Homeland Security

Public Safety Architecture Framework

The SAFECOM Program

Department of Homeland Security

Volume II: Product Descriptions

Version 1.0

February 10, 2006

SAFECOM

SAFECOM

Background on Public Safety and Wireless Communications

Inadequate and unreliable wireless communications have been issues plaguing public safety organizations for decades. In many cases, agencies cannot perform their mission-critical duties. These agencies are unable to share vital voice or data information via radio with other jurisdictions in day-to-day operations and in emergency response to incidents including acts of terrorism and natural disasters.

According to a report published by the National Task Force on Interoperability (February 2003), the public safety community has identified the following key issues that hamper public safety wireless communications:

- Incompatible and aging communications equipment
- Limited and fragmented radio spectrum
- Limited and fragmented budget cycles and funding
- Limited equipment standards
- Limited and fragmented planning and coordination

In short, the Nation is heavily invested in an existing infrastructure that is largely incompatible. The SAFECOM Program was established by the Office of Management & Budget and approved by the President's Management Council to address the public safety communications issues identified above.

The SAFECOM Program

SAFECOM, a communications program of the Department of Homeland Security's (DHS) Office for Interoperability and Compatibility (OIC), with its federal partners, provides research, development, testing and evaluation, guidance, tools, and templates on communications-related issues to local, tribal, state, and federal public safety agencies. The OIC is managed by the Science and Technology (S&T) Directorate's Office of Systems Engineering and Development (SED).

SAFECOM, a public safety practitioner-driven program, is working with existing federal communications initiatives and key public safety stakeholders to address the need for better technologies and processes for the cross-jurisdictional and multi-disciplinary coordination of existing systems and future networks. SAFECOM harnesses diverse federal resources in service of the public safety community. The scope of this community is broad, including more than 60,000 local and state public safety agencies and organizations. Federal customers include agencies engaged in public safety disciplines such as law enforcement, firefighting, public health, and disaster recovery, as well as federal agencies that provide funding and support to state and local public safety agencies. SAFECOM makes it possible for the public safety community to leverage resources by promoting coordination and cooperation across all levels of government.

SAFECOM's Near-Term Initiatives

- Develop a process to advance standards that improve public safety communications interoperability
- Develop and disseminate grant guidance for all agencies providing grants for public safety communications and interoperability
- Provide tools and models for communications and interoperability training and technical assistance
- Create a one-stop shop for public safety communications and interoperability
- Develop, test, and evaluate technologies for public safety communications and interoperability

SAFECOM'S Long-Term Goals

- Achieve a systems-of-systems environment supported by communications standards, tools, and best practices
- Facilitate coordination of funding assistance through tailored grant guidance to maximize limited resources available for public safety communications and interoperability
- Pilot tools and methods as national models for public safety at the rural, urban, state, and/or regional levels
- Provide policy recommendations to promote efficiency in public safety communications

Publication Notice

Abstract

This document provides definitions, guidelines, and uses of the Public Safety Architecture Framework (PSAF).

The PSAF supports the development of interoperable and interactive architectures for public safety organizations. It uses a structured approach and common methodologies for defining and resolving wireless communications interoperability challenges related to public safety.

Change Log

Version	Date	Changes
1.0 Draft	April 5, 2005	Initial Draft Document
1.0 Draft	June 2, 2005	Edited text and organization
1.0 Draft	December 15, 2005	Minor text additions and edits as proposed by the PSAF working group and the executive level of SAFECOM
1.0	February 10, 2006	Text and format edits.

ACKNOWLEDGEMENTS

The SAFECOM program extends its appreciation to the many public safety practitioners, individuals, and government organizations that directly contributed to the creation of the PSAF.

Contact Information

Please address comments or questions to: The SAFECOM Program
P.O. Box 57243
Washington, D.C. 20073

Telephone: 1-866-969-SAFE
Program E-mail: safecom@dhs.gov

Director: David Boyd, Ph.D.

Executive Summary

The SAFECOM process for identifying and developing standards began with the Public Safety Statement of Requirements (PS SoR). Those requirements are driving the vision for a migration from current *as-is* architectures to the future *to-be* interoperable public safety communications enterprise architecture. The Public Safety Architecture Framework (PSAF) provides an industry-validated enterprise architecture methodology to plan and develop the migration from current public safety architectures to the interoperable systems outlined in the PS SoR.

Three key living documents describe and reflect the PSAF methodology:

- *PSAF Volume I* provides definitions, guidelines, and related background material.
- *PSAF Volume II* contains detailed descriptions of the three PSAF views and the products that create each of the views.
- *PSAF Volume III* will document procedures for using the methodology outlined in *PSAF Volume I* and *PSAF Volume II* upon development of a supporting PSAF tool vetted with the practitioner community. Note that *PSAF Volume III* will likely be a user guide, although for simplicity it is referred to here as *PSAF Volume III*.

PSAF Volume I and *PSAF Volume II* draw upon the organization and discussion of architecture principles and concepts published in Department of Defense Architecture Framework (DoDAF) documents.[1]

The PSAF documents will evolve as public safety provides additional input and as lessons are learned through application in the field. Although the fundamental approach will remain the same, the PSAF documents will be modified as necessary to ensure the usefulness of the PSAF across multiple disparate agencies. Lessons learned during development and piloting of the PSAF methodology may result in updates to Volume I & II. As the PSAF evolves, best practices will be developed to support a variety of applications including interoperability analysis, gap analysis, systems planning, systems migration, business case development, and RFP development.

Audience

PSAF Volume I and *PSAF Volume II* describe process architecture goals to technology architects and engineers tasked with implementing public safety wireless communications networks. While volumes I and II are not meant for end-user public safety practitioners, future documentation will target end-user practitioner needs.

Structured Process Concepts

The term *architecture framework* (AF) may not be as commonly known as *enterprise architecture* (EA), yet both concepts are associated with the same structured process that industry and government around the globe use to accomplish mission goals and save resources. While the concept of an enterprise

[1]. DoD Architecture Framework Working Group, *DoD Architecture Framework Version 1.0*, "Volume I: Definitions and Guidelines," and "Volume II: Product Descriptions," February 2004.

architecture has its roots in the information technology (IT) world, it is also fits the voice, data, and video applications of the public safety wireless communications world.

Enterprise Architecture Successes

The Enterprise Architecture Interest Group (www.eaig.org) points out impressive EA implementation successes by such companies as Volkswagen of America, Disney, Best Buy, GM, and Swiss Mobile. At the same time, the General Accountability Office (GAO) has, for over a decade, promoted the creation of EAs through the use of AFs. The GAO recognizes that AFs can clarify and help optimize the interdependencies and relationships between business operations, the underlying infrastructure, and the supporting applications across a large federated organization.[2] The Office of Management and Budget (OMB), Federal Enterprise Architecture Program Management Office, and Federal Departments and Agencies have concurred with this assessment and are actively undertaking EA planning and implementation. It is logical that public safety apply the same structured approach that uses a common methodology provided by an AF, to produce the Public Safety Architecture Framework (PSAF) for defining and resolving large-scale interoperability challenges.

Structured Approach to Interoperability

The architecture framework outlines "what" the overall structured approach is for assisting interoperability and, through the details of this structure, indicates "how" the architecture and its components will operate through the development of interface standards. In short, the PSAF provides rules and guidance for developing and presenting architecture descriptions.

The PSAF provides the following three perspectives (or views) of public safety communications and information systems. The combination of these three views form a comprehensive architecture description.

- The Operational View (OV) — Shows how public safety performs its mission
- The Systems View (SV) — Shows the systems of equipment and the flows of information that support public safety
- The Technical Standards View (TV) — Shows the technical rules and guidelines that allow these systems to interoperate.

The PSAF supports the development of interoperating and interacting architectures. It defines the preceding three related views of architecture: OV, SV, and TV. Each view is composed of sets of architecture data elements that are depicted via graphic, tabular, or textual products. The PSAF also clearly defines the relationships between these architectural views and the data elements they contain.

Quick Model Analysis

By using the PSAF to develop architectural models, you can perform a swift, simple, and automated analysis to determine if the communication systems of two public safety agencies in the same city, or two different counties, for example, can interoperate. You apply the PSAF in the same way to both

[2] GAO-04-798T, "The Federal Enterprise Architecture and Agencies' Enterprise Architectures are Still Maturing," May 19, 2004.

organizations to create common architectural descriptions of each. This provides an accurate comparison of organizations. If you determine the two systems are non-interoperable, the PSAF will also identify the interfaces that need further standards development.

While the PSAF will greatly assist the standards process related to communications interoperability by focusing on interfaces, it will not dictate specific technical solutions. This limitation will allow public safety to later consider creative and competing alternative architectures, as the PSAF is applied to various technologies.

The PSAF is a necessary step in identifying gaps in public safety needs and is therefore necessary to identify where standards need to be developed. As such, the PSAF, and the standards, will be carefully vetted by practitioners within the public safety governance structure.

Version 1.0 of the PSAF defines a common public safety architecture development, presentation, and integration approach for mission-critical operations as well as business operations and processes. The intent of the PSAF is to ensure the comparison and relation of architecture descriptions across organizational boundaries, including jurisdictional and first responder discipline boundaries.

This document applies to architectures developed by and for fire response organizations, emergency medical, and law enforcement agencies and services. In addition, any agency that needs to integrate with public safety agencies will find it useful to apply the PSAF.

Organization of this Volume

Volume II includes the following sections:

Section 1 **Introduction** describes the meaning and intention of the PSAF, the audience of this document, and the structure of product definitions as well as architecture data element tables.

(See Section 1, "Introduction.")

Section 2 **Framework Architecture Data Element Relationships** identifies the relationships between the three views of the PSAF: Operational View (OV), Systems View (SV), and Technical Standards View (TV). It describes architecture data element relationships, and system elements that map to standards.

(See Section 2, "Framework Architecture Data Element Relationships.")

Section 3 **Architecture Product Development** describes product development methodologies. The discussion includes architecture product development, product templates, and data element relationships.

(See Section 3, "Architecture Product Development.")

Section 4	**All-Views Products** identifies the products that provide executive-level summary information and definitions of terms used in a given architecture. Discusses summary information and an integrated dictionary. (See Section 4, "All-Views Products.")
Section 5	**Operational View Products** identifies the products that define the tasks and activities, operational elements, and information exchanges required to conduct operations. Provides product definitions and descriptions, as well as data element definitions. (See Section 5, "Operational View Products.")
Section 6	**Systems View Products** identifies the products that define the functional aspects of the systems and interconnections architectures supporting public safety functions. Provides product definitions and descriptions, as well as data element definitions. (See Section 6, "Systems View Products.")
Section 7	**Technical Standards View Products** identifies the products that define the technical systems implementation standards upon which engineering specifications are based, common building blocks are established, and product lines are developed. Provides a technical standards profile and technical standards forecast. (See Section 7, "Technical Standards View Products.")
Appendix A	**Glossary of Acronyms** lists the terminology and acronyms used in this document. (See Appendix A, "Glossary of Acronyms.")
Appendix B	**Dictionary of Terms** developed at the SAFECOM-AGILE-NIST (National Institute of Standards and Technology) Summit on Interoperable Communications for Public Safety. (See Appendix B, "Dictionary of Terms.")
Appendix C	**Dictionary of UML Terms** identifies Universal Modeling Language terms used in this document. UML is a graphical language for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system. (See Appendix C, "Dictionary of UML Terms.")
Appendix D	**References** identifies the prior publications referenced in this document. (See Appendix D, "References.")

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Purpose and Intended Audience

This second volume of the PSAF provides a detailed description of each PSAF product. This volume is organized with various readers in mind.

For the manager who makes acquisition, budget, or resource decisions for architecture development projects, each product section provides product definition and product purpose information to:

- Help these managers understand the architecture components or products
- Provide an appreciation of the potential level of effort in developing architectures
- Assist in discerning the potential uses of an architecture

For the architect and engineering team that needs to develop architecture products to support analysis by high-level decision makers, each product section provides a detailed product description and an architecture data element table to:

- Help identify the architecture products to include based on the architecture's intended use
- Determine architecture data needs
- Identify sources for the architecture data
- Analyze and relate the architecture data gathered
- Compose the architecture data into architecture products

1.2 Product Definitions Structure

This volume presents each view in its own top-level section that lists products and provides:

- A product overview for managers (*what* is the product)
- A brief statement on the purpose of the product for managers and architects (*why* is it useful)
- A detailed description for architects and engineers who are developing architectures, including:
    - One or more generic templates or examples. For most products, one or more generic templates are shown to illustrate the basic format of the product; when a generic template is not appropriate, one or more examples are shown.
    - A table listing definitions of the architecture data elements and their attributes, relationships between these architecture data elements, and relationships of these architecture data elements to architecture data elements in other products
    - Several Federal policies related to the development of architecture descriptions are worth noting. The next sections highlight aspects of these policies relating to architectures.

1.3 Architecture Data Element Tables Structure

An architecture data element table for each product provides definitions of the metadata — the architecture data types that comprise the products. For each architecture data type, attribute definitions detail its characteristics.

The architecture data elements provide structure for storing architecture data about a given architecture. The product should capture and store them in the Integrated Dictionary. Not all architecture data elements are applicable to every architecture description. However, architecture data elements marked with an asterisk (*) are essential for products developed as part of an integrated architecture. In general, a minimally integrated architecture consists of the following set of products:

- Overview and Summary Information (AV-1)
- Integrated Dictionary (AV-2)
- Operational Node Connectivity Description (OV-2)
- Operational Activity Model (OV-5)
- Systems Interface Description (SV-1)
- Technical Standards Profile (TV-1)

For additional products, your architectural development team should include architecture data elements marked with an asterisk (*) if you choose the product for development as part of an integrated architecture effort. Chapter 3 in *PSAF Volume I* provides a use matrix that offers use-based guidelines on which products to develop.

To assist architects and engineers in developing products, this volume organizes the architecture data element tables by architecture data element categories, as follows:

- Graphical Box Types — Architecture data elements expressed by icons shown in the product graphic.
- Graphical Arrow Types — Architecture data elements expressed by lines shown in the product graphic.
- Non-Graphical Types — Architecture data elements expressed by textual labels, or implied architecture data elements not explicit in the product graphic but 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 their containment or their subordinate relationships.
- Referenced Types — Architecture data elements defined in other products and related to the architecture data elements in the current product.
- Relationships — Architecture data elements that define relationships among architecture data elements from the preceding four categories.

2 Framework Architecture Data Element Relationships

2.1 Overview

The three views (OV, SV, and TV) of the PSAF provide different areas of emphasis into a single, integrated architecture, including defined work processes and supporting IT. Logical linkages among the architecture data elements underlying the products and the views ensure that the single architecture can actually be built and operated. In particular, these linkages ensure the architecture remains mutually consistent. The linkages provide traceability from view to view, from product to product within a view, and across views. The linkages ensure:

- Integration of systems within a family of systems (FoS), or system of systems (SoS)
- Alignment of IT functionality to mission and operational needs
- Relationships between current and future systems to current and future standards

The next sections discuss these linkages by summarizing the major relationships among the various PSAF products.

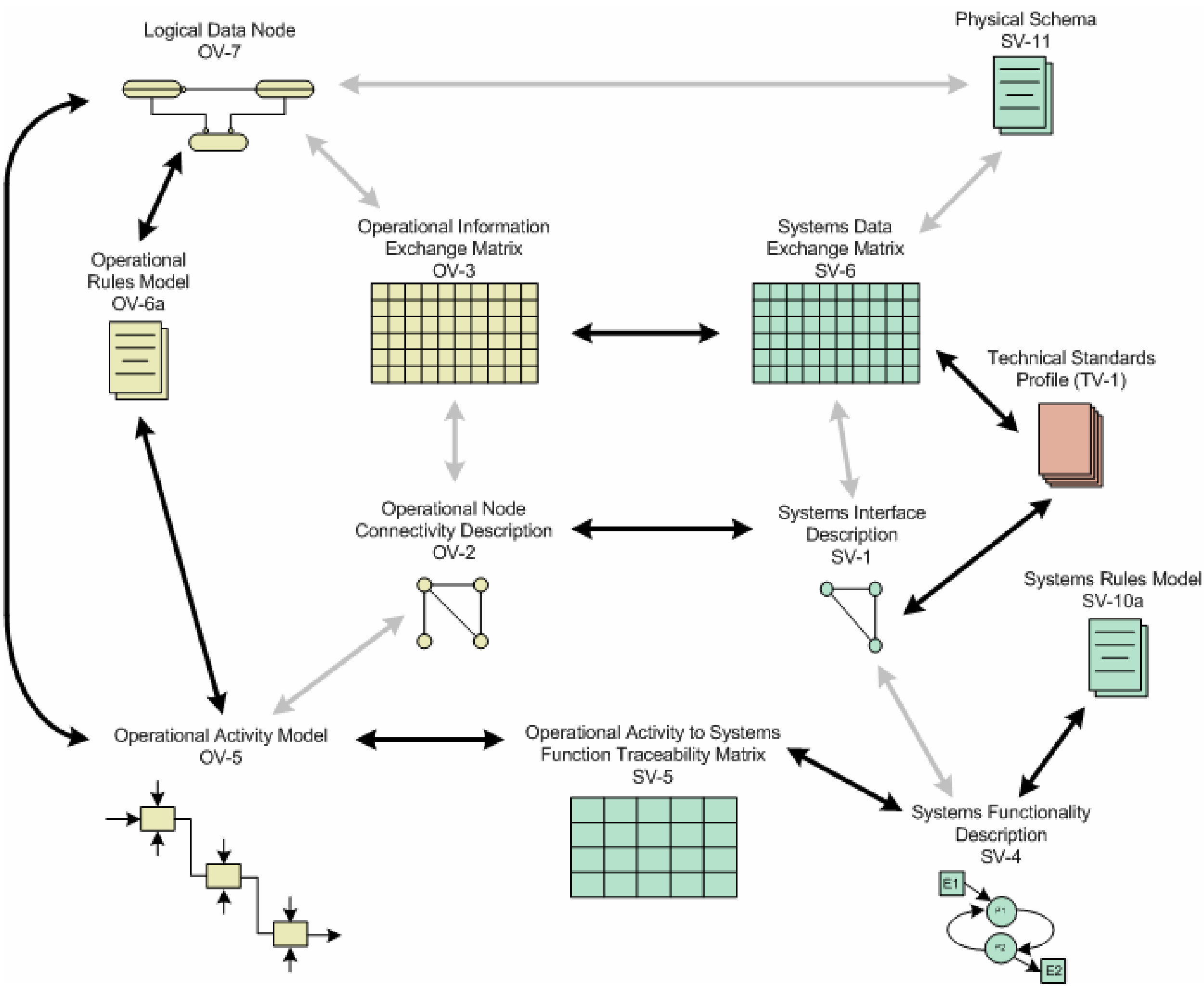
2.2 Architecture Data Element Relationships Across Views

The key to maintaining architecture product interoperability is the preservation of meaning and relationships during architecture data reuse. Figure 1 summarizes major relationships among the OV, SV, and TV architecture products. Major architecture data element relationships (belonging to these products) include:

- Operational nodes in OV-2 map to operational activities of OV-5, which are conducted by these operational nodes.
- Needlines in OV-2 map to information exchanges in OV-3, which detail the information exchanges between these operational nodes. (A needline is the logical expression of the need to transfer information among nodes.)
- OV-3 is constructed from system data elements described in OV-7.
- OV-6a contains structural and validity rules constraining OV-5.
- OV-6a also contains structural and validity rules constraining OV-7.
- SV-10a contains structural and validity rules constraining SV-4.
- Operational nodes in OV-2 map to the systems in SV-1, which support these operational nodes by providing automated support.
- Needlines in OV-2 map to interfaces in SV-1, which represent an automated needline.
- Interfaces in SV-1 map to system data elements exchanged in SV-6, which in turn map to system data flows in SV-4.
- System functions in SV-4 map to the systems in SV-1, which execute these functions.

