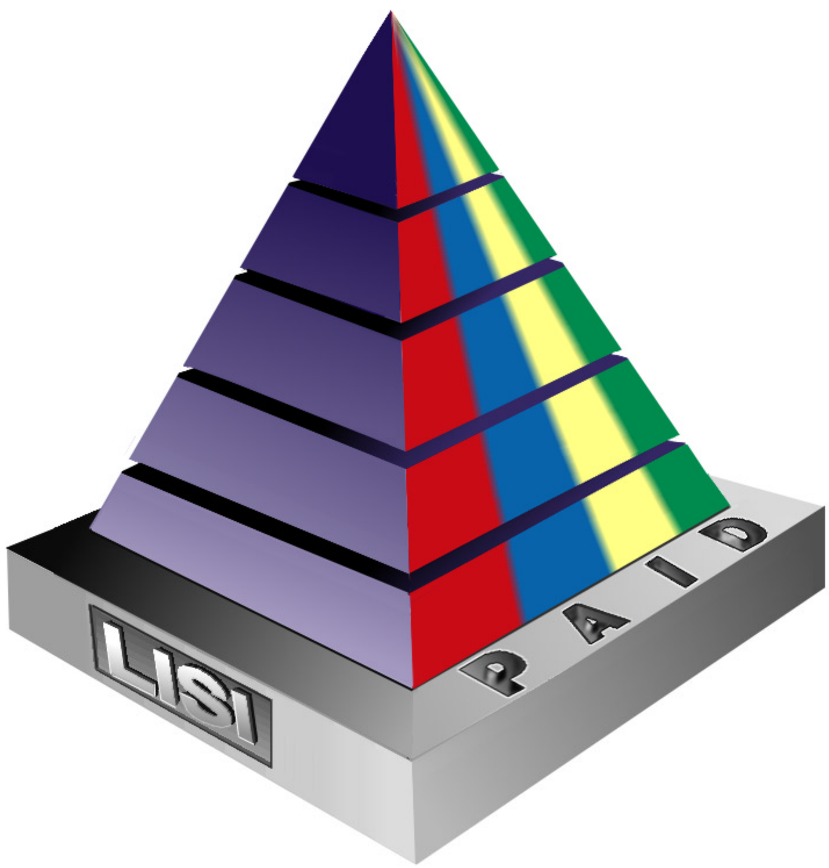
Information Type	Information Category	Definition
3. Resource Management	3.1 Resource Allocation	In a general sense, distribution of limited forces, materiel, and other assets or capabilities apportioned or allocated to the commander of a unified or specified command among competing requirements for employment. Specific allocations (e.g., air sorties, nuclear weapons, forces and transportation) are described as allocation of air sorties, nuclear weapons, etc. (JCS 1-02)
	3.2 Sustainability	A component of military capability (JCS 1-02)
	3.3 Support Services	Consumables/Repair Parts, Logistic Support Assets, Medical Facilities, Host Nation Assets
	3.4 Medical	Medical information including medical intelligence and medical threat assessment
	3.5 Personnel	Information regarding those individuals required in either a military or civilian capacity to accomplish the assigned mission.
4. Orders & Directives	4.1 Mission Plans & Orders	Information describing the broad objectives of combat actions to be carried out under the cognizance of combat action commanders
	4.2 Mission Accomplishment Status	Such as mission reporting, Commander's Estimate, OPLAN/OPORDER Execution Status, Movement of Forces, C2W Effectiveness, Intel Collection/Dissemination Status, MIW Status, Mission Order Acknowledgment, Air Defense Activity
	4.3 Conditions and Constraints	Prescriptions and proscriptions on combat actions formulated by proper authority to control operations. Conditions specify when an action may be considered to be authorized without further coordination with the imposing authority; constraints describe limitations
	4.4 Tactical Systems Interoperability	Data, requests, and orders for coordinating and controlling C4I assets including surveillance and communications.
	4.5 Tactical Orders	Short term orders and coordination for the performance of specific tasking. Includes explicit tasking, rules of engagement and guidance for decentralized command, and coordination among platforms necessary to carry out tasking.
	4.6 Tactical Employment	Information that must be known or specified in order to control the implementation of a directed action.
	4.7 Tactical Order Status	Status of engagement orders

Intentionally Blank

APPENDIX D: LEVELS OF INFORMATION SYSTEMS INTEROPERABILITY (LISI) REFERENCE MODEL

APPENDIX D

LEVELS OF INFORMATION SYSTEMS INTEROPERABILITY (LISI) REFERENCE MODEL



D.1.0 Improving Interoperability

Today, more than ever before, the primary challenge of conducting joint operations is increasingly summed up in one word: interoperability. The Joint Task Force (JTF) that fights the next conflict, small or large, does not exist until the need arises. Its approach to information management and the set of electronic information systems will be based in large part on which Service is in charge of the operation. Though all Services provide their essential sets of automated "tools," the particulars of which ones, how many, where they are located, etc., are all dependent on the situation and the decisions of the assigned Service Commander.