- System functions in SV-4 put into effect automated portions of the operational activities in OV-5. SV-5 documents this relationship.
- System data exchanges of SV-6 map to the information exchanges of OV-3. System data exchanges constitute the automated portions of the operational information exchanges.
- SV-11 puts into effect and details the elements of OV-7.
- TV-1 documents standards that constrain SV-1, SV-2, SV-4, SV-6, OV-7, and SV-11.
- SV-8 documents system evolution timelines and milestones that have corresponding time periods associated with timed technology forecasts in SV-9 and timed standard forecasts in TV-2.

Figure 1: Major Product Relationships

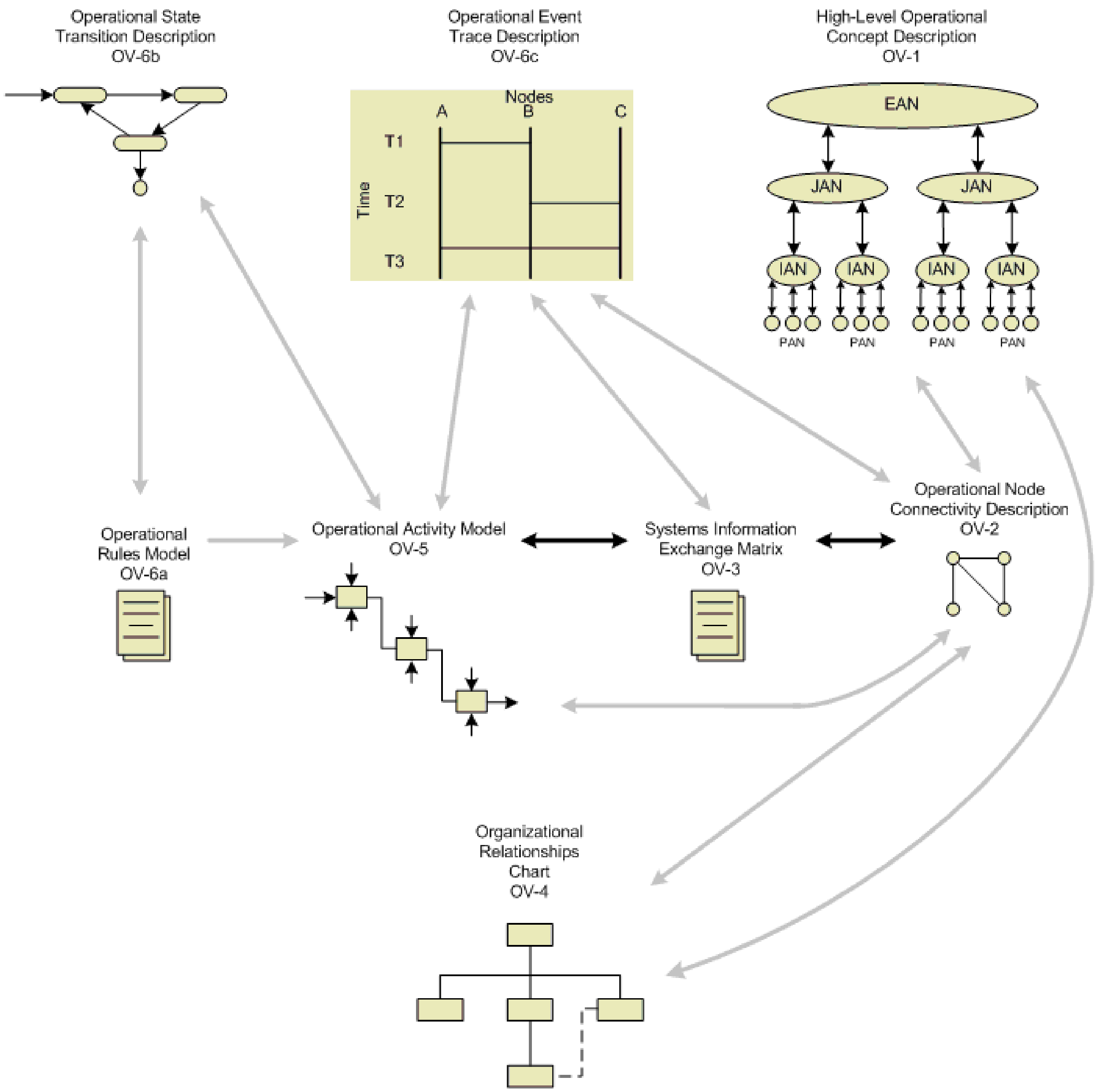


2.3 Operational View Relationships

Figure 2 covers relationships among products describing operations in the architecture's OV. Some relationships in Figure 2 may not appear in the next list because the have already been stated in Section 2.2. Figure 2 summarizes the following major architecture data element relationships:

- Organizations in OV-1 should match those in OV-2 and annotations identifying responsible operational nodes in OV-5.
- Operational nodes in OV-2 match the lifelines of OV-6c.
- Organizations in OV-4 map to the operational nodes in OV-2.
- Information elements of OV-3 map to the inputs and outputs (I/Os) that belong to activities of OV-5 conducted across two or more operational nodes of OV-2.
- Events in OV-6b map to events in OV-6c.
- State transitions in OV-6b and events in OV-6c should be consistent with activities in OV-5.
- Dynamic rules in OV-6a may constrain state transitions in OV-6b or decision points in OV-5.
- Events in OV-6b and OV-6c map to triggering events in OV-3.

Figure 2: Operational View Product Relationships



2.4 Systems View Relationships

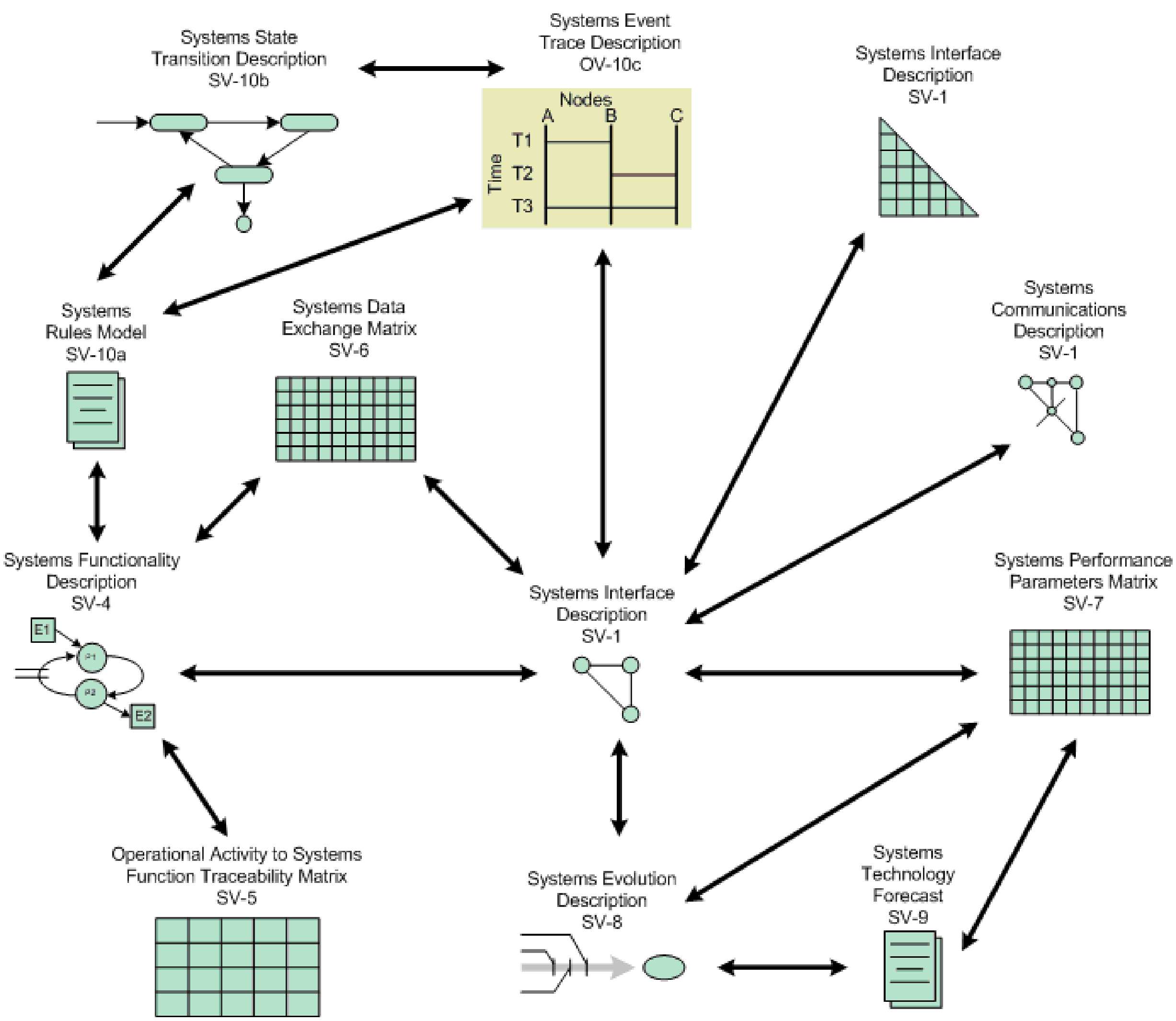
Major SV architecture product and system data element relationships include:

- Interfaces in SV-1 map to system data exchanges in SV-6.
- System data input or output by system functions in SV-4 map to system data elements of SV-6.
- Interfaces in SV-3, communications links in SV-2, and some performance requirements in SV-7 map to interfaces in SV-1.

- SV-8 phases in interface requirements and implementation in SV-1.
- SV-8 phases in the applicability of performance requirements in SV-7 and the availability of new products and technologies in SV-9.
- Rules in SV-10a constrain transitions in the SV-10b, events in SV-10c, and functions in SV-4.
- Events in SV-10b map to events in SV-10c.
- Lifelines in SV-10c map to systems in SV-1 and to system functions in SV-4.

Figure 3 summarizes relationships (does not show all relationships) among SV products.

Figure 3: Systems View Product Relationships



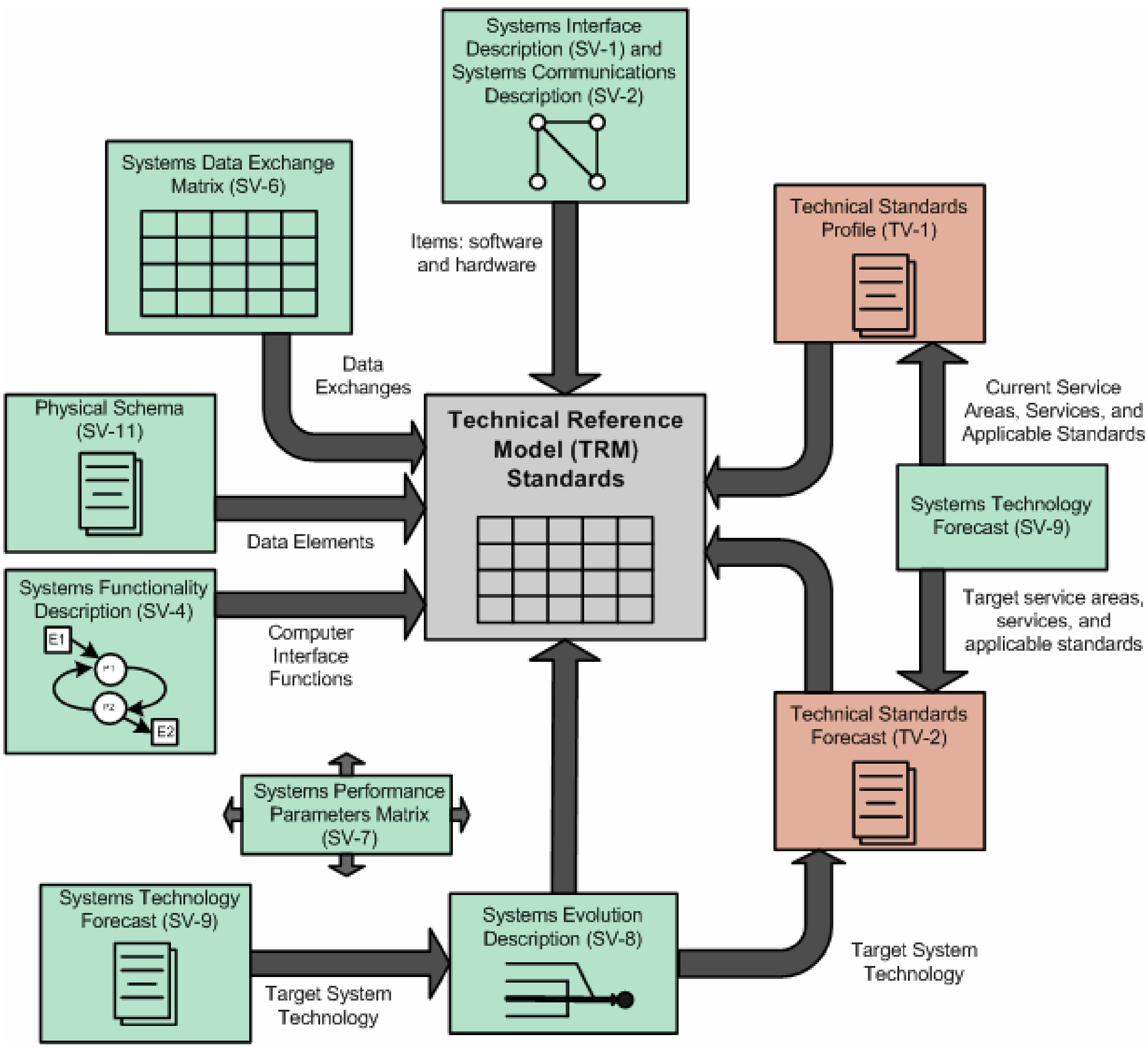
2.5 System Elements that Map to Standards

Systems architecture products and the associated systems elements that have current (TV-1) or forecast (TV-2) technical standards associated with them include:

- Systems and system software and hardware items from SV-1
- Communications links, communications paths, communications networks, and communications systems from SV-2
- System functions from SV-4
- System data elements from SV-6
- Evolving systems from SV-8
- Data modeling techniques used in OV-7 and SV-11
- Timed technology forecasts in SV-9

Figure 4 illustrates the major (not all) systems products and associated systems elements that have current (TV-1) or forecast (TV-2) technical standards associated with them.

Figure 4: Detail of Systems Elements that are Associated with Standards



2.6 Summary of Relationships

Table 1 details architecture data element relationships among the products in the PSAF. The table lists every product in the "Link" column and, in the "To" column, every product to which the Link product is related. Thus, each entry is repeated. Note that rows are color-coded according to the products in the Link and To columns.

Table 1: Detailed Architecture Data Element Relationships

Link	To…	Relationships
OV-1		
	OV-2	Organizations, organization types, and human roles depicted in OV-1 should be traceable to operational nodes in OV-2. Relationships in OV-1 should trace to needlines in OV-2.
OV-2		
	OV-1	Organizations, organization types, and human roles, depicted in OV-1 should be traceable to operational nodes in OV-2. Relationships in OV-1 should trace to needlines in OV-2.
	OV-3	A needline in OV-2 maps to one or more information exchanges in OV-3.
	OV-4	Organizations, organization types, and human roles in OV-4 may map to one or more operational nodes in OV-2. This indicates that the node represents the organization.
	OV-5	The activities annotating an operational node in OV-2 map to the activities described in an OV-5. Similarly, OV-5 should document the operational nodes that participate in each operational activity.
	OV-6c	Lifelines in OV-6c should map to operational nodes in OV-2.
	SV-1	An operational node in OV-2 may be supported by one or more systems in SV-1. This indicates that the operational node owns or uses the system. A needline in OV-2 may map to one or more interfaces in SV-1, and an interface in SV-1 maps to one or more needlines in OV-2.
OV-3		
	OV-2	A needline in OV-2 maps to one or more information exchanges in OV-3.
	OV-5	An information exchange in OV-3 should map in OV-5 to one or more information flows (an external input, an external output, or an output from one operational activity mapped to an input of another) in OV-5. This occurs if OV-5 decomposes to a level that permits such a mapping. Above that level of decomposition, a single information flow in an OV-5 may map to more than one information exchange, or none, if the information flow does not cross node boundaries.
	OV-6b	Events in OV-6b map to triggering events in OV-3.
	OV-6c	Events in OV-6c map to triggering events in OV-3.
	OV-7	An information element in OV-3 should be constructed of entities in OV-7.