Determining how various systems are pulled together to accomplish a joint mission is one of the major challenges facing information systems architecture developers throughout the Department of Defense (DoD). Information systems built to meet specific Service requirements must still provide for the appropriate level of C4ISR interoperability to meet joint requirements. As such, understanding the specific nature and degree of interoperability required is a key consideration that must be accounted for when designing, constructing, and deploying any information technology architecture.

The Levels of Information Systems Interoperability (LISI) Reference Model presents a logical structure and a discipline or "maturity model" for improving interoperability incrementally between information systems. As such, LISI strengthens the ability to effectively manage information systems in context with mission effectiveness. It complements other activities that support improved use of information technology in the DoD mission, such as the Defense Information Infrastructure (DII) Master Plan, the DII Common Operating Environment (COE), the DoD Technical Reference Model (TRM), and Joint Technical Architecture (JTA).

The LISI Reference Model (LRM):

- Facilitates a common understanding of interoperability and its enablers at each level of sophistication of system-to-system interaction.

- Translates interoperability levels into requisite capabilities (procedures, applications, infrastructure, data) that form the basis for making comparisons between heterogeneous systems and for determining the degree to which system implementations conform to current DoD technical criteria.

- Builds on current DoD prescriptions to provide a methodology, maturity model, and process for assessing and improving interoperability incrementally in context with requirements analysis, systems development, acquisition, and fielding, and technology insertion.

- Provides the interoperability assessment "contribution" to the information technology "measure of performance (MOP)" called for in the ITMRA and other recent government legislation

Section 2 presents a brief overview of the LISI Reference Model. Section 3 discusses the relationships between LISI and operational, systems, and technical architecture views.

D.2.0 The LISI Reference Model

There are differing opinions across the DoD of what is meant by interoperability. Some users consider the ability to translate data into text files and exchange them using simple e-mail as “achieving” interoperability. This is one way for two systems to work together, but this restricted view leaves out many other capabilities that are needed to satisfy an operational need. LISI expands the definition of interoperability beyond the ability to move data from one system to another — it considers the ability to exchange and share services between systems. LISI focuses on increasing levels of sophistication for system-to-system interaction; i.e., thresholds of capabilities that systems exhibit as they improve their ability to interact with other systems. The specific capabilities needed to achieve each level are described in terms of four attributes – *procedures, applications, infrastructure,* and *data,* which are represented by the “PAID” acronym.

D.2.1 Orientation – Incremental Levels of Information Interactions and the Corresponding Computing Environments

The LISI Reference Model is oriented by levels that represent increasing degrees of sophistication required to accomplish interactions between information systems. The use of levels provides a discipline for describing the nature of information interaction between operational nodes, translating that nature into the suite of information system capabilities — the computing environment — necessary to support the information interaction in context with the operational need (e.g., timeliness, accuracy), and determining the implementation rules for each system capability.

A level in the LISI model is characterized by the most demanding exchanges the level embodies, as well as the enabling capabilities it requires. The LISI Reference Model defines five levels, currently numbered 0, 1, 2, 3, and 4. Figure D-1 depicts these levels.

Information Exchange	*Level*
Distributed global info. and apps. **Simultaneous interactions w/ complex data** **Advanced collaboration** e.g., Interactive COP update Event-triggered global database update	**4 -- Enterprise** Interactive manipulation Shared data & applications
Shared databases **Sophisticated collaboration** e.g., Common Operational Picture	**3 -- Domain** Shared data “Separate” applications
Heterogeneous product exchange **Group Collaboration** e.g., Exchange of annotated imagery, maps w/ overlays	**2 -- Functional** Minimal common functions Separate data & applications
Homogeneous product exchange e.g., FM voice, tactical data links, text file transfers, messages, e-mail	**1 -- Connected** Electronic connection Separate data & applications
Manual Gateway e.g., diskette, tape, hard copy exchange	**0 -- Isolated** Non-connected

Computing Environment

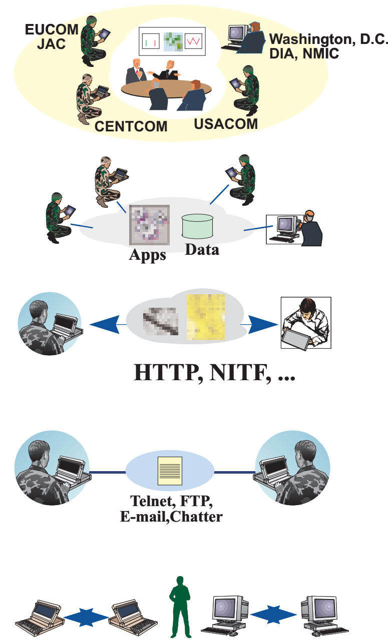


Figure D-1 LISI Levels and Corresponding Computing Environments

Each level can be generally defined as follows:

Level 0 — Isolated: Level 0 systems have no direct electronic connection. Data exchange between these systems typically occurs via either manual keyboard entry or an extractable common media format (e.g., diskette).