Link	To…	Relationships
	SV-6	If any part of an information element in OV-3 originates from or flows to an operational activity that is to be automated, then that Information element should map to one or more system data elements in SV-6.
OV-4		
	OV-2	Organizations, organization types, and human roles in an OV-4 may map to one or more operational nodes in an OV-2, indicating that the node represents the organization.
OV-5		
	OV-2	The activities annotating an operational node in an OV-2 map to the activities described in an OV-5. Similarly, OV-5 should document the operational nodes that participate in each operational activity.
	OV-3	An information exchange in an OV-3 should map in an OV-5 to one or more information flows (an external input, an external output, or an output from one operational activity mapped to an input to another). This mapping applies if OV-5 decomposes to the requisite level. Above that level of decomposition, a single information flow in OV-5 may map to more than one information exchange (or none, if the information flow does not cross node boundaries).
	OV-6a	A rule may define conditions that constrain the execution of an operational activity in a specific way, or constrain the organization or human role authorized to execute an operational activity.
	OV-6b	Actions in OV-6b map to operational activities in OV-5.
	OV-6c	Events in OV-6c map to inputs and outputs of operational activities.
	SV-5	Operational activities in SV-5 match operational activities in OV-5.
OV-6a		
	OV-5	A rule may define conditions that constrain the execution an operational activity in a specific way, or constrain the organization or human role authorized to execute an operational activity.
	OV-6b	A rule may define guard conditions for an action in OV-6b.
	OV-7	The rules in OV-6a may reference the elements of OV-7 to constrain their structure and validity.
OV-6b		
	OV-3	Events in OV-6b map to triggering events in OV-3.
	OV-5	Actions in OV-6b map to operational activities in OV-5.
	OV-6a	A rule may define guard conditions for an action in OV-6b.
	OV-6c	Events associated with transitions in OV-6b map to events of OV-6c.
OV-6c		
	OV-2	Lifelines in OV-6c should map to operational nodes in OV-2.

Link	To…	Relationships
	OV-3	Events in OV-6c map to triggering events in OV-3.
	OV-5	Events in OV-6c map to inputs and outputs of operational activities.
	OV-6b	Events associated with transitions in OV-6b map to events of OV-6c.
	SV-5	A capability associated with a specific sequence in OV-6c matches a capability in SV-5.
OV-7		
	OV-3	An information element in OV-3 should be constructed of entities in OV-7.
	OV-6a	The rules in OV-6a may reference the elements of OV-7 to constrain their structure and validity.
	TV-1	Technical standards in TV-1 apply to modeling techniques in OV-7.
SV-1		
	OV-2	An operational node in OV-2 may be supported by one or more systems in SV-1 (indicating that the operational node owns or uses the system). A needline in OV-2 may map to one or more interfaces in an SV-1, and an interface in SV-1 maps to one or more needlines in OV-2.
	SV-2	An interface in SV-1 is implemented by communications links or communications networks in SV-2.
	SV-3	One entry in an SV-3 matrix represents one interface in SV-1.
	SV-4	SV-4 defines system functions that are executed by systems defined in SV-1.
	SV-5	Systems in SV-1 match systems in SV-5.
	SV-6	Each system data element appearing in a system data exchange is graphically depicted by one of the interfaces in SV-1; an interface supports one or more system data exchanges.
	SV-7	The performance parameters of SV-7 apply to systems, subsystems, and system hardware/software items of SV-1.
	SV-8	The systems, subsystems, and system hardware/software items of SV-8 should match the corresponding elements in SV-1.
	SV-9	Timed technology forecasts in SV-9 impact systems, subsystems, and system hardware/software items of SV-1.
	TV-1	Technical standards in TV-1 apply to and sometimes constrain systems, subsystems, and system hardware/software items in SV-1.
	TV-2	Timed standard forecasts in TV-2 affect systems, subsystems and system hardware/software items in SV-1.
SV-2		
	SV-1	An interface in SV-1 is implemented by communications links or communications networks in SV-2.

Link	To…	Relationships
	SV-7	Performance parameters of SV-7 that deal with communications systems, communications links, and communications networks should map to the corresponding elements in SV-2.
	SV-8	If a grouping link references communications items, they should appear in SV-2.
	SV-9	Timed technology forecasts in SV-9 affect communications systems, communications links, and communications networks in SV-2.
	TV-1	Technical standards in TV-1 apply to, and sometimes constrain, communications systems, communications links, and communications networks in SV-2.
	TV-2	Timed standard forecasts in TV-2 affect communications systems, communications links, and communications networks in SV-2.
SV-3		
	SV-1	One entry in a SV-3 matrix represents one interface in SV-1.
SV-4		
	SV-1	SV-4 defines system functions that are executed by systems defined in SV-1.
	SV-5	System functions in SV-4 should map one-to-one to system functions in SV-5.
	SV-6	System data flows in SV-4 should map to system data elements appearing in system data exchanges of SV-6.
	SV-8	If SV-8 identifies system functions to be implemented in each phase of a system development, they should match the system functions in SV-4.
	SV-10a	If system rules in SV-10a deal with system behavior, they should reference system functions in SV-4.
	SV-10b	System functions in SV-4 may be associated with either states or state transitions in SV-10b.
	SV-10c	Events in SV-10c map to system data flows in SV-4.
	TV-1	Technical standards from TV-1 apply to system functions in SV-4.
	TV-2	Timed standard forecasts in TV-2 affect system functions in SV-4.
SV-5		
	OV-5	Operational activities in SV-5 match operational activities in OV-5.
	OV-6c	A capability associated with a specific sequence in OV-6c matches a capability in SV-5.
	SV-1	Systems in SV-1 match systems in SV-5.
	SV-4	System functions in SV-4 should map one-to-one to system functions in SV-5.
SV-6		

Link	To…	Relationships
	OV-3	If any part of an information element in OV-3 originates from or flows to an operational activity that is to be automated, then that information element should map to one or more system data elements in SV-6.
	SV-1	Each system data element appearing in a system data exchange is graphically interfaced in SV-1; an interface supports one or more system data exchanges.
	SV-4	System data flows in SV-4 should map to system data elements appearing in system data exchanges of SV-6.
	SV-7	Performance parameters in SV-7 that deal with interface performance and system data exchange capacity requirements should trace to system data exchanges in SV-6.
	SV-10b	Events in SV-10b map to triggering events in SV-6.
	SV-10c	Events in SV-10c map to triggering events in SV-6.
	TV-1	Technical standards in TV-1 apply to, and sometimes constrain, system data elements in SV-6.
	TV-2	Timed standard forecasts in TV-2 affect system data elements in SV-6.
SV-7		
	SV-1	The performance parameters of SV-7 apply to systems, subsystems, and system hardware/software items of SV-1.
	SV-2	Performance parameters of SV-7 that deal with communications systems, communications links, and communications networks should map to the corresponding elements in SV-2.
	SV-6	Performance parameters in SV-7 that deal with interface performance and system data exchange capacity requirements should trace to system data exchanges in SV-6.
	SV-8	If required performance ranges defined in SV-7 are associated with an overall system evolution or migration plan defined in SV-8, then the time periods in SV-7 should correspond to the milestones in SV-8.
	SV-9	If the future performance expectations or goals defined in SV-7 are based on expected technology improvements, then the performance parameters and their time periods in SV-7 should be coordinated with the timed technology forecasts defined in SV-9.
SV-8		
	SV-1	The systems, subsystems, and system hardware/software items of SV-8 should match the corresponding elements in SV-1.
	SV-2	If a grouping link references communications items, they should appear in SV-2.
	SV-4	If SV-8 identifies system functions to be implemented in each phase of a system development, they should match the system functions in SV-4.
	SV-7	If required performance ranges defined in SV-7 are associated with an overall system evolution or migration plan defined in SV-8, then the time periods in SV-7 should correspond to the milestones in SV-8.

Link	To…	Relationships
	SV-9	The time periods associated with timelines and milestones in SV-8 should be coordinated with time periods associated with timed technology forecasts in SV-9.
	TV-1	Technical standards in TV-1 constrain evolving systems, subsystems, and system hardware/software items of SV-8.
SV-9		
	SV-1	Timed technology forecasts in SV-9 affect systems, subsystems, and system hardware/software items of SV-1.
	SV-2	Timed technology forecasts in SV-9 affect communications systems, communications links, and communications networks in SV-2.
	SV-7	If the future performance expectations or goals defined in SV-7 are based on expected technology improvements, then the performance parameters and their time periods in SV-7 should be coordinated with the timed technology forecasts defined in SV-9.
	SV-8	The time periods associated with timelines and milestones in SV-8 should be coordinated with time periods associated with timed technology forecasts in SV-9.
	TV-1	Timed technology forecasts in SV-9 may force standards in TV-1 to move to its next version.
	TV-2	Timed standard forecasts in TV-2 may depend on timed technology forecasts in SV-9 becoming available.
SV-10a		
	SV-4	If system rules in SV-10a deal with system behavior, they should reference system functions in SV-4.
	SV-10b	A rule may define guard conditions for an action in SV-10b.
SV-10b		
	SV-4	System functions in SV-4 may be associated with either states or state transitions in SV-10b.
	SV-6	Events in SV-10b map to triggering events in SV-6.
	SV-10a	A rule may define guard conditions for an action in SV-10b.
	SV-10c	Events associated with transitions in SV-10b map to events of SV-10c.
SV-10c		
	SV-4	Events in SV-10c map to system data flows in SV-4.
	SV-6	Events in SV-10c map to triggering events in SV-6.
	SV-10b	Events associated with transitions in SV-10b map to events of SV-10c.
SV-11		

Link	To…	Relationships
	TV-1	Technical standards in TV-1 apply to modeling techniques in SV-11.
TV-1		
	OV-7	Technical standards in TV-1 apply to modeling techniques in OV-7.
	SV-1	Technical standards in TV-1 apply to, and sometimes constrain, systems, subsystems, and system hardware/software items in SV-1.
	SV-2	Technical standards in TV-1 apply to, and sometimes constrain, communications systems, communications links, and communications networks in SV-2.
	SV-4	Technical standards from TV-1 apply to system functions in SV-4.
	SV-6	Technical standards in TV-1 apply to, and sometimes constrain, system data elements in SV-6.
	SV-8	Technical standards in TV-1 constrain evolving systems, subsystems, and system hardware/software items of SV-8.
	SV-9	Timed technology forecasts in SV-9 may force a standard in TV-1 to move to its next version.
	SV-11	Technical standards in TV-1 apply to modeling techniques in SV-11.
TV-2		
	SV-1	Timed standard forecasts in TV-2 affect systems, subsystems and system hardware/software items in SV-1.
	SV-2	Timed standard forecasts in TV-2 affect communications systems, communications links, and communications networks in SV-2.
	SV-4	Timed standard forecasts in TV-2 affect system functions in SV-4.
	SV-6	Timed standard forecasts in TV-2 affect system data elements in SV-6.
	SV-9	Timed standard forecasts in TV-2 may depend on timed technology forecasts in SV-9 becoming available.

3 Architecture Product Development

The PSAF products portray the basic architecture data elements and relationships that constitute an architecture description. *PSAF Volume I* describes a five-step process for developing architecture descriptions:

1. Determine the intended use of the architecture description.
2. Establish the architecture description's scope, context, environment, and consider any other assumptions to be considered.
3. Based on the intended use and scope, determine what information the architecture description needs to capture.
4. Establish products to build. (This step depends on the scope and the purpose of the architecture.)
5. Gather the architecture data and build the requisite products.

The first four steps are independent of any methodology[3] you might use in designing the architecture. They require the involvement of a few persons — the architect and the users. Step five of the architecture development process involves building the requisite products.

3.1 Product Development Methodology Support

To complete step 5 in the preceding list, the PSAF does not advocate the use of any one methodology (e.g., structured analysis as opposed to object orientation), or one notation over another (e.g., IDEF1X UML notation). However, products should contain the required instances of architecture data elements and relationships (i.e., those marked with an asterisk [*]). Still, the need for a well-defined and rigorous methodology is acknowledged and emphasized, and will be presented in *PSAF Volume III*. *PSAF Volume III* is a living document that will guide public safety through the initial production of an architecture that is PSAF-compliant. It will facilitate the analysis necessary to move the public safety community forward in terms of operation capabilities and interoperability. Several candidate methodologies are available for consideration, but the choice is ultimately governed by the following factors:

- The nature of the architecture being defined
- The expertise and preferences of the architecture team
- The needs of the customer and the architecture end users

To actually gather, analyze, and synthesize information into an integrated architecture, you can use an integrated tool or set of tools that allow the development of the products and accompanying text. The use

[3] *Webster's II New College Dictionary*, 2001, defines methodology as: 1) the system of principles, procedures, and practices applied to a particular branch of knowledge, and 2) the branch of logic dealing with the general principles of the formation of knowledge. While the PSAF defines an approach for developing architecture descriptions, it does not specify a methodology for developing an architecture description.

of an integrated tool or tool suite is highly recommended. It will support development of an integrated architecture that is consistent and that incorporates version control. Ensure the tools you select meet the following requirements:

- Allows the architect to produce consistent products across and within the different viewpoints by performing cross- and intra-viewpoint product checking
- Includes a mechanism for storing, updating, and retrieving architecture data and their relationships
- Supports automatic generation of an integrated dictionary
- Can import and export data from an AFDM-compliant database (i.e., GJXDM)

The SAFECOM program is currently evaluating several architecture tools and methods for storing the architecture data captured. Once a tool and interface have been tested in the field with practitioners, Volume III will be published to describe the best practices and use of the selected tool and interface.

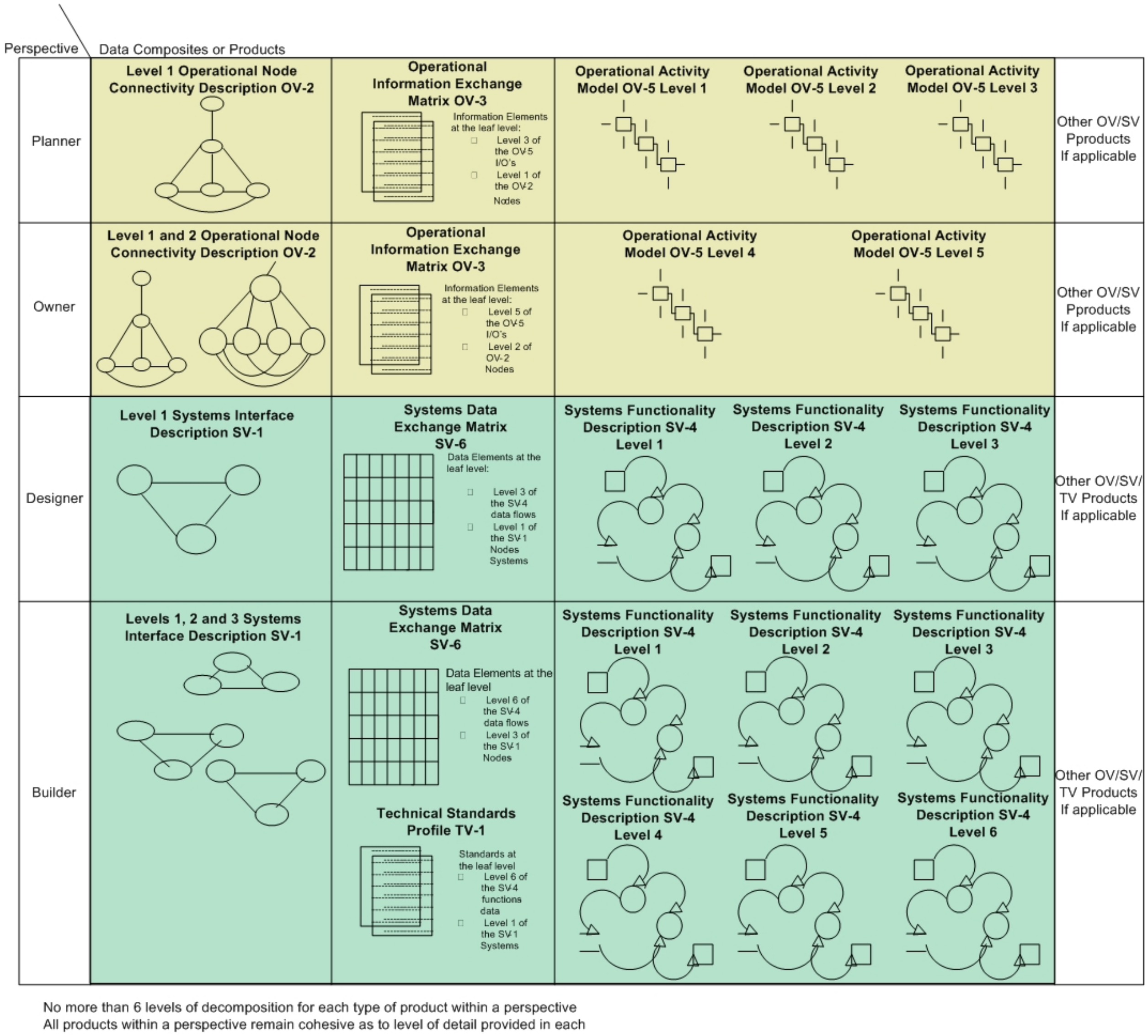
3.2 Architecture Products and Levels of Detail

Most graphical products (e.g., OV-2, OV-5, SV-1, and SV-4) permit the modeling of their respective architecture data elements using decomposition. Decomposition involves several diagrams of the same product that may be developed for the same architecture, with each diagram shows an increasing level of detail. Examples of levels of detail are the various perspectives such as planner, owner, designer, or builder defined by Zachman (1987). In general, the level of usable detail increases as the perspective changes from that of the planner, to the owner, to the designer, to the builder.

Within each perspective, ensure all products you develop remain cohesive with respect to the level of detail. For example, if you develop one diagram of OV-2 operational nodes that shows aggregated organizations only, it is imperative you develop the corresponding OV-5 product to show only those operational activities that are meaningful with respect to these operational nodes. Similarly, the information exchanges of OV-3 should remain at a high level of aggregation to represent actual information workflow products used at the operational nodes shown in OV-2 (and not their subordinate operational nodes).

A good guide to tracking the level of detail in an architecture is to always ensure that the information is at a level of detail meaningful to the intended user of the architecture. In addition, a good rule is to restrict decomposition levels to no more than three levels for any one type of diagram within the same perspective. That is generally sufficient to provide the required level of granularity for a stated objective. Figure 5 illustrates some of the decomposition best practices for various perspectives.

Figure 5: Perspectives and Decomposition Levels



The products and the number of decomposition levels Figure 5 shows are only examples. The collection of products for each perspective or level of detail comprises one model of the architecture. To conduct adequate analyses, you usually need an iterative process that involves developing multiple architecture models, one for each perspective, as discussed below.

3.3 Iterative Development of the Products

Depending on the architecture level needed (e.g., high levels of abstraction that hide design and implementation details) and the intended audience, you can apply an iterative method to develop PSAF products. Iterative development crosses all views. OVs can drive SV and TV changes, SVs can drive OV and TV changes, and so forth. Products iterate across views in the same way that they iterate within one view but across levels of detail.

During this iterative development process, you develop different models at varying levels of abstraction with products that trace from one model to another (Booch, 1999). That is, at the highest level of abstraction, when you develop only a minimum of PSAF products to help describe a new concept of operations, you can develop a few products to produce one model of this architecture (denoted Model A).

This first model may consist of only highly abstract and generic sets of operational nodes, operational activities, and so forth. Later, when you need to add new details and expand the architecture to show more design detail, you must develop a new model. This is Model B, consisting of modified Model A products, plus additional products as necessary.

The new products that make up Model B will include and trace back to the original group of products that make up Model A of the architecture. For example, an operational node in an OV-2 product, as part of Model A may have been used to represent an aggregated organization or command. (This may consist of multiple subordinate operational nodes, where it is deemed unnecessary to show those subordinate nodes at the Model A level.)

In Model B, you can now expand the operational node of Model A's product to show the subordinate nodes. You should not introduce new root level PSAF operational nodes at this level that do not trace back to the previous model. For example, if, in the process of model refinement, you determine that an operational node is part of the architecture, and that this node is not yet a part of any of the aggregated operational nodes of OV-2 included in Model A, then you need to update Model A's OV-2 to include the newly identified node. Model B's OV-2 can then include that subordinate node, which will be a decomposition of the Model A node, and will trace back to that node.

Management of this iterative process will be centralized through *PSAF Volume III*. that the goal is to achieve a consistent level of granularity across the public safety community. This should ensure that when the architectures are collected in a centralized repository, the data can be consistently analyzed.

3.4 Product Templates

Where applicable, the templates for the PSAF products reference industry standard methodologies and techniques. However, there is no requirement to comply with the template's chosen standard. Regardless of the technique you use to develop the product, you must accurately reflect the architecture data elements and their relationships, as defined in the architecture data elements tables, including relationships to architecture data elements in other products. Where applicable, the templates for the PSAF products reference structured analysis (SA) or object-oriented (OO) standard notations.

3.5 OO and UML Support

3.5.1 Relationship to the Unified Modeling Language

The Object Management Group's (OMG) Unified Modeling Language™ (UML) helps you specify, visualize, and document models of software systems, including their structure and design, in a way that meets all of these requirements (OMG, 2000).[4] During the last few years, UML has emerged as the

[4] http://www.omg.org/gettingstarted/what_is_uml.htm

dominant and most prevalent language for modeling regardless of the development process used. (UML also works well for business modeling and modeling of other non-software systems.)

The UML representation is provided in this version of the PSAF to assist architects who choose to use OO methodologies. This representation includes a collection of UML diagram types that describe the same information as the PSAF products.

Note that this document is not a complete and thorough tutorial on the entire UML and the processes for using that language. UML and its applications for software and systems engineering is an ongoing research area. You can consult numerous reference books for additional information on UML and application techniques.

3.5.2 PSAF UML Multi-Diagram and Multi-Model Approach

The PSAF UML representation uses UML notation to model both PSAF OV and SV architecture products. In the case of OV diagrams, the UML semantics have a different meaning from one intended for a software development environment. For example, in a software development environment, a use case diagram depicts actors that are outside the scope of the system to be implemented and whose requirements (what needs to be done) are modeled in the diagram via the use cases. However, in the PSAF's OV, a use case diagram models the operational requirements (what needs to be done, or the operational activities) via a set of use cases, as well as modeling *who* conducts these activities (the actors of the use case diagram). The actors are *within* the scope of the architecture. Corresponding collaboration diagrams in the OV are not intended to model realizations or implementations of the use case diagram. Instead, the actors represent the same operational nodes as those on the use case diagram. They are intended to add information and flow detail to the use cases.

In addition, with a PSAF UML representation, you can use the same set of diagram types to model several operational products. The UML diagram for each type of operational product will represent a different aspect of the architecture. For example, you use a UML class diagram to model OV-4 organizational charts, as well as to define a Logical Data Model (OV-7). In the case of OV-4, using class diagram notation allows the modeling of relationships among organizations (as opposed to relationships among classes of data objects).[5] Classes in the OV-4 diagrams represent organizations, and UML association relationships among these classes represent operational command, control, and organizational relationships among the organizations. Class diagrams in OV-7, on the other hand, show classes that relate to OV-5 activities and information flows.

Appendix C, "Dictionary of UML Terms," lists definitions of the UML terms in this section. Where this PSAF document uses a UML term to convey a slightly different meaning, UML Representation sections fully qualify the term. For example, use of the term *components* in the UML sections refers to any type of system software (as in systems engineering), while the term *component* as referenced in the PSAF denotes PSAF organizational units. The UML Representation sections qualify the term as *UML Component*.

Note that while UML has advantages in some modeling situations, it can be restrictive in modeling the OV and SV products. You might want to consider other modeling notations or standards that if they are more applicable to the needs of the architecture team.

---

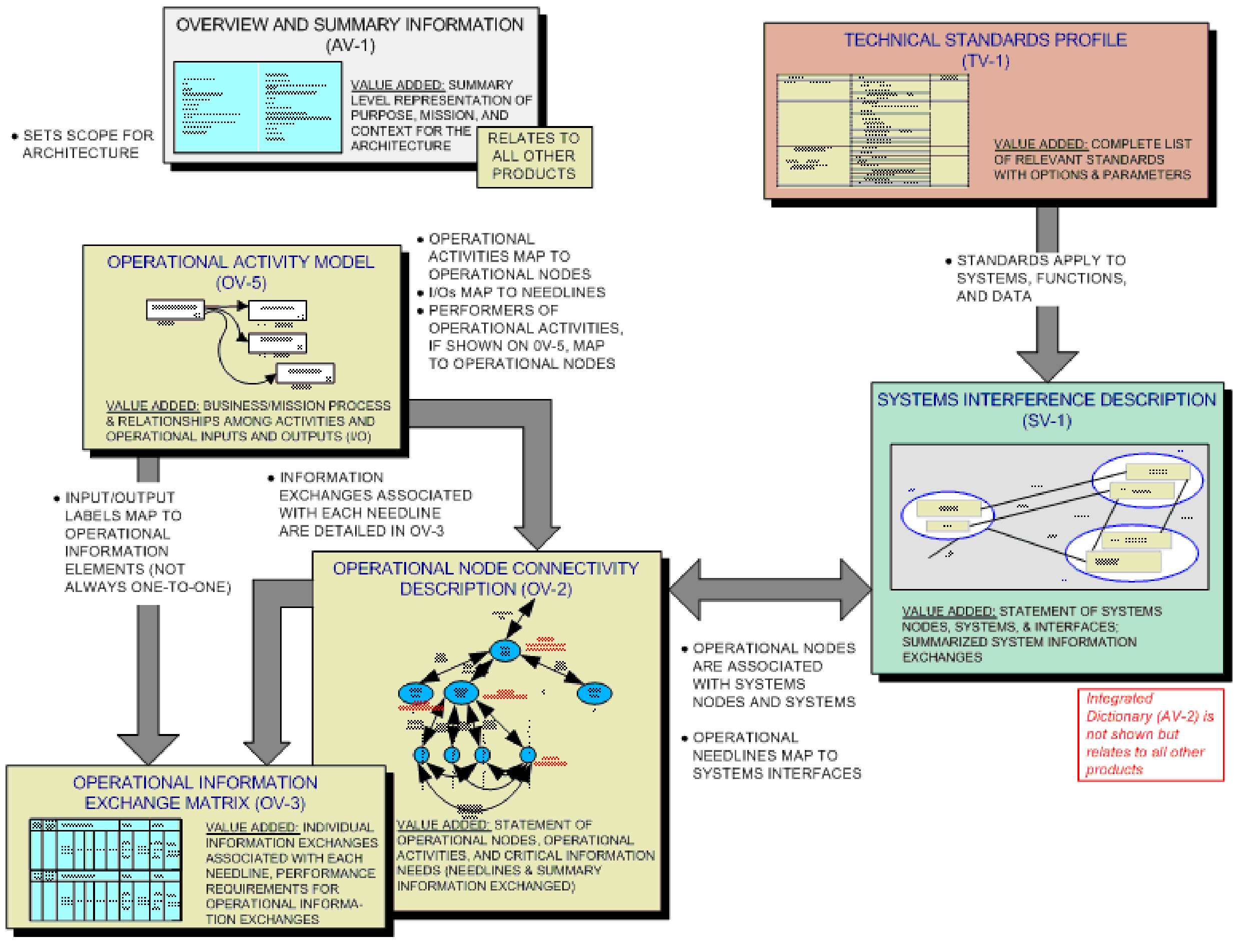
[5] An object is an instance of a class.

3.6 Product and Architecture Data Element Relationships

There are general relationships that logically interconnect the PSAF products from one view to the products of another view. The architect needs to be continuously aware of these necessary relationships to produce an architecture that is consistent across the three views and to provide clear traceability and connections from one view to another. Figure 6 illustrates some relationships among the architecture data elements for a subset of the products.

Section 2 contains a detailed description of these product and architecture data element relationships.

Figure 6: Relationships Among the Products and Architecture Data Elements



3.7 AFDM Support for Architecture Products

The Global Justice XML Data Model (GJXDM) is being proposed for the Architecture Framework Data Model (AFDM) for specification of the architecture data. The GJXDM is an object-oriented data model, database, and XML schema specification, generated from the database. It represents the semantics and structure of common data elements and types required to exchange information consistently within the justice and public safety communities. You can extend this data model to create and store product view information in a database in a way that assists analysis across products generated by different users for different architectures. While the GJXDM does not currently support this application, analysis is underway to extend it to support the PSAF. Subsequent versions of the PSAF will note the progress of this work effort.

4 All-Views Products

The All-Views section defines two products:

- Overview and Summary Information (AV-1)
- Integrated Dictionary (AV-2)

4.1 Overview and Summary Information (AV-1)

4.1.1 Product Definition

The Overview and Summary Information (AV-1) product provides brief executive-level information in a consistent form that allows quick reference and comparison among architectures. AV-1 includes assumptions, constraints, and limitations that may affect high-level decision processes involving the architecture.

4.1.2 Product Purpose

AV-1 contains sufficient textual information to enable a reader to select one architecture from among many to read in more detail. In the initial phases of architecture development, AV-1 serves as a planning guide. Upon completion of an architecture, AV-1 provides summary textual information concerning the architecture.

4.1.3 Product Detailed Description

The AV-1 product comprises a textual executive summary of a given architecture and documents the following descriptions:

- *Architecture Project Identification* identifies the architecture project name, the architect, and the organization developing the architecture. It also includes assumptions and constraints, identifies the approving authority and the completion date, and records the level of effort and costs (projected and actual) required to develop the architecture.
- *Scope* identifies the views and products that have been developed and the temporal nature of the architecture, such as the time frame covered, whether by specific years or by designations such as current, target, transitional, and so forth. Scope also identifies the organizations that fall within the scope of the architecture.
- *Purpose and Viewpoint* explains:
  - The need for the architecture
  - What it should demonstrate
  - The types of analyses (for example, Activity-Based Costing) that will be applied to it
  - Who is expected to perform the analyses

- ◻ What decisions are expected to be made on the basis of an analysis
- ◻ Who is expected to make those decisions
- ◻ What actions are expected to result
- ◻ The viewpoint from which the architecture is developed (e.g., planner or decision maker)

- *Context* describes the setting in which the architecture exists. It includes such things as mission, relevant goals and vision statements, concepts of operation, scenarios, information assurance context (e.g., types of system data to be protected and expected information threat environment), other threats and environmental conditions, and geographical areas addressed, where applicable. Context also identifies authoritative sources for the rules, criteria, and conventions that were followed. The tasking for the architecture project and known or anticipated linkages to other architectures are identified.
- *Tools and File Formats Used* identifies the tool suite used to develop the architecture and file names and formats for the architecture and each product.
- *Findings* states the findings and recommendations that have been developed based on the architecture effort. Examples of findings include identification of shortfalls, recommended system implementations, and opportunities for technology insertion.
- During the course of developing an architecture, several versions of this product may be produced. An initial version may focus the effort and document its scope, the organizations involved, and so forth. After other products within the architecture's scope have been developed and verified, another version may be produced to document adjustments to the scope and to other architecture aspects that may have been identified as a result of the architecture development. After the architecture has been used for its intended purpose, and the appropriate analysis has been completed, yet another version may be produced to summarize these findings for the high-level decision makers. In this version, the AV-1 product, along with a corresponding graphic in the form of an OV-1 product, serve as the executive summary for the architecture. Figure 7 shows a representative format for the AV-1 product.

Figure 7: AV-1 - Representative Format

- **Architecture Project Identification**
  - Name
  - Architect
  - Organization Developing the Architecture
  - Assumptions and Constraints
  - Approval Authority
  - Date Completed
  - Level of Effort and Projected and Actual Costs to Develop the Architecture
- **Scope: Architecture View(s) and Products Identification**
  - Views and Products Developed
  - Time Frames Addressed
  - Organizations Involved
- **Purpose and Viewpoint**
  - Purpose, Analysis, Questions to be Answered by Analysis of the Architecture
  - From Whose Viewpoint the Architecture is Developed
- **Context**
  - Mission
  - Goals and Vision
  - Rules, Criteria and Conventions Followed
  - Tasking for Architecture Project and Linkages to Other Architectures
- **Tools and File Formats Used**
- **Findings**
  - Analysis Results
  - Recommendations

4.1.4 UML Representation

The UML tools used for analysis and design usually allow for the addition of documentation to annotate the model/architecture being designed. There is no specific UML product (diagram) that is equivalent to the AV-1 product.

4.1.5 Data Element Definitions

Table 2 defines the data elements related to the AV-1 product. As noted earlier, data elements marked with an asterisk (*) should be included by the architecture development team, if the product is chosen for development as part of an integrated architecture effort.

Table 2: Data Element Definitions for AV-1

Data Element	Attributes	Explanation
Non-Graphical Types		
Architecture Project Identification*		
	Architecture Project Name*	Name/identifier of architecture development project, i.e., name of architecture being described (e.g., Boulder County Law Enforcement)

Data Element	Attributes	Explanation
	Architecture Project Description*	Description of the architecture project (e.g., Boulder County Law Enforcement describes…)
	Architect Name*	Name of chief architect charged with development or documentation of the architecture
	Developing Organization*	Name of organization charged with development or documentation of the architecture (who owns the architecture)
	Assumptions and Constraints*	Text description of architecture project assumptions and constraints, including budget and schedule constraints
	Approval Authority*	Name of organizations approving the architecture
	Date Completed*	Date on which the architecture description was completed
Architecture*		
	Name*	Name of the architecture
	Description*	Text description of the architecture
Architecture View*		
	Name*	Name/identifier of an architecture view
	Description*	Description of an architecture view
Architecture Product*		
	Name*	Name/identifier of an architecture product
	Description*	Description of an architecture product
Organization*		See OV-4 definition table
Architecture Scope*		
	Scope Identifier*	Identifier for architecture scope
	Views	Free-text field listing view included
	Products	Free-text field listing products developed
	Time Frame*	Current and target dates together with relevant dates (e.g., current as of July 2005; target for 2010, target for 2015)
	Organizations Involved	Free-text field listing organizations involved
Tasking		
	Name	Name/identifier of the tasking that created the architecture project
	Source	Source of the tasking (e.g., organization, directive, order)

Data Element	Attributes	Explanation
	Description	Text summary of the tasking that created the architecture project
Architecture Purpose*		
	Identifier*	Name/identifier of an architecture purpose
	Description*	Text description of the architecture purpose
Architecture Viewpoint		
	Identifier*	Name/identifier of an architecture viewpoint
	Description	Text indicating from whose viewpoint that architecture is described (e.g., planner)
Development Effort		
	Effort Identifier	Identifier for the level of effort
	Type	Actual or projected
	Level	Text description of the level of effort (actual or projected)
	Cost	Actual or projected cost
Analysis		
	Name	Name/identifier of analysis process that was or will be applied to the architecture
	Description	Description of the analysis process
Mission		
	Name	Mission name/identifier
	Description	Description of mission
	Geographic Configuration	Description of the geographical configuration of the architecture – may be generic (e.g., Personal Area Network, Incident Area Network, etc.)
	Political Situation	Description of the political context (e.g., mutual aid agreement between two counties; task force operations)
Software Tools*		
	Software Tool Name*	Full name of software tools used to develop an architecture product, including version number and platform used
	Software Tool Vendor*	Name and contact information for software tool vendor
	Software Tool Description*	Text description of software tool, including software tool functions used
	Software Tool Output Formats*	File formats for software tool, including software tool functions used