Level 1 — Connected: Level 1 systems are linked electronically. These systems conduct peer-to-peer exchange of homogeneous data types, such as simple “text,” e-mail, or fixed graphic files (e.g., GIF, TIFF images). Generally, level 1 systems allow decision makers to simply exchange files with one another.

Level 2 — Functional: Level 2 systems are distributed, i.e., they reside on local networks that allow complex, heterogeneous data sets (e.g., annotated images, maps with overlays) to be passed from system to system. Formal data models (logical and physical) are present; but generally, only the logical data model is agreed to across programs and each program defines its own physical data model. Generally, decision makers are able to share fused information between systems or functions.

Level 3 — Domain: Level 3 systems are integrated, i.e., capable of being connected via wide area networks (WAN) that allow multiple users to access data. Information at this level is shared between independent applications. A domain-based data model is present (logical and physical) that is understood, accepted, and implemented across a functional area or group of organizations that comprises a domain. Systems are capable of implementing business-rules and processes to facilitate direct database-to-database interactions, such as those required support database replication servers. Individual applications at this level may share central or distributed data repositories. Systems at this level support group collaboration on fused information products. Generally, decision-making is supported by fused information from a localized problem domain.

Level 4 — Enterprise: Level 4 systems are capable of operating using a distributed global information space across multiple domains. Multiple users can access and interact with complex data simultaneously. Data and applications are fully independent and can be distributed throughout this space to support information fusion. Advanced forms of collaboration (the virtual office concept) are possible. Data has a common interpretation regardless of form, and applies across the entire enterprise. The need for redundant, functionally equivalent applications is diminished since applications can be shared as readily as data at this level. Decision-making takes place in the context of, and is facilitated by, enterprise-wide information found in this global information space.

Each higher level of the LISI Reference Model represents a demonstrable increase in capabilities over the previous level of system-to-system interaction — in terms of the data transferred, the applications that act on that data, the infrastructure required, and the procedures (e.g., policies and processes) for information management.

D.2.2 Attributes — The PAID Paradigm

Many factors influence the ability of information systems to interoperate. LISI categorizes these factors into four key attributes that comprise the domain of interoperability: *Procedures, Applications, Infrastructure* and *Data*. These attributes, referred to collectively as PAID, encompass the full range of interoperability considerations. They assist in defining the sets of characteristics for the exchange of services at each level of sophistication. Consideration of all the PAID attributes is critical for moving interoperability beyond the simple connection between systems. It facilitates assessing DoD architectures by helping to identify specific interoperability gaps or weaknesses.

Figure D-2 graphically depicts the PAID paradigm, showing the range of consideration for each attribute.