Data Element	Attributes	Explanation
	Hardcopy Reference	Name of reference to the hardcopy document (i.e., name, date, etc.) in which product is included; may include primary Point of Contact (POC) for changes
	Softcopy Reference	Reference to softcopy database or file name (e.g., URL); may include primary POC for changes
Analysis Results*		
	Identifier*	Name/identifier of a particular instance of applying an analysis process to the architecture
	Date Analysis Performed*	Date on which the analysis technique was used
	Technique Used*	Name and description of analysis technique used
	Description*	Text summary of results
	Location*	Reference to keeper of document, could be URL, organization name, or physical location
Recommendations*		
	Identifier*	Name/identifier of recommendation or recommendation set
	Description*	Description of recommendation
	Date Made*	Date on which recommendations were made
Relationships		
Architecture Project Develops Architecture*		
	Architecture Project Name	Architecture project name/identifier
	Architecture Name	Name of architecture whose description is a product of the project
Architecture has Scope*		
	Architecture Name	Architecture name/identifier
	Scope Identifier	Identifier for architecture scope
Architecture Contains Architecture View*		
	Architecture Name*	Architecture name/identifier
	View Name*	Name of view included in the architecture (e.g., Single Discipline/Single Jurisdiction Operational Architecture)
View Contains Product*		
	Architecture View*	Name/identifier of architectural view

Data Element	Attributes	Explanation
	Architecture Product*	Name/identifier of architecture product contained in the view
Architecture Product Developed Using Software Tool		
	Architecture Product Name	Name/identifier of a specific architecture product
	Software Tool Name	Full name of software tool (including version number and platform) used to develop this architecture product
Architecture Involves Organizations		
	Architecture Name	Name/identifier of an architecture
	Organization Name	Name/identifier of an organization involved in this architecture
Architecture has Purpose*		
	Architecture Name	Architecture name or identifier
	Organization Name	Identifier of an architecture purpose
Architecture is Described from Viewpoint		
	Architecture Name	Architecture name/identifier
	Architecture Viewpoint Identifier	Identifier of an architecture viewpoint
Development Effort is Projected for Architecture		
	Architecture Project Name	Name/identifier of an architecture project
	Development Effort Identifier	Identifier of a development effort projected for development of an architecture
Development Effort is Expended on Architecture		
	Architecture Project Name	Name/identifier of an architecture project
	Development Effort Identifier	Identifier of a development effort expended for development of an architecture
Architecture has Context with Respect to Mission		
	Architecture Name	Name/identifier of architecture
	Mission Name	Name/identifier of mission associated with this architecture
Architecture Project has Context with Respect to Tasking		
	Architecture Project Name	Name/identifier of architecture project

Data Element	Attributes	Explanation
	Tasking Name	Name/identifier of tasking that generated the architecture project
Architecture has Context with Respect to Other Architectures		
	Architecture Name	Name/identifier of architecture
	Related Architecture Name	Name/identifier of another architecture whose views or products are referenced by this architecture
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
Analysis is Performed on Architecture		
	Architecture Name	Architecture 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 for recommendation set that was based on this specific set of results
Results Obtained Using Software Tool		
	Analysis Results Identifier	Identifier for results set associated with a specific execution of the analysis process
	Software Tool Name	Full name of software tool (including version number and platform) used to help produce results for this particular execution of the analysis process
Architecture Project has Findings in the Form of Analysis Results*		
	Architecture Project Name*	Name/identifier of architecture project

Data Element	Attributes	Explanation
	Analysis Results*	Identifier of analysis results set produced based on architecture views and products developed by this project
Architecture Project has Findings in the Form of 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

4.1.6 AFDM Support

GJXDM is being proposed for the AFDM. See Section 3.7 for information.

4.2 Integrated Dictionary (AV-2)

4.2.1 Product Definition

The Integrated Dictionary (AV-2) contains definitions of terms used in the given architecture. It consists of textual definitions in the form of a glossary, a repository of architecture data, their taxonomies, and their metadata (i.e., data about architecture data). The metadata includes metadata for tailored products, associated with the architecture products developed. Metadata are the architecture data types, possibly expressed in the form of a physical schema. Note that in this document, architecture data types are referred to as architecture data elements.

4.2.2 Product Purpose

AV-2 provides a central repository for a given architecture's data and metadata. AV-2 enables the set of architecture products to stand alone, allowing them to be read and understood with minimal reference to outside resources. AV-2 is an accompanying reference to other products, and its value lies in unambiguous definitions.

Note: The key to long-term interoperability can reside in the accuracy and clarity of architecture product definitions.

4.2.3 Product Detailed Description

AV-2 defines terms used in an architecture, but it is more than a simple glossary. Many architectural products have implicit or explicit information in the form of a glossary, a repository of architecture data, their taxonomies, and their metadata. Each labeled item (e.g., icon, box, or connecting line) in the graphical representation has a corresponding entry in AV-2. Each item from a textual representation of an architectural product also has a corresponding entry in AV-2. The type of metadata included in AV-2 for each item depends on the type of architectural product from which the item is taken. For example, the metadata for an operational node in AV-2 includes the attributes Name, Description, and Level Identifier. You can consult a taxonomy of operational nodes applicable to the architecture and choose a name for a specific operational node from that taxonomy. The AV-2 entry for the node consists of the metadata data fields (a name field, a description field, and a level identifier field), a value for each of these fields, and the taxonomy for operational nodes.

Metadata, which refers to the architecture data types, are defined in the Data Element tables provided for each product in this volume. These tables identify key architecture data types (concepts about which architecture data is recorded), their attributes, and explanations. The tables form the primary requirements for the GJXDM. At a minimum, AV-2 contains the data values for one or more specific architectures; it is ideally a repository conforming to the GJXDM.

Architects should use standard terms where possible. For example, they should use terms from existing, approved dictionaries, glossaries, and lexicons, such as the National Incident Management System (NIMS). However, when a given architecture is at a lower level of detail than existing dictionaries or lexicons, or when new concepts are devised for objective architectures, you might need new terms or modified definitions of existing terms. All definitions that originate in existing dictionaries should provide a reference for the source of the definition. The definition itself should also be provided, so that the architectures are self-contained.

4.2.4 Taxonomies

AV-2 defines the architecture data and their common terms of reference in creating, maintaining, and using architecture products. The Operational, Systems, and Technical Standards View products are interrelated, sometimes very extensively. Because of this interrelationship among products and across architecture efforts, it is useful to define a common terminology with common definitions, referred to as *taxonomies*, in the development of the architecture products. These taxonomies are the building blocks for architecture products.

The need for standard taxonomies derives from lessons learned from early architecture development issues, including the independent development of multiple operational architectures that could not be integrated. Integration was impeded because of the use of different terminology to represent the same architecture data. Use of taxonomies to build architecture products has the following benefits over free-text labeling:

- Provides consistency across products
- Supplies consistency across architectures
- Aids architecture development, validation, maintenance, and re-use
- Traces architecture data to authoritative data sources

The critical taxonomies requiring concurrence and standardization for integrated architectures are the following:

- Operational Nodes. The taxonomy involves operations nodes that represent organizations, organization types, and occupational specialties. The taxonomy minimally consists of names and descriptions, and breakdowns into the parts of the organization, organization type, or human role.

- Operational Activities or Tasks. The taxonomy minimally consists of names, descriptions, and decomposition into the constituent parts that comprise a process-activity.
- Information Elements. Consists at a minimum of names of information elements exchanged, descriptions, decomposition into constituent parts and subtypes, and mapping to system data elements exchanged.
- Systems Nodes. Involves systems nodes that represent facilities, platforms, units, and locations. The taxonomy minimally consists of names, descriptions, breakdowns into constituent parts of the node, and categorizations of types of facilities, platforms, units, and locations.
- Systems. Involves systems consisting of FoS, SoS, networks of systems, individual systems, and items such as hardware and software. The taxonomy minimally consists of names, descriptions, breakdowns into the constituent parts of the system and categorization of types of systems. Subtyping may also address variations across time and systems node installation.
- System Functions. Consists at a minimum of names, descriptions, and decomposition into the constituent parts that comprise a system function.
- Triggers or Events. Consists at a minimum of names, descriptions, and breakdown into constituent parts of the event or trigger, and categorization of types of events or triggers.
- Performance Parameters. Consists at a minimum of names, descriptions, units of measure, and conditions that may be applicable to performance parameters.
- Technical Standards. Consists at a minimum of categories of standards.
- Technology Areas. Consists at a minimum of names, descriptions, and categories of technologies into which individual science and technology initiatives and programs can be categorized.

These taxonomies are used to construct various architecture products as shown in Figure 8. In the figure, taxonomy refers to a set of relationships, often hierarchical, among pairs of instances. (An instance is an individual entity with its own identity and value.) Composition refers to the use of one instance to represent and include as a subset a group of instances. The symbols in the table — representing a primary or secondary role for the taxonomy — represent the potential role played by the taxonomy, and not by the architecture data elements themselves.

For example, Operational nodes are important to OV-1, but taxonomies of these nodes are more important for products OV-2 through OV-5. The table shows that taxonomies potentially have a strong role to play in AV-2, as well as many of the OV, SV, and TV products.

Figure 8: Taxonomies Used in Products

TAXONOMY TYPES	STRUCTURE	AV		Operational View (OV)							System View (SV)											TV	
		1	2	1	2	3	4	5	6	7	1	2	3	4	5	6	7	8	9	10	11	1	2
Operational Nodes *Organizations, Types of Organizations, and Occupational Specialties*	Taxonomy & Composition		●		●	●	●	●	●		⊛					⊛							
Operational Activities and Tasks	Taxonomy & Composition		●		●	●		●	●	●					●	⊛					●		
Information Elements *And mappings to systems Data Elements*	Taxonomy & Composition		●		⊛	●		●	●	●				⊛		⊛				⊛	●	⊛	⊛
Systems Nodes *Facilities, Platforms, Units, and Locations*	Taxonomy & Composition		●								●	●	⊛	⊛		●							
Systems *Family of Systems, System of Systems, Networks, Applications, Software, and Equipment*	Taxonomy & Composition		●								●	●	●	●	⊛	●	●	●	●	●		⊛	⊛
System Functions	Composition		●								⊛			●	●	●	⊛	⊛	⊛	●			
Triggers/Events	Taxonomy & Composition		●			●		●	●					●		●				●			
Performance Parameters	Taxonomy & Composition		●					●						●		●	●	●	●				
Technical Standards *Info Processing, Info Transfer, Data, Security, and Human Factors*	Taxonomy & Composition		●								●	●		⊛		●		●	●		⊛	●	●
Technology Areas *Systems and Standards*	Taxonomy & Composition		●																●				●

● = Taxonomy element plays a primary role ⊛ = Secondary role blank = Element not part of this product

Note: Not all architecture data in a given taxonomy is useful in every architectural development. However, given the ongoing evolutionary change in organizations, systems, and processes, the value of using established, validated taxonomic structures that can be expanded or contracted as needed becomes obvious. Moreover, the development of new products over time is greatly simplified as understanding of the taxonomies is increased. Standard taxonomies become building blocks for more comprehensive, quality architectural products.

In some cases, a specific community may have its own operational vocabulary. This local operational vocabulary may use the same terms in radically different ways from other operational communities. In these cases, the internal community versions of the architecture products *should* use the vocabulary of the local operational community to achieve community cooperation and buy-in. These architecture products should include notes on any unique definitions used and provide a mapping to standard definitions, where possible.

4.2.5 UML Representation

UML tools used for analysis and design usually include a data dictionary facility that generates a data dictionary or glossary of terms from the annotated definitions that users enter as they build PSAF products. The collection of products stored in the tool for a specific architecture comprises the repository of architecture data and the metadata specification.

4.2.6 Data Element Definitions

As discussed in Section 4.2.1, data element tables for each of the products collectively describe architecture data types to be captured in AV-2 on a product-by-product basis. The architecture data as completed for a specific architecture forms part of AV-2 for that architecture. Each definition table is described in following subsections of this volume. They consist of architecture data types that define the related product, consisting of architecture data element names and attributes, as well as descriptions of the architecture data element relationships that compose the architecture data elements to create an architecture product. In addition, AV-2 includes a glossary of terms and any taxonomy used. For every architecture data element that is sourced from a predefined taxonomy, the AV-2 entry should specify the reference, that is, the source of the taxonomy definition, and the structure.

4.2.7 AFDM Support

GJXDM is being proposed for the AFDM. See Section 3.7 for information.

5 Operational View Products

An Operational View (OV) describes the tasks and activities, operational elements, and information exchanges required to conduct operations. A pure OV is material-independent. This means the view does not depend on the technology in use, but instead focuses on the operational description. However, operations and their relationships may be influenced by new technologies such as collaboration technology, where process improvements are being practiced before policy can reflect the new procedures. At times, given the restrictions of current systems, it might be necessary to document the way processes are performed. The aim is to examine ways in which new systems could aid in streamlining the processes. In such cases, an OV may have material constraints and requirements that must be addressed. For this reason, it may be necessary to include some high-level SV architecture data as overlays or augmenting information onto the OV products.

There are seven OV products described in this section:

- High-Level Operational Concept Graphic (OV-1)
- Operational Node Connectivity Description (OV-2)
- Operational Information Exchange Matrix (OV-3)
- Organizational Relationships Chart (OV-4)
- Operational Activity Model (OV-5)
- Operational Activity Sequence and Timing Descriptions (OV-6a, 6b, and 6c)
- Logical Data Model (OV-7)

5.1 Incident Scenario Reference

A number of the OV product examples in the next sections use aspects of the multi-discipline, multi-jurisdiction chemical plant explosion incident response scenario described in the PS SoR. This useful scenario requires information exchanges that span disciplinary and jurisdictional boundaries. This span of information exchange provides a good test of how interoperable various public safety agencies' communications systems must be to achieve operational mission goals.

The incident scenario describes the response to a chemical plant explosion and fire that requires not only fire response assets, but also law enforcement and emergency medical services assets. These would involve commanders from multiple jurisdictions, as follows:

- An Incident Commander (IC) provides overall incident supervisory command on scene, with individual incident public safety discipline, or branch, commanders providing the next level of subordinate supervisory command to their respective discipline response teams. Examples of such commanders include the Incident Law Enforcement Commander (ILEC), Incident Fire Commander (IFC), and Incident Emergency Medical Services Commander (IEMSC).
- The Emergency Manager (EM) provides logistical support to the IC and communicates with other organizations and personnel external to the incident. In addition to command responsibilities, the IC and EM have all on-scene monitoring and alerting responsibilities relating to assets. The

individual public safety discipline commanders have the responsibility to monitor and alert the assets of their respective disciplines.

- At the next level below the ICs and EMs, responding jurisdictional commanders exist for each respective discipline, such as the Jurisdiction-n Fire Commander (Jn FC), who provides subordinate supervisory command over the corresponding discipline's and jurisdiction's first responders (for example, Jurisdiction-n Fire First Responder-m (Jn FFR-m)).

The incident scenario focuses on the information exchanges between the IC and EM, and their activities as they interact with the various discipline commands. The products that follow provide the level of detail that supports this perspective.

The seven OV products described in this section are:

- High-Level Operational Concept Graphic (OV-1)
- Operational Node Connectivity Description (OV-2)
- Operational Information Exchange Matrix (OV-3)
- Organizational Relationships Chart (OV-4)
- Operational Activity Model (OV-5)
- Operational Activity Sequence and Timing Descriptions (OV-6a, 6b, and 6c)
- Logical Data Model (OV-7)

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

5.2.1 Product Definition

The High-Level Operational Concept Graphic describes a mission and highlights main operational nodes (see OV-2 definition) and interesting or unique aspects of operations. It provides a description of the interactions between the subject architecture and its environment, and between the architecture and external systems. A textual description accompanying the graphic is crucial. Graphics alone are not sufficient for capturing the necessary architecture data.

5.2.2 Product Purpose

The purpose of OV-1 is to provide a quick, high-level description of what the architecture is supposed to do, and how it is supposed to do it. This product can be used to orient and focus detailed discussions. Its main utility is as a facilitator of human communication. It is intended for presentation to high-level decision makers.

5.2.3 Product Detailed Description

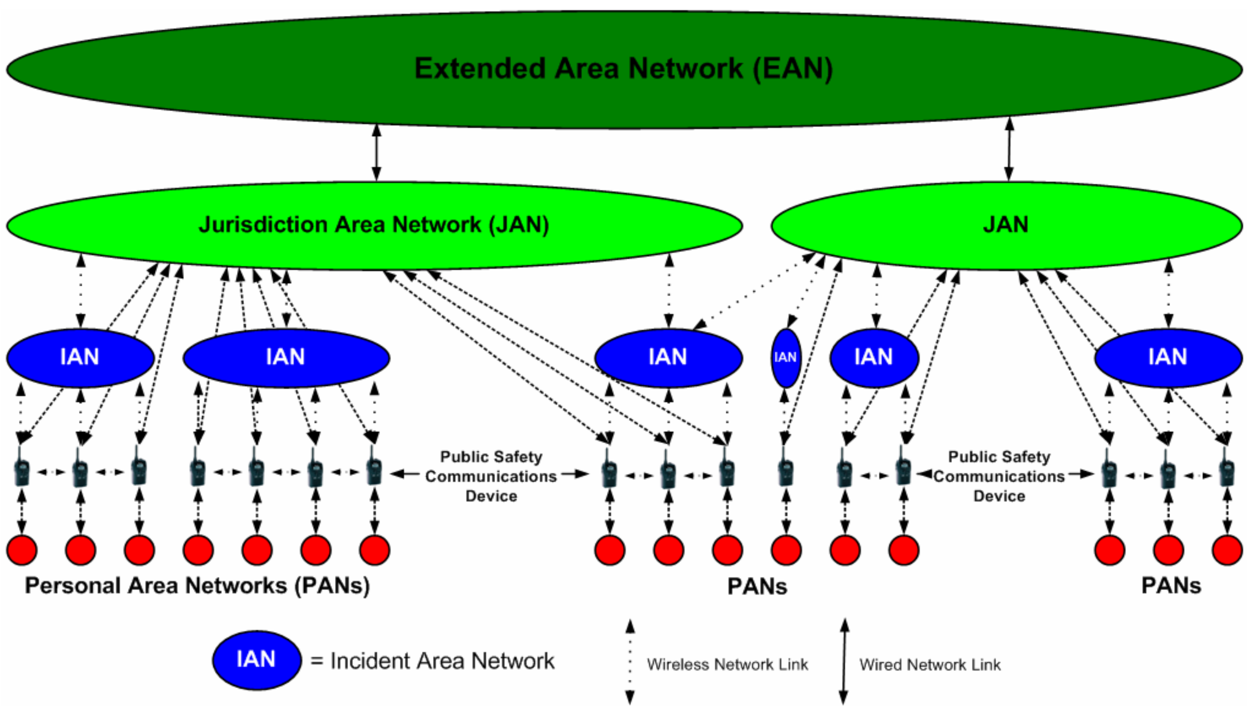
OV-1 consists of a graphical executive summary, with accompanying text, for a given architecture. The product identifies the mission or domain covered in the architecture and the viewpoint reflected in the architecture. OV-1 should convey, in simple terms, what the architecture is about and offers an idea of the players and operations.