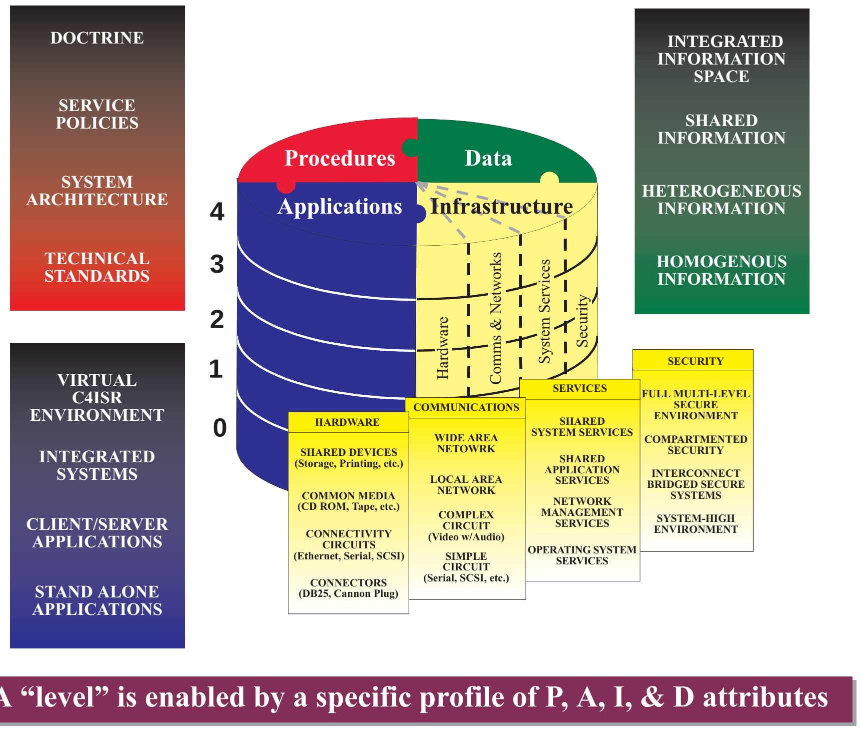


Figure D-2 The PAID Paradigm

The PAID attributes are summarized below:

- **Procedures**: focus on the many forms of guidance that impact on system interoperability, including doctrine, mission, architectures, and standards.
- **Applications**: represent the functional aspects of the system. These functions are manifest in the system's software components, from single processes to integrated applications suites.
- **Infrastructure**: defines the range of components that enable interactions between systems, including hardware, communications, system services, and security. For example, infrastructure considers the protocols, enabling software services, and supporting data structures for information flow between applications and data.
- **Data**: includes the data formats and standards that support interoperability at all levels. It embodies the entire range of styles and formats from simple text to enterprise data models.

D.2.3 THE CURRENT LISI REFERENCE MODEL

A reference model is defined as a set of concepts, entities, interfaces, and diagrams that provides common ground for comparisons. A reference model is also a valuable tool for evaluating and comparing information systems. It does not provide a specific system design, but rather it defines a common set of services and interfaces for building specific designs. For example, the DoD Technical Reference Model (DoD TRM) was developed as a framework for evaluating technical implementations and for determining DoD systems characteristics. The Joint Technical Architecture (JTA) was developed from the DoD TRM to specify technical implementations when building a system. The TRM/JTA should allow systems to incorporate and exhibit the technical characteristics that were determined as important to DoD.

The LISI Reference Model is the foundation for a similar process that focuses on the interoperability of DoD systems. The LISI Reference Model, extended to include detailed definitions of capabilities, options, and implementation criteria, can support rigorous system interoperability evaluations and comparative assessments.

The current LISI Reference Model is shown in Figure D-3. The reference model provides a framework for understanding operational information interactions in context with the technologies and system interactions required for interoperability. It defines major thresholds of operational information interaction and provides direct translation at each level to a requisite suite of information system capabilities.

The current LISI Reference Model provides a baseline of capability thresholds, described in terms of the PAID attributes. The reference model provides the common vocabulary and framework needed to discuss interoperability between systems. At each level, a word or phrase highlights the most important aspect of PAID needed to achieve that level. For example, a system targeting interactions with other systems working at Level 3 (Integrated) must build toward the specific set of capabilities listed in the LISI Reference Model for Level 3. As stated earlier, the reference model can be extended to address specific PAID capabilities, characteristics, and implementation criteria.

Nature of Operational Information Interaction	Corresponding Interoperability Level		Implications			
			P	**A**	**I**	**D**
Cross-Domain Interactive Manipulation	**Enterprise**	**4**	**Enterprise Level**	**Interactive**	Multiple Topologies	**Enterprise Model**
Shared Applications & Databases	**Domain**	**3**	**Domain Level**	**Groupware**	World Wide Networks	**Domain Model**
Complex Media Exchange	**Functional**	**2**	**Program Level**	**Desktop Automation**	Local Networks	**Program Model**
Simple Electronic Exchange	Connected	1	Local/Site Level	Standard System Drivers	Simple Connection	Local
Manual Gateway	Isolated	0	Access Control	N/A	Independent	Private

Figure D-3 The Current LISI Reference Model

D.2.3.1 Level 0

Level 0 systems need to exchange data or services, but cannot directly interoperate. The lack of direct, electronic connectivity may rest solely on differing security or access control policies.

- **Procedures** – system has locally established procedures governing access control. A user must access the system directly to share information with other systems.
- **Applications** –functionally independent in most isolated systems. The resulting data is important but the ability to consistently manipulate that data does not come into play.
- **Infrastructure** – primarily independent between systems. Most information exchange is by physical access. At most an isolated system can exchange data by common physical media such as disks or tapes.
- **Data** – private data models

D.2.3.2 Level 1

Level 1 systems have an established electronic link characterized by separate peer-to-peer connections. They can locally support simple file exchanges between systems. The types of exchanged files are typically homogeneous in context (e.g., text only, a bitmap file—GIF, TIFF).

- **Procedures** – beyond simple access control most still primarily relate to local or site level policies.
- **Applications** – independent among systems but use common drivers and interfaces such as those specified by the JTA.
- **Infrastructure** – support simple peer to peer connections to allow for local data transfer consistent with the local procedures established
- **Data** – local data models may exist, but are usually specific to a particular program. Simple reports or graphics are one example.

D.2.3.3 Level 2

Level 2 systems must be able to exchange and process complex (i.e., heterogeneous) files. These consist of items such as annotated images, maps with overlays, multi-media or hyper-linked documents. The systems are connected to multiple systems on local networks. A key capability provided by the system or applications at this level is the ability to provide web-based access data.

- **Procedures** – focus on the individual program level, COE specifies many of the implementations programs must support.
- **Applications** – functions include desktop automation and the ability to exchange some structured data. Office automation programs are one example. Web interfaces are significant.
- **Infrastructure** – systems interact with other system in the local area through LANs. These LANs may use protocols (such as TCP/IP) that support wide area networking.

- **Data –** advanced data structures may exist but they still primarily support individual applications (program data models). There is increasing commonality of data formats across programs.

D.2.3.4 Level 3

Level 3 is characterized by multiple application-to-application interactions. Systems and applications are interconnected, but generally operate on a single functional set of data (e.g., intelligence, C2, logistics). Implementations at this level usually have only a "localized" view of the distributed information space and cross only one operational or functional domain.

- **Procedures –** focus on domain interaction where a domain may span many geographic areas but is focused on one functional area (C2, intelligence, logistics).
- **Applications –**advanced beyond individual programs, basic group collaboration capability is supported such as tracking revisions in documents, or workflow management.
- **Infrastructure** – networks are global. At this level interaction takes place in parts of the global information space, though not all of it.
- **Data** – defined data models exist and are understood between applications, however they only represent a particular domain (MIDB, etc.).

D.2.3.5 Level 4

Level 4 is the ultimate goal of information systems seeking interoperability across functional activities and informational domains (Intelligence, C2, Logistics, etc.). At this enterprise level, information is shared globally through a distributed information architecture. Applications and systems operate as necessary across all the functional data domains. The "virtual" workspace uses shared applications operating against an integrated information space. This level represents the capabilities necessary to achieve concepts proposed in DoD's "Joint Vision 2010" documents.

- **Procedures –** enterprise level Joint/DoD procedures, based on enterprise level understandings of tasks such as the UJTL.
- **Applications –** integrated into the common distributed information space. Multiple users can access the same instances of enterprise wide data.
- **Infrastructure** – global networks that support multi-dimensional topologies. These networks may have different areas based on security or access control, but they are integrated appropriately to support the users needs. Current efforts to support Secret and Below Interoperability (SABI) and guards or filters that support multiple security levels are examples of this infrastructure.
- **Data –** enterprise data models support the integration of applications. There is a common understanding of the data across the enterprise.

D.3.0 LISI Relationship to C4ISR Architectures

The LISI Reference Model provides sufficient information to support architecture development and linkages between the operational, systems, and technical architecture views. The operational architecture view provides details about the required needline interactions between organizational nodes to determine the specific interoperability level required. Even before looking at systems or technical details, the particular details regarding who is exchanging information and what is the nature of the information exchanged enables a "table lookup" to the LISI Reference Model to identify the required interoperability level. For example, voice interaction between two low-level organizations requires a different interoperability level than multiple enterprise-level organizations that must collaborate on a multimedia product. The LISI Reference Model helps to frame the need for interoperability in specific and meaningful terms that can guide systems acquisition and design decisions.

In recent years, DoD has steadily enhanced its information technology architecture guidelines and tools. The DoD architectural community has produced an interrelated set of policies and guidance, including the TRM, the JTA, the DII COE and the *C4ISR Framework*. By defining the interoperability relationships DoD seeks between systems, LISI becomes an integral part of these guidelines. Specifically, the LISI Reference Model is designed to support the development and analysis of DoD architectures by helping to identify, up-front, issues, problems, gaps, and shortfalls that may be present within any information technology architecture.

D.3.1 Operational Architecture View

In an operational architecture view, the needlines that connect operational nodes represent interoperability requirements. Use of the LISI Reference Model begins with the *Operational Node Connectivity Description* and the articulation of each operational information interchange (e.g., "transfer target folders to support target selection within 15 minutes"). The operational requirement is then further defined in terms of the nature of the information interchange (e.g., "transfer maps, annotated images, text, and graphics"). Based on the nature of the required information interchange and the operational performance parameters that need to be met for mission accomplishment, each needline is labeled with an interoperability level requirement via LISI Reference Model table lookup. This requirement forms the basis for assessing existing or candidate information systems supporting the needline.