The content of OV-1 depends on the scope and intent of the architecture. However, in general it describes the business processes or missions, high-level operations, organizations, and geographical distribution of assets. The product should frame the operational concept — what happens, who does what, in what order, to accomplish what goal. It also should highlight interactions to the environment and other external systems.

During the course of developing an architecture, several versions of this product may be produced. An initial version may focus the effort and illustrate its scope. After other products within the architecture's scope have been developed and verified, another version may reflect adjustments to the scope and other architecture details identified as a result of the architecture development. After the architecture is used for its intended purpose and the appropriate analysis is completed, yet another version may summarize findings for high-level decision makers.

OV-1 is the most general of the architecture products and the most flexible in format. Because the format is free-form and variable, this product displays no template. However, the product usually consists of one or more graphics, or possibly a video, as needed, as well as explanatory text. Figure 9 provides a graphical example of the natural network hierarchy as discussed in the PS SoR.

Figure 9: Public Safety Example Concept of Operations (OV-1)



5.2.4 UML Representation

It is suggested that OV-1s be constructed in freeform format. If you like, you can construct additional, more detailed operational concept diagrams using the UML use case diagrams. In this situation, you can build several use case diagrams, each focusing on a different architecture mission or operational objective. The use case diagrams are further refined to form OV-5.

5.2.5 Data Element Definitions

OV-1 consists of an informal, graphical representation of operations, as well as explanatory text. It is a free-form product. However, architecture data elements that appear in an OV-1 are related to others appearing in several OV and SV products. Table 3 lists some of these architecture data elements.

Note: As previously stated, architecture data elements marked with an asterisk (*) are essential for products developed as part of an integrated architecture.

Table 3: Data Element Definitions for OV-1

Data Element	Attributes	Explanation
Graphical Box Types		
Asset Icon*		
	Name*	The generic asset name that appears on the graphics (e.g., fire engine, police cruiser)
	Representation Type*	Type of asset represented by the icon (e.g., operational node, operational activity, systems node, or system)
	Description*	Textual description of the asset
	Generic Location	Location with respect to geographic configuration on graphic (e.g., specific land location or installation).
Target Area		A geospatial feature of operational interest that is represented graphically in an architecture product
	Identifier	Label on a graphic or other assigned identifier
	Description	Text description of target importance or role
	Generic Location	Location with respect to geographic configuration on graphic
Graphical Arrow Types		
Line*		
	Identifier*	A generic line identifier
	Descriptive Name*	A generic name that indicates the kind of line represented (e.g., needline, interface, communications link)
	Description*	A description of the kind of line represented

Data Element	Attributes	Explanation
	From	Name of the originating asset icon
	To	Name of the destination asset icon
Referenced Types		
Needline		See OV-2 data element definition
External Node		See OV-2 data element definition
Operational Node		See OV-2 data element definition
Operational Activity		See OV-5 data element definition
Systems Node		See SV-1 data element definition
System		See SV-1 data element definition
Interface		See SV-1 data element definition
Communications Network		See SV-2 data element definition
Communications Link		See SV-2 data element definition
Relationships		
Asset Icon Represents an Operational Node, an Operational Activity, a Systems Node, or a System		
	Asset Icon Name	Name of asset icon
	Operational Node, Operational Activity, Systems Node, or System Name	Name/identifier of an operational node, an operational activity, a systems node, or a system represented by an icon
Asset Icon is Associated with Operational Activity		
	Asset Icon Name	Name of asset icon
	Operational Activity Name	Name/identifier of operational activity associated with asset icon
Asset Icon is Associated with Target Area		
	Asset Icon Name	Name of asset icon
	Target Area Name	Name/identifier of target area associated with icon

Data Element	Attributes	Explanation
Line Represents Needline, Interface, Communications Link, or Communications Network		
	Line Name	Name/identifier of line connecting asset icons
	Needline, Interface, Communications Link, or Communications Network Name	Name/identifier of needline, interface, communications link, or communications network represented by line
Line Connects to Asset Icon		
	Asset Icon Name	Name of asset icon
	Line Name	Name/identifier of line connecting asset icon
Line Connects to Target Area		
	Target Area Name	Name of target area
	Line Name	Name/identifier of line connecting target area

5.2.6 AFDM Support

GJXDM is being proposed for the AFDM. See Section 3.7 for information.

5.3 Operational Node Connectivity Description (OV-2)

5.3.1 Product Definition

The Operational Node Connectivity Description (OV-2) graphically depicts the operational nodes, or organizations, with needlines between those nodes that indicate a need to exchange information. The graphic includes internal operational nodes, internal to the architecture, as well as external nodes.

5.3.2 Product Purpose

OV-2 tracks the need to exchange information from specific operational nodes that play a key role in the architecture to others. OV-2 does not depict the connectivity between the nodes.

5.3.3 Product Detailed Description

The main features of this product are the operational nodes and the needlines between them that indicate a need to exchange information. The product indicates the key players and the interactions necessary to conduct the corresponding operational activities of OV-5.

5.3.4 Operational Nodes

An operational node is an element of the operational architecture that produces, consumes, or processes information. What constitutes an operational node can vary among architectures. Architectures may include, but not be limited to, those representing an operational or human role (e.g., Incident Commander), an organization (e.g., Department of Homeland Security) or organization type, i.e., a logical or functional grouping (e.g., Logistics Node, Operations Node). The operational node will also vary depending on the level of detail the architecture effort addresses.

5.3.5 Needlines and Information Exchanges

A needline documents the requirement to exchange information between nodes. The needline does *not* indicate how the information transfer is implemented. For example, if information is produced at node A, and is simply routed through node B, and is used at node C, then node B would not be shown on the OV-2 diagram. The needline would go from node A to node C.

OV-2 is not a communications link or communications network diagram. The system implementation, or what systems nodes or systems are used to execute the transfer, is shown in the Systems Interface Description (SV-1). Further, the needline systems equivalent is the interface line depicted in SV-1. The actual implementation of an interface may take more than one form. It is documented in a Systems Communications Description (SV-2). Therefore, a single needline shown in the OV may translate into multiple interfaces in SV-1 and multiple physical links in SV-2.

Needlines are represented by arrows that indicate the direction of information flow. They are annotated with a diagram-unique identifier, and a phrase that is descriptive of the principal types of information exchanged. It is important to note that the arrows on the diagram represent *needlines* only. This means that each arrow indicates only that there is a need for some kind of information transfer between the two connected nodes.

A one-to-many relationship exists from needlines to information exchanges. For instance, a single needline on OV-2 represents multiple individual information exchanges. The mapping of the information exchanges to the needlines of OV-2 occurs in the Operational Information Exchange Matrix (OV-3). For example, OV-2 may list Situational Awareness as a descriptive name for a needline between two operational nodes. In this case, the needline represents a number of information exchanges, consisting of various types of reports, or information elements, and their attributes (such as periodicity and timeliness) that are associated with the Situational Awareness needline. The identity of the individual information elements and their attributes are documented in OV-3.

OV-2 should also illustrate needs to exchange information between operational nodes and external nodes. This involves operational nodes that are not strictly within the scope of the subject architecture, but are important sources of information required by nodes within the architecture, or are important destinations for information provided by nodes within the architecture.

5.3.6 Operational Activities

You can list operational activities (from the OV-5 Operational Activity Model) performed by a given node on the graphic, if space permits. OV-2, in effect, turns OV-5 inside out, focusing first-order on the operational *nodes* and second-order on the activities. OV-5, on the other hand, places primary attention on operational *activities* and only secondary attention on nodes, which can be shown as annotations on the activities.

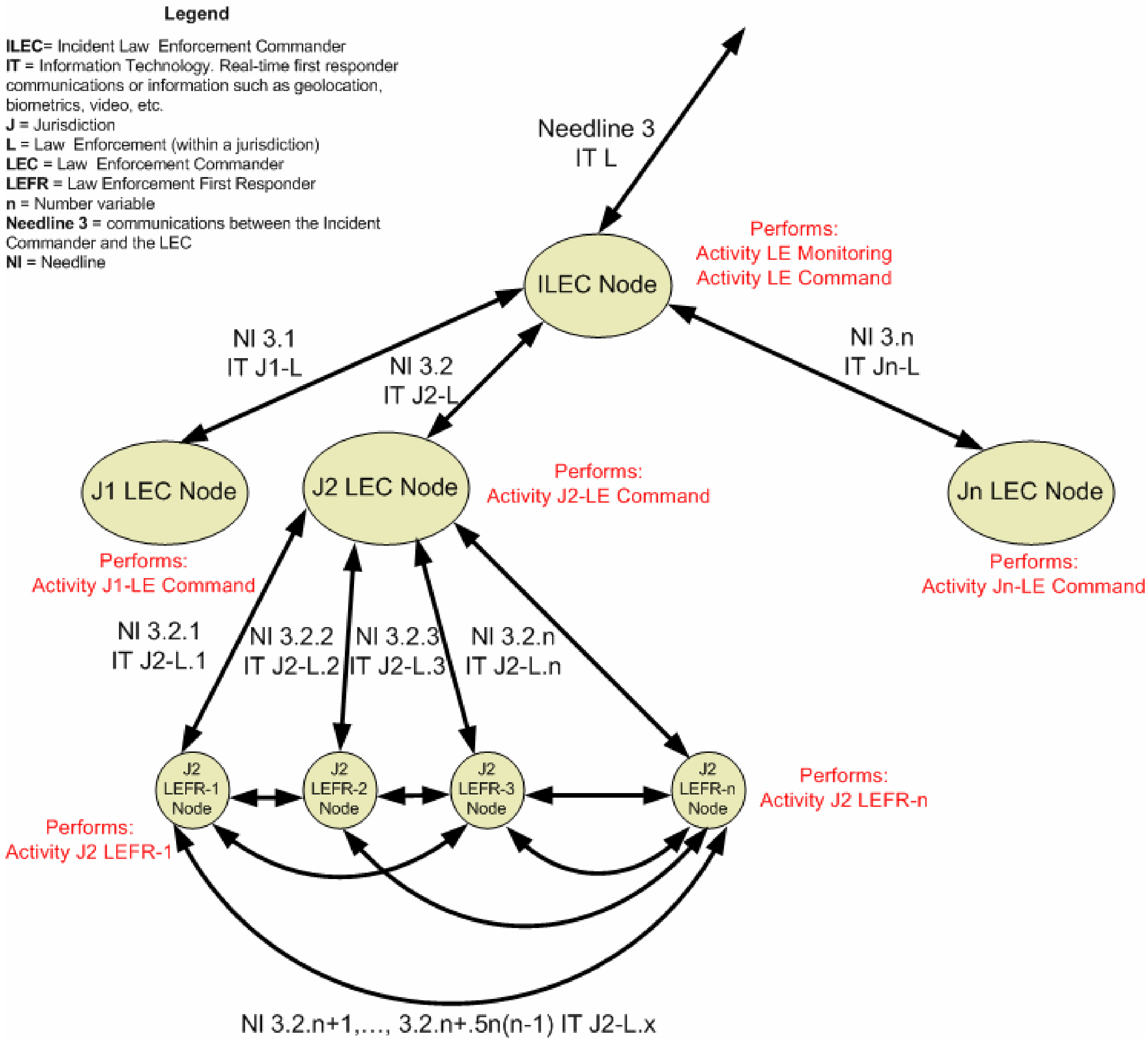
5.3.7 Representation of the Product

For complex architectures, OV-2 may consist of multiple graphics. There are at least two different ways to decompose OV-2. One method involves using multiple levels of abstraction and decomposing the nodes. Another method involves restricting the nodes and needlines on any given graphic to those associated with a subset of operational activities. Both of these methods are valid and can be used them together.

OVs usually avoid representing real physical facilities as operational nodes, and focus on virtual or logical nodes you can base on operational or human roles and missions. Operational nodes are independent of material considerations. Indeed, they exist to fulfill the missions of the enterprise and to perform its tasks and activities, such as business processes, procedures, and operational functions. Use of operational nodes supports analysis and design by separating business process modeling and information requirements from the material solutions that support them. Similarly, you organize tasks and activities, and define communities of interest to suit the mission and process requirements. The material is automatically configurable to support the operational processes. However, an OV often has material constraints and requirements you must address.

Where appropriate, system or physical nodes that constitute the location of an operational node may augment its description. These are often taken as recommendations or boundaries for further SV details. Figure 10 provides an example of an OV-2. Section 5.1 identifies the incident scenario referenced in Figure 10.

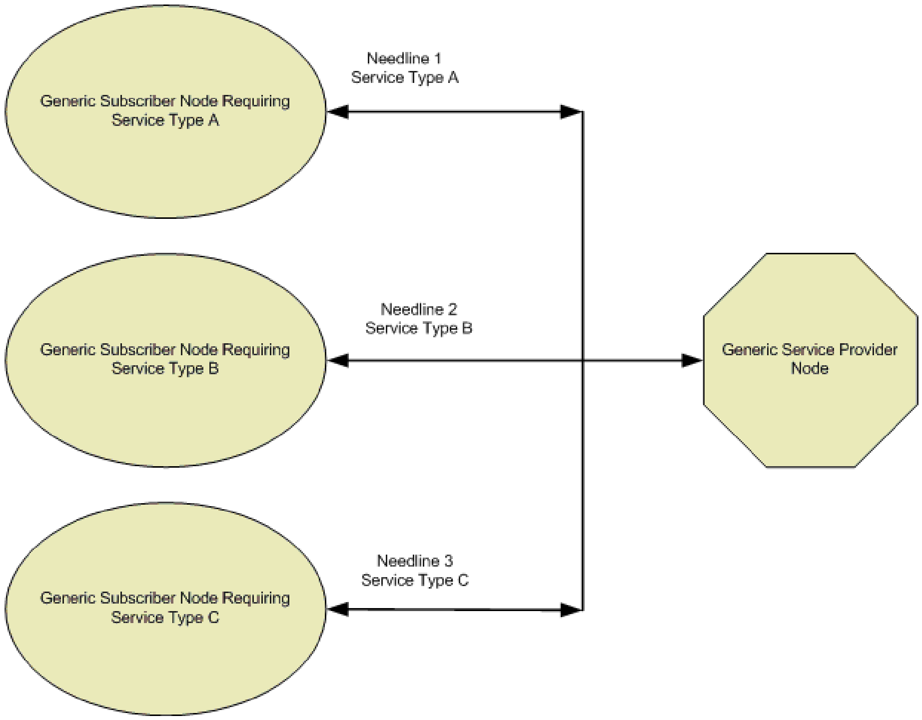
Figure 10: OV-2 Example



Service provider architectures also use a type of logical node. One purpose of a service provider architecture can be to communicate an external view of the available services to potential subscriber communities. In this situation, the service provider's OV-2 (and OV-3) can use generic representations of the subscriber environments it supports and, potentially, of the service provider facilities as well.

For the service provider, needlines may focus on the characteristics of the service provided or on a generic type of information to be exchanged, and not on the exact type or critical attributes of the actual information exchanged. What is represented depends on the type of service provided. For example, a communications service provider will describe needlines in terms of the type of information to be transferred, with reliability or priority and security features. A human resources (HR) services provider will describe needlines in terms of the complete set of HR information produced or consumed; however, any given subscriber may only deal with a subset of this information. Figure 11 is a notional example of service providers and subscribers.