D.3.2 Systems Architecture View

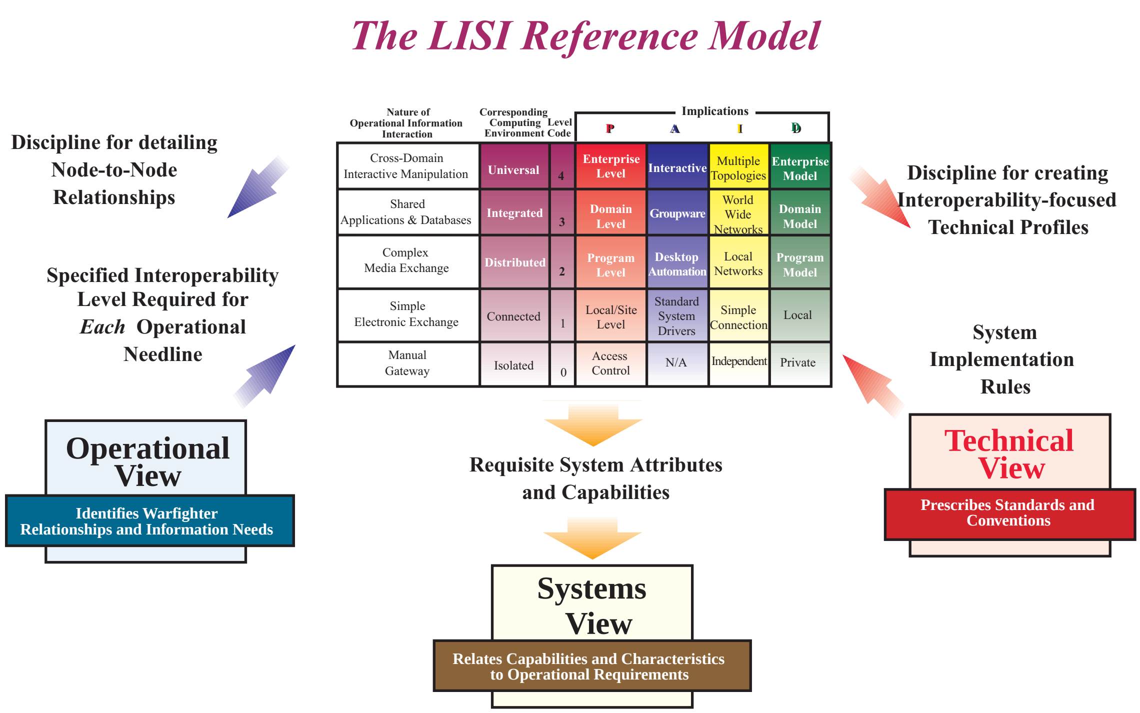
Application of the LISI Reference Model to the systems architecture view begins with the *Systems Node Connectivity Description*, and supplements the operational architecture view by depicting the system-to-operational node assignments. Based on the level of interoperability to be achieved, the LISI Reference Model and its extensions can be used to penetrate the requisite PAID capabilities and characteristics.

D.3.3 Technical Architecture View

Application of the LISI Reference Model to the technical architecture view begins with the *Technical Architecture Profile*. Based on the system PAID capabilities and characteristics (identified in the *Systems Node Connectivity Description*) the LISI Reference Model provides a convenient construct for interoperability-focused cross-walks to existing implementation requirements and mandates (e.g., JTA, DII-COE, …).

D.3.4 Cross-View Relationships

Figure D-4 outlines the relationships between the LISI Reference Model and the operational, systems, and technical architecture views. In summary, the operational architecture view describes the interoperability requirement – the LISI model relates that requirement to a specific interoperability level. The systems architecture view depicts the system-to-node assignments – the LISI model provides a means for identifying the systems' capabilities in context with the capabilities necessary to meet the required interoperability level. The technical architecture view profiles the implementation rules for the requisite system capabilities – the LISI model provides a means for articulating the applicable rules sets (e.g., JTA) in context with the suite of capabilities defined by the interoperability level.



Nature of Operational Information Interaction	Corresponding Computing Environment	Level Code	Implications: P	A	I	D
Cross-Domain Interactive Manipulation	Universal	4	Enterprise Level	Interactive	Multiple Topologies	Enterprise Model
Shared Applications & Databases	Integrated	3	Domain Level	Groupware	World Wide Networks	Domain Model
Complex Media Exchange	Distributed	2	Program Level	Desktop Automation	Local Networks	Program Model
Simple Electronic Exchange	Connected	1	Local/Site Level	Standard System Drivers	Simple Connection	Local
Manual Gateway	Isolated	0	Access Control	N/A	Independent	Private

Figure D-4 LISI Relationship to Architecture Views

D.4.0 Summary

The principles of information system interoperability extend beyond just architecture planning to include activities such as system acquisition, technical design, implementation, and certification. LISI extends to all of these by considering the increasing levels of sophistication for system-to-system interaction; i.e., the thresholds of capabilities that systems exhibit as they improve their ability to interact with each other. The LISI Reference Model provides an accepted representation of system interoperability, including a common vocabulary that allows agreement on standards for facilitating interoperability in terms of the PAID paradigm. The LISI Reference Model also provides automated methods for conducting interoperability assessments and for deriving performance metrics based on operational testing and evaluation. Finally, the reference model serves as a process that can be used for analyzing and establishing cooperative interoperability agreements within and among communities of interest.

For more information concerning the LISI Reference Model, its use for evaluating architectures, applicability to the acquisition process, and its relationship to the test and evaluation community refer to the Architecture Working Group Final Report.

Intentionally Blank

APPENDIX E: GENERIC DOD TECHNICAL REFERENCE MODEL

APPENDIX E

GENERIC DOD TECHNICAL REFERENCE MODEL

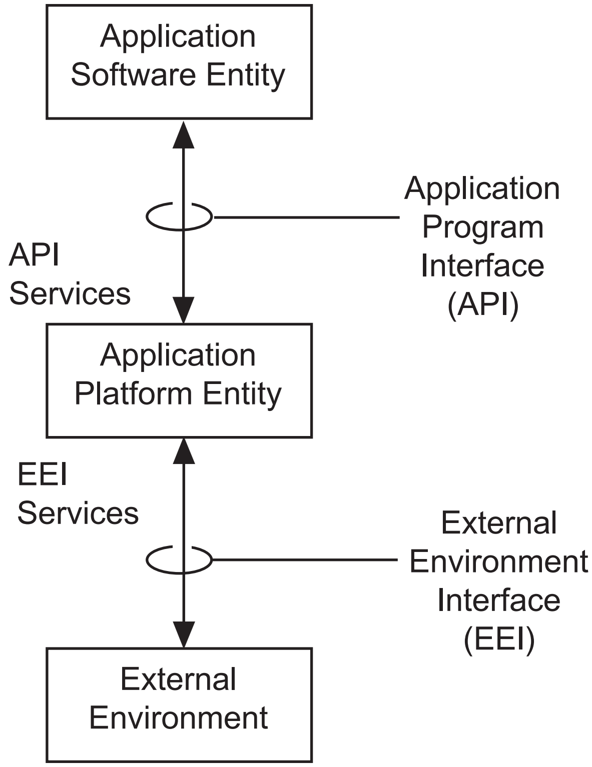
The generic DoD Technical Reference Model is a set of concepts, entities, interfaces, and diagrams that provides a basis for the specification of standards. To a large extent, the Technical Reference Model adopts the foundation work of the IEEE POSIX P1003.0 Working Group as reflected in their Guide to the POSIX Open System Environment (POSIX.0). Within the guide, an interface is defined as “a shared boundary between the two functional units.” The functional units are referred to as “entities” when discussing the classification of items related to application portability.

The basic elements of the generic DoD Technical Reference Model are those identified in the POSIX Open System Reference Model and are presented in Figure E-1. As shown in the figure, the model includes three classes of entities and two types of interfaces as follows:

- Application Software Entity
- Application Program Interface (API)
- Application Platform Entity
- External Environment Interface (EEI)
- External Environment.

This model has been generalized to such a degree that it can accommodate a wide variety of general and special purpose systems.

From the perspective of the application software entity, these services are provided by an application platform whether the particular services are provided from the local platform or from remote platforms that may comprise one or more nodes of a larger distributed system. Volume 3 of the TAFIM explains how this generic model can be applied in a distributed environment.



Reference: IEEE Draft Guide to the POSIX Open System Environment, June 1992

Figure E-1. Generic DoD Technical Reference Model

E.1 Application Software Entity

In the past, custom systems were developed for specific hardware platforms using proprietary systems software (e.g., operating system, text editor, file management utilities). Such customization was necessary because Government requirements were often more localized than those of the commercial marketplace. These systems were not designed to interoperate with other systems nor to be portable to other hardware platforms. In addition, different systems were developed to perform similar functions at different levels of the overall DoD organization (national, theater, and unit) and for the different Services, (Army, Navy, Air Force, Marine Corps). As a result, many of the systems that were developed included functions redundant with those of other applications. This situation often hindered systems evolution toward greater interoperability, data sharing, portability, and software reuse.