Figure 11: Notional Example of an OV-2 Depicting Service Providers and Subscribers

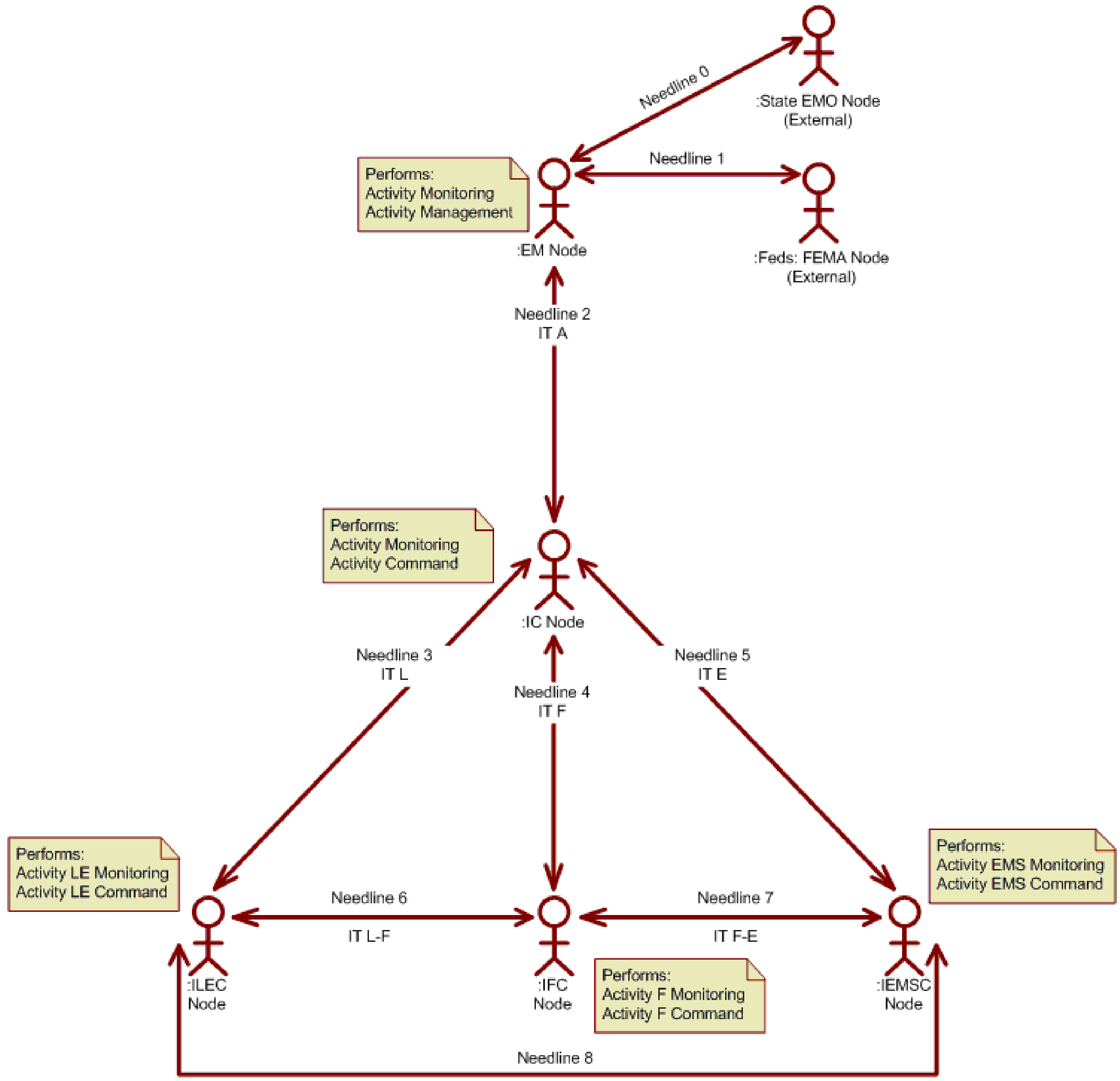


Subscribers can use information from the service provider's OV-2 to build the portions of their own OV-2 that include use of services from the service provider. The subscribers fill in the blanks or make specific the relevant generic portions of the service provider's OV-2.

5.3.8 UML Representation

You can develop OV-2, describing operational (logical, organization, role) nodes, in UML using collaboration diagrams. This application of a collaboration diagram does not represent a use case. The actors represent operational nodes, and they are the same actors appearing on an OV-5 use case diagram. You can display these operational nodes using the actor icon or a class icon, but an actor icon is preferable as it visually emphasizes the nature of the actor (i.e., the operational node) responsible for carrying out the related OV-5 operational activities. The OV-5 use case diagram emphasizes the operational activities that need to be conducted. OV-2's corresponding collaboration diagrams emphasize who (the actors) and what (the UML links [needlines] and UML messages [information exchanges] exchanged between them). Figure 12 is an example of such a diagram.

Figure 12: UML OV-2 Example



OV-2 is a graphical representation of information exchanges between operational nodes within and outside the architecture, where operational nodes represent human roles and organizations. You can model such a product using collaboration diagrams, where actors (instances of those in OV-5's use case diagrams) represent the roles or organizations that communicate via UML links and UML messages (needlines and information exchanges). The purpose here is to delineate the need to collaborate to exchange information. Operational nodes also correlate to classes on the corresponding OV-4 UML class diagram (which may be represented using the actor icon instead of the class icon). Section 5.1 identifies the incident scenario referenced in Figure 12.

5.3.9 Data Element Definitions

OV-2 focuses on the operational nodes, the needlines between them, and the type of information associated with the needline. You can also note associated operational activities. Table 4 describes the architecture data elements for OV-2.

Table 4: Data Element Definitions for OV-2

Data Element	Attributes	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
	Level Identifier	If using hierarchical decomposition of nodes: identifier that corresponds to the node's place in the node hierarchy – should be unique
External Operational Node		An operational node that is outside the scope of the architecture
	Name	Name of external operational node (i.e., label on diagram)
	Description	Textual description of the role performed by the external operational node
Graphical Arrow Types		
Needline*		
	Identifier*	Unique identifier of the needline (may be a number)
	Descriptive Name*	Descriptive name for the needline, usually associated with the types of information associated with the needline
	Description	Text description of the needline
	From Operational Node*	Name of the node box that is the source of the node connector on the diagram (if designated as an external node, then destination node must be an internal node)

Data Element	Attributes	Explanation
	To Operational Node*	Name of the node box that is the destination of the node connector on the diagram (if designated as an external node, then destination node must be an internal node)
Hierarchy Chart Connector		
	Parent Operational Node	Name or level identifier of a node that has a decomposition
	Child Operational Node	Name or level identifier of a child or subordinate node
Referenced Types		
Operational Activity		See OV-4 Definition Table.
Organization		See OV-4 Definition Table.
Organization Type		See OV-4 Definition Table.
Human Role		See OV-4 Definition Table.
Relationships		
Operational Node is Decomposed into Sub-Operational Nodes		
	Operational Node Name	Name or identifier of a decomposed operational node
	Sub-Operational Node Name	Name or identifier of a sub-operational node
Operational Node Represents Organization*		
	Operational Node Name*	Name/identifier of operational node representing the organization
	Organization Name*	Name/identifier of organization represented by organizational node
Operational Node Represents Human Role		
	Operational Node Name	Name/identifier of operational node representing the human role
	Human Role Name	Name/identifier of human role represented by operational node

Data Element	Attributes	Explanation
Operational Node has Associated Operational Activity		
	Operational Node Name	Name/identifier of operational node where the operational activity is performed
	Operational Activity Name	Name/identifier of operational activity associated with operational node

5.3.10 AFDM Support

GJXDM is being proposed for the AFDM. See Section 3.7 for information.

5.4 Operational Information Exchange Matrix (OV-3)

5.4.1 Product Definition

The Operational Information Exchange Matrix details information exchanges and identifies *"who* exchanges *what* information, with *whom*, *why* the information is necessary, and *how* the information exchange must occur." There is not a one-to-one mapping of OV-3 information exchanges to OV-2 needlines. Instead, you can associate many individual information exchanges with one needline.

5.4.2 Product Purpose

Information exchanges express the relationship across the three basic architecture data elements of an OV (operational activities, operational nodes, and information flow) with a focus on the specific aspects of the information flow and the information content. Certain aspects of the information exchange can be crucial to the operational mission and should be tracked as attributes in OV-3.

5.4.3 Product Detailed Description

OV-3 identifies information elements and relevant attributes of the information exchange, and associates the exchange to the producing and consuming operational nodes and activities, as well as to the needline that the exchange satisfies.

Information exchange is an act of exchanging information between two distinct operational nodes and the characteristics of the information exchange, including the information element that needs to be exchanged and the attributes associated with the information element (e.g., scope). The characteristics also include attributes associated with the exchange (e.g., transaction type). A needline represents one or more information exchanges.

Information element is a formalized representation of information subject to an operational process (e.g., the information content that is required to be exchanged between nodes). In contrast, an information exchange is comprised of the needline, the information element, and other attributes. The specific attributes of the information elements included are dependent on the objectives of the specific architecture effort. They also include the information scope and accuracy. You can use an information element in one or more information exchanges.

It is important to note that OV-3 is not intended to be an exhaustive listing of all the details contained in every information exchange of every operational node associated with the architecture in question. Nor should the production of such a matrix be considered sufficient to replace an integrated architecture development effort. Instead, this product is intended to capture the most important aspects of selected information exchanges. This product's emphasis is on the logical and operational characteristics of the information.

Figure 13 shows a representative format for OV-3 Operational Information Exchange Matrix.

Figure 13: OV-3 – Template

Needline Identifier	Information Exchange Identifier	Information Element Description				Producer		Consumer	
		Info. Element Name and ID	Content	Scope	Accuracy	Send. Op Node Name and Id.	Send. Op Activity Name and Id.	Rec. Op Node Name and Id.	Rec. Op Activity Name and Id.

Nature of Transaction				Performance Attributes		Information Assurance				
Transaction Type	Triggering Event	Interop Level Required	Criticality	Periodicity	Timeliness	Access Control	Availability	Confidentiality	Dissemination Control	Integrity

In Figure 13, each information exchange is associated with the needline it helps satisfy. Many individual exchanges may collectively satisfy a single needline. Note that each information element exchanged is related to the leaf operational activity (from the Operational Activity Model [OV-5]) that produces or consumes it. However, there may not be a one-to-one correlation between information elements listed in the matrix and the information inputs and outputs that connect activities in a related OV-5.

Information inputs and outputs between activities performed at the same node (i.e., not associated with a needline on OV-2) will not display in OV-3. Information inputs and outputs between activities for some levels of operational activity decomposition may be at a higher level of abstraction than the information elements in the matrix. In such a case, multiple information exchanges will map to a single operational activity input or output. Similarly, the information inputs and outputs between activities at a low level of activity decomposition may be at a higher level of detail than the information exchanges in the matrix, and multiple information inputs and outputs may map to a single information exchange. Information elements trace to the entities in OV-7.

5.4.4 UML Representation

There is no equivalent diagram to the OV-3 product in UML. The information exchanges of OV-3 trace to the collaboration or sequence diagram information flows supporting OV-5 and depicted in OV-2 and OV-6c products.

This product consists of a detailed table that expands on the information associated with OV-2 operational nodes, OV-5 operational activities, and OV-6 elements. If an automated tool creates the other products in UML, then the OV-3 expanded definitions can be attached to the applicable elements by using adornments and the documentation facilities n the UML tool. (Adornments are textual or graphical items added to an element's basic notation, and used to visualize details from the element's specification.) As a result, a script may be developed that will automatically generate the matrix.

5.4.5 Data Element Definitions

Figure 13 illustrates how the end-product should look to the user. Table 5 describes the architecture data elements for OV-3.

Table 5: Data Element Definitions for OV-3

Data Element	Attributes	Explanation
Non-Graphical Types		
Information Exchange*		
	Needline Identifier*	Identifier for the needline that carries the exchange
	Information Exchange Identifier*	Identifier for the information exchange – usually based on the relevant needline identifier; should be unique for the architecture
	Sending Operational Node Name*	Name/identifier of operational node that sends the information
	Sending Operational Activity Name*	The identity of the operational activity sending the information
	Receiving Operational Node Name*	Name/identifier of the operational node that receives the information
	Receiving Operational Activity*	The identity of the operational activity receiving the information
	Mission/Scenario	Single/multi-jurisdiction, single/multi-discipline, related specific scenario, or other task-related basis of the architecture
	Transaction Type*	Description field that identifies the type of exchange

Data Element	Attributes	Explanation
	Triggering Event*	Brief textual description of the events that trigger the information exchange
	Interoperability Level Required*	Level of interoperability required
	Criticality*	The criticality of the information exchanged in relationship to the mission performed
	Periodicity*	How often the information exchange occurs; may be an average or a worst case estimate and may include conditions (e.g., emergency or non-emergency)
	Timeliness*	Required maximum allowable time in seconds of exchange from node to node
	Access Control	The class of mechanisms used to ensure that only those authorized can access the information
	Availability	The relative level of effort to ensure that the information can be accessed
	Confidentiality	The kind of protection required for information to prevent unintended disclosure
	Authorization	The kind of restrictions on receivers of the information based on sensitivity of information
	Integrity	The kind of requirements for checks that the content of the information has not been altered
	Accountability	Security principle that ensures that responsibility for actions/events can be given to an organization willingly or by obligation
	Protection Type Name	The name for the type of protection
	Protection Duration	How long the information must be safeguarded
	Projection Suspense Calendar Date	The calendar date on which the designated level of safeguarding discontinues
Information Element*		Information to be exchanged between nodes
	Identifier*	Identifier for the information element – usually based on the relevant information exchange identifier; should be unique for the architecture
	Name*	Name for the information element – indicative

Data Element	Attributes	Explanation
		of the information content
	Content*	The content of the information element (i.e., actual information to be exchanged)
	Scope	Text description of the extent or range of the information element content
	Accuracy*	Description of the degree to which the information conforms to actual fact, as required by operational node
	Language	Identifier/name of the natural languages in the information exchange; relevant for border operations
Referenced Types		
Needline*		See OV-2 Definition Table.
Operational Node*		See OV-2 Definition Table.
Operational Activity*		See OV-5 Definition Table.
Triggering Event*		See OV-6c Definition Table.
Relationships		
Needline Supports Information Exchange*		
	Needline Identifier*	Unique needline identifier from corresponding OV-2 needline
	Information Exchange Identifier*	Identifier of an information exchange supported by the needline
Information Exchange has Producing Operational Node*		
	Operational Node Name*	Name of the operational node that produces the information exchange
	Information Exchange Identifier*	Identifier of the information exchange produced
Information Exchange has Consuming Operational Node*		
	Operational Node Name*	Name of the operational node that consumes the information exchange
	Information Exchange Identifier*	Identifier of the information exchange

Data Element	Attributes	Explanation
		consumed
Information Exchange has Producing Operational Activity*		
	Operational Activity Name*	Name of the operational activity, at the originating node of the needline, that produces the information exchange
	Information Exchange Identifier*	Identifier of the information exchange produced
Information Exchange has Consuming Operational Activity*		
	Operational Activity Name*	Name of the operational activity (at the originating node of the needline) that consumes the information exchange
	Information Exchange Identifier*	Identifier of the information exchange consumed
Information Element is Exchanged Via Information Exchange*		
	Information Element Identifier*	Identifier of the information element
	Information Exchange Identifier*	Identifier of the information exchange
Information Exchange has Triggering Event*		
	Event Name*	Name of the event that triggers the information exchange
	Information Exchange Identifier*	Identifier of the information exchange

5.4.6 AFDM Support

GJXDM is being proposed for the AFDM. See Section 3.7 for information.

5.5 Organizational Relationships Chart (OV-4)

5.5.1 Product Definition

The Organizational Relationships Chart (OV-4) illustrates the command structure or relationships among human roles, organizations, or organization types that are the key players in an architecture. This differs from relationships related to a business process flow.

5.5.2 Product Purpose

This product clarifies the various relationships that can exist between organizations and suborganizations within the architecture and between internal and external organizations.

5.5.3 Product Detailed Description

OV-4 illustrates the relationships among organizations or resources in an architecture. These relationships can include supervisory reporting, command and control relationships, and command subordinate relationships. Another type of relationship is a coordination relationship between equals, where two organizations coordinate or collaborate without one having a supervisory or command relationship over the other. You can define other relationships depending on the purpose of the architecture.

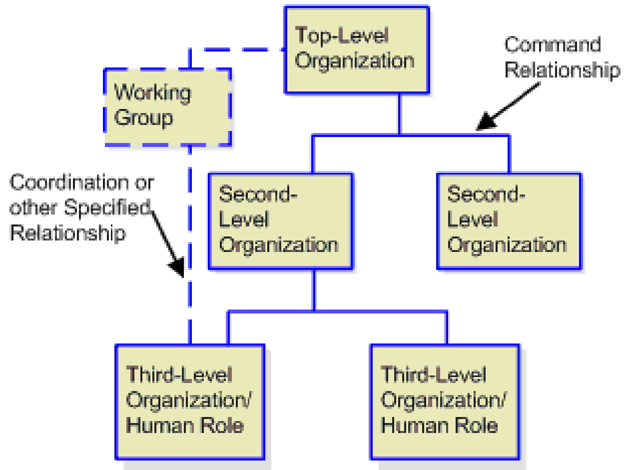
Architects should feel free to define any kinds of relationships necessary and important within their architecture to support its goals. For example, dynamic teams or task forces (i.e., new operational nodes) for instance in a multi- emergency, for example, may be created in real time with only limited life spans and assigned missions. They also could have needlines assigned to them. The creating node and the created node have a unique relationship you should document. This relationship may not be one of lines of command or organizational hierarchies, as these do not necessarily map to the needlines of OV-2. In this product, the dynamic organizations represented by operational nodes in OV-2 have a limited lifespan and a temporary collaboration relationship.

The product illustrates the relationships among organizations or organization types that are the key players in an architecture. These key players correspond to the operational nodes of an OV-2, which contains added detail on how the key players interact together to conduct their corresponding operational activities in OV-5.

You can also define human roles in OV-4 that require the skills necessary to perform the operational activities or business processes the architecture describes. Compose the corresponding operational activities to a degree that allows correlations to specific human roles within organizations. In the case of target architectures, you can use human roles that do not reflect a specific supervisory reporting, command and control, or coordination organizational structure. In this case, you can develop OV-4 using strictly human roles that are the key players in an architecture.

For a current architecture, organizational relationships are important to depict in an OV, because they can illustrate fundamental human roles such as who or what type of skill is needed to conduct operational activities, as well as management relationships. An example of the latter is the command structure or relationship to other key players. Further, organizational relationships may influence how the operational nodes in an OV-2 are connected. Figure 14 shows a template of organizational relationships.

Figure 14: OV-4 – Template

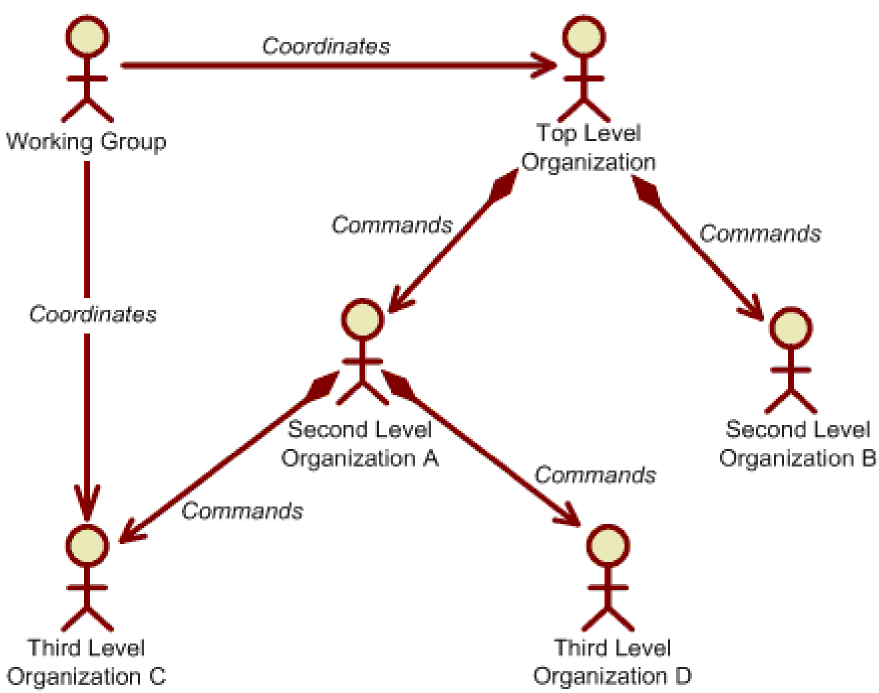


As the template illustrates, boxes can show hierarchies of organizations, and different colors or styles of connecting lines can indicate various types of relationships among the organizations.