The Technical Reference Model promotes the goals of developing modular applications and promoting software reuse to support the broad range of activities that are integral to any organization. To satisfy these goals, functional (mission-area) applications development will, in many respects, become an integration activity as much as a development activity. Application development will likely be accomplished by dividing and/or consolidating common functional requirements into discrete modules. Previously developed reusable code or Government-off-the-shelf (GOTS) applications that could satisfy some, if not all, of the new functional requirements would be identified. Such reusable code/applications would then be integrated, to the extent possible, to become the software pieces necessary to complete the mission and/or support applications that will satisfy all of the requirements.

In the Technical Reference Model, applications are divided into mission area applications and support applications. A common set of support applications forms the basis for the development of mission-area applications. Mission-area applications should be designed and developed to access this set of common support applications. As explained in Volume 3, APIs are also used to define the interfaces between mission-area applications and support applications.

E.2 Application Program Interface

The API is defined as the interface between the application software and the application platform across which all services are provided. It is defined primarily in support of application portability, but system and application software interoperability also are supported via the communication services API and the information services API. The API specifies a complete interface between the application and the underlying application platform and may be divided into the following groups:

- System Services API (including APIs for Software Engineering Services and Operating System Services)
- Communications Services API (including APIs for Network Services)
- Information Services API (including APIs for Data Management Services and Data Interchange Services)
- Human/Computer Interaction Services API (including APIs for User Interface Services and Graphics Services).

The first API group, System Services, is required to provide access to services associated with the application platform internal resources. The last three API groups (Communications Services, Information Services, and Human/Computer Interaction Services) are required to provide the application software with access to services associated with each of the external environment entities. APIs for services that cut across the areas are included among all groups where applicable.

A standardized API should be used for accessing security mechanisms. The use of the operating system kernel for maintaining separation among processes executing at different security levels means that this API would be included in the System Services API category above. Such an API will promote independence of security services and security mechanisms, offering transparency to users and applications. This independence will allow different security mechanisms to be accommodated at various stages in an information system life cycle.

E.3 Application Platform Entity

The Application Platform is defined as the set of resources that support the services on which application software will execute. It provides services at its interfaces that, as much as possible, make the implementation-specific characteristics of the platform transparent to the application software.

To assure system integrity and consistency, application software entities competing for application platform resources must access all resources via service requests across the API. Examples of application platform services may include an operating system kernel, a realtime monitor program, and all hardware and peripheral drivers.

The application platform concept does not imply or constrain any specific implementation beyond the basic requirement to supply services at the interfaces. For example, the platform might be a single processor shared by a group of applications, a multiprocessor at a single node, or it might be a large distributed system with each application dedicated to a single processor.

The application platform implementations that use the Technical Reference Model may differ greatly depending upon the requirements of the system and its intended use. It is expected that application platforms defined to be consistent with the Technical Reference Model will not necessarily provide all the features discussed here, but will use tailored subsets for a particular set of application software.

E.4 External Environment Interface

The External Environment Interface (EEI) is the interface between the application platform and the external environment across which information is exchanged. It is defined primarily in support of system and application software interoperability. User and data portability are directly provided by the EEI, but application software portability also is indirectly supported by reference to common concepts linking specifications at both API and EEI. The EEI specifies a complete interface between the application platform and the underlying external environment, and may be divided into the following groups:

- Human/Computer Interaction Services EEI
- Information Services EEI
- Communications Services EEI.

The Human/Computer Interaction (HCI) Services EEI is the boundary across which physical interaction between the human being and the application platform takes place. Examples of this type of interface include CRT displays, keyboards, mice, and audio input/output devices. Standardization at this interface will allow users to access the services of compliant systems without costly retraining.

The Information Services EEI defines a boundary across which external, persistent storage service is provided, where only the format and syntax are required to be specified for data portability and interoperability.

The Communications Services EEI provides access to services for interaction between application software entities and entities external to the application platform, such as application software entities on other application platforms, external data transport facilities, and devices. The services provided are those where protocol state, syntax, and format all must be standardized for application interoperability.

Security mechanisms to provide for security services in EEIs will be implemented similarly to those required for communications among distributed platforms. That is, the EEIs facilitate communications among distributed platforms. Such implementations will occur primarily in the cross-platform service areas of security and system management.

E.5 External Environment

The External Environment contains the external entities with which the application platform exchanges information. These entities are classified into the general categories of human users, information interchange entities, and communications entities. Human users are not further classified, but are treated as an abstract, or average person. Information interchange entities include, for example, removable disk packs and floppy disks. Communications entities include telephone lines, local area networks, cabling, and packet switching equipment.

Doctrinal mechanisms (physical, administrative, and personnel) will provide for required security protection of information system components in the external environment.