5.5.4 UML Representation

Using the actor icon, you can use a class diagram to represent organizational relationships charts. You can use class relationship notation to show relationships among organizations. Figure 15 illustrates such a diagram.

Figure 15: UML OV-4 – Template



Classes and their relationships that are represented in these diagrams trace to the operational nodes or actors represented in an OV-2 collaboration diagram, and to the actors of the operational activity model (use case diagram). Classes in these diagrams represent organizations, and UML class relationships represent operational command, control, and organizational relationships among the organizations.

5.5.5 Data Element Definitions

Table 6 describes the architecture data elements for OV-4.

Table 6: Data Element Definitions for OV-4

Data Element	Attributes	Explanation
Graphical Box Types		
Organization*		
	Name*	Name of the organization that appears on the graphic
	Description*	Text description of the organization's purpose, including the spelling out of any acronyms
	Discipline	Fire response, law enforcement, EMS
	Role/Responsibility	Text description of the role played by the described organization
Organization Type		
	Name	Name of the organization type
	Description	Text description of the organization type
Human Role		
	Name	Name of a human role or personnel position in an organization possessing a certain skill
	Description	Text description of the human role
	Role/Responsibility	Text description of the responsibility the human role performs (skill or skill set needed to perform role)
Graphical Arrow Types		
Organizational Relationship*		
	Name/Label*	Relationship label used on graphic
	Description*	Textual description of relationship

Data Element	Attributes	Explanation
	Relationship Type*	The type of relationship documented (e.g., direct/command, indirect, situation dependent, coordination, and backup)
	Organization Name 1*	Name of source organization for relationship
	Organization Name 2*	Name of destination organization for relationship
Referenced Types		
Operational Node		See OV-2 Definition Table.
Relationships		
Organization is Represented as an Operational Node*		
	Organization Name	Name of an organization
	Operational Node Name	Name of an operational node that represents the organization in OV-2
Organization Type is Represented as an Operational Node		
	Organization Type Name	Name of an organization type
	Operational Node Name	Name of an operational node that represents the organization type in OV-2
Human Role is Represented as an Operational Node		
	Human Role Name	Name of a human role
	Operational Node Name	Name of an operational node that represents the human role in OV-2
Organization, or Organization Type has Human Roles		
	Organization, or Organization Type Name	Name of an organization or organization type
	Human Role Name	Name of a human role

5.5.6 AFDM Support

GJXDM is being proposed for the AFDM. See Section 3.7 for information.

5.6 Operational Activity Model (OV-5)

5.6.1 Product Definition

The Operational Activity Model (OV-5) describes the operations that are normally conducted in the course of achieving a mission or a business goal. It describes capabilities, operational activities or tasks, input and output (I/O) flows between activities, and I/O flows to and from activities that are outside the scope of the architecture. High-level operational activities should trace to, or are decompositions of, a business area, an internal line of business, or a business sub-function, as published in OMB's Business Reference Model (OMB, 2003).

5.6.2 Product Purpose

OV-5 is used to:

- Clearly delineate lines of responsibility for activities when coupled with OV-2.
- Uncover unnecessary operational activity redundancy.
- Make decisions about streamlining, combining, or omitting activities.
- Define or flag issues, opportunities, or operational activities and their interactions, such as information flows among the activities, that need further scrutiny.
- Provide a necessary foundation for depicting activity sequencing and timing in OV-6a, OV-6b, and OV-6c.

5.6.3 Product Detailed Description

The PSAF does not endorse a specific modeling methodology. However, if you use the Integration Definition for Function Modeling (IDEF0) method (FIPS 183, 1993), the activities also show controls: factors that affect the way the activity is performed. The activities also may show mechanisms: the resources, including operational nodes, that perform the activity. While some may illustrate corresponding systems as mechanisms in this model, keep in mind that the introduction of system data early in the development of the OV may result in limiting system design and implementation decisions.

I/Os of operational activities relate to information elements of OV-3, and are further characterized by the information exchange attributes described in OV-3. OV-2 needlines carry I/Os that are produced or consumed by leaf operational activities that cross operational node boundaries. In addition, you can annotate operational activities, for instance, through the mechanism arrow in an IDEF0 diagram, with the corresponding operational node from OV-2.

Annotations to the activities may also identify the actual or estimated costs associated with performing each activity. The business rules that govern the performance of the activities can also be keyed to each activity. (Business rules are described in OV-6a.) Annotations to OV-5s can further the purposes of the description by adding specific attributes of exchanged information, which you can use later in OV-3.

OV-5 is a key product for describing capabilities and relating capabilities to mission accomplishment. You can define a capability by one or more sequences of activities, referred to as operational threads or scenarios. You can further describe capability in terms of the attributes required to accomplish the set of activities — such as the sequence and timing of operational activities or material that enable the capability

— to achieve a given mission objective. You can associate capability-related attributes with specific activities or with the information flow between activities, or both. When represented by a set of operational activities, you can link a capability to an operational node in an OV-2.

OV-5 graphics may include a hierarchy chart of the activities covered in the model. A hierarchy chart helps provide an overall picture of activities and a quick reference for navigating the OV-5 I/O flow model.

OV-5 is frequently used with a process flow model such as an IDEF3 model or a UML sequence diagram. Such a model describes the sequence and other attributes (e.g., timing) of the activities. A process flow model further captures precedence and causality relations between situations and events by providing a structured method for expressing knowledge about how a process or organization works. In addition, you can annotate a process flow model with the names of the operational nodes responsible for conducting those activities. You can describe a process flow model in OV-6c.

You should align the decomposition levels and the amount of detail shown on OV-5 with the operational nodes responsible for conducting the operational activities. (The latter are shown on corresponding OV-2 products.) It is important to note that OV-5 is intended to be only as exhaustive as necessary to attain the objectives for the architecture, as stated in AV-1. Figure 16 depicts an example for the Operational Activity Model. Section 5.1 identifies the incident scenario referenced in Figure 16.

Figure 16: OV-5 – Example

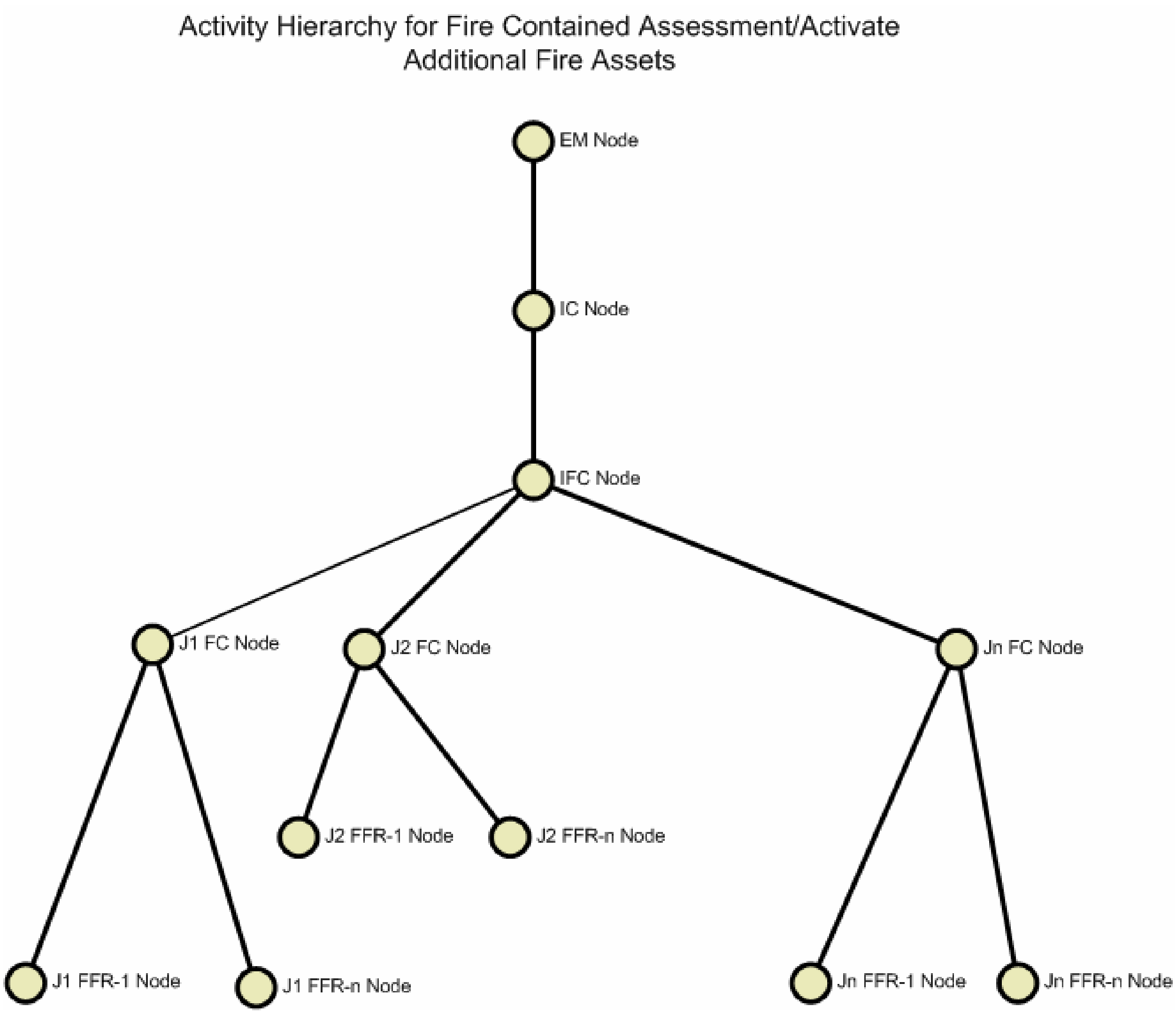
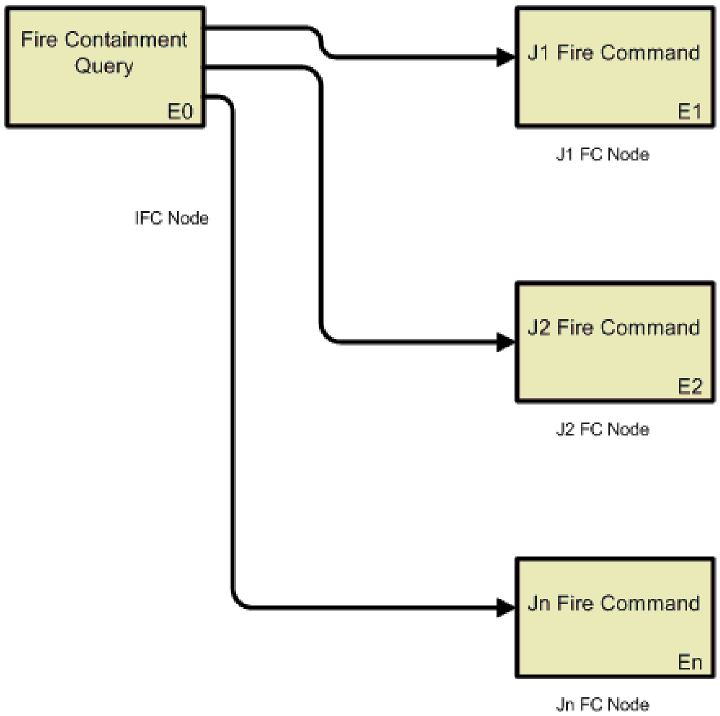


Figure 17 is a process-oriented OV-5 example showing how some architecture data could be added as annotations. For example, you can annotate activities with information concerning the operational nodes that conduct them, the material that supports them, the cost of conducting the activity, and so forth. (The types of additional architecture data are notional.) Section 5.1 identifies the incident scenario referenced in Figure 17.

Figure 17: OV-5 – Example with Notional Annotations



5.6.4 UML Representation

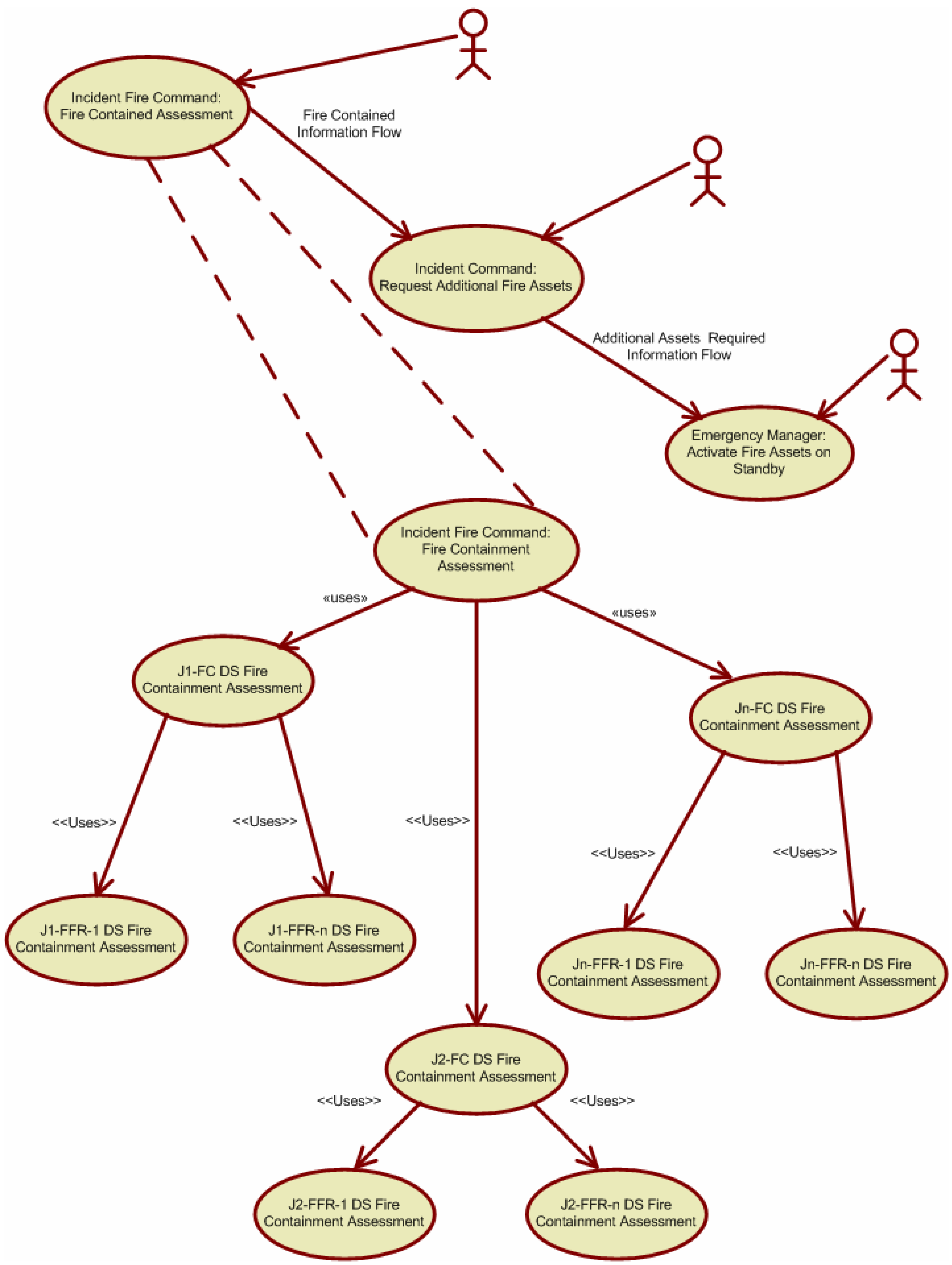
You can document OV-5 by UML use case diagrams and activity diagrams. The UML specification defines a use case as "A specification of sequences of actions including variant sequences and error sequences, that a system, subsystem or class can perform by interacting with outside actors. The elements in the use cases package are primarily used to define the behavior of an entity, like a system or a subsystem, without specifying its internal structure." (Object Management Group [OMG], 2001)

Operational use cases depict operational activities, where actors represent operational nodes, collaborating to conduct these use cases. In such a diagram, it makes sense to adapt the UML slightly. The actors are not outside the *system* boundaries as is usually the case with using UML in a system development effort, but represent entities that conduct these operational activities. The use cases representing these operational activities are not going to be implemented via classes, which is usually done within the context of a system development effort. Instead, operational use cases depict the major functions or actions at the operational level.

You can decompose these operational use cases into systems use cases (via <<include>> relationships). It is the systems use cases that then become the starting point for the Systems View, and which represent major system uses or capabilities. Supporting sequence diagrams (OV-6c) correlate to use case scenarios and describe the events that map to information exchanges passed between the operational nodes collaborating on the use case. The representation of OV-5 as a use case diagram allows the related OV-2 product to be generated automatically from the supporting sequence diagram, and ensures consistency between the interrelated OV-2, OV-3, and OV-5 products.

By way of comparison with IDEF0 diagrams: IDEF0 mechanisms become actors, and IDEF0 controls can be modeled as classes that supply a specific type (constraints) of architecture data. IDEF0 I/Os are reflected as event-action pairs appearing in sequence diagrams (OV-6c). Figure 18 is an OV-5 use case diagram example. Section 5.1 identifies the incident scenario referenced in Figure 18.

Figure 18: UML Use Case Diagram for OV-5 – Example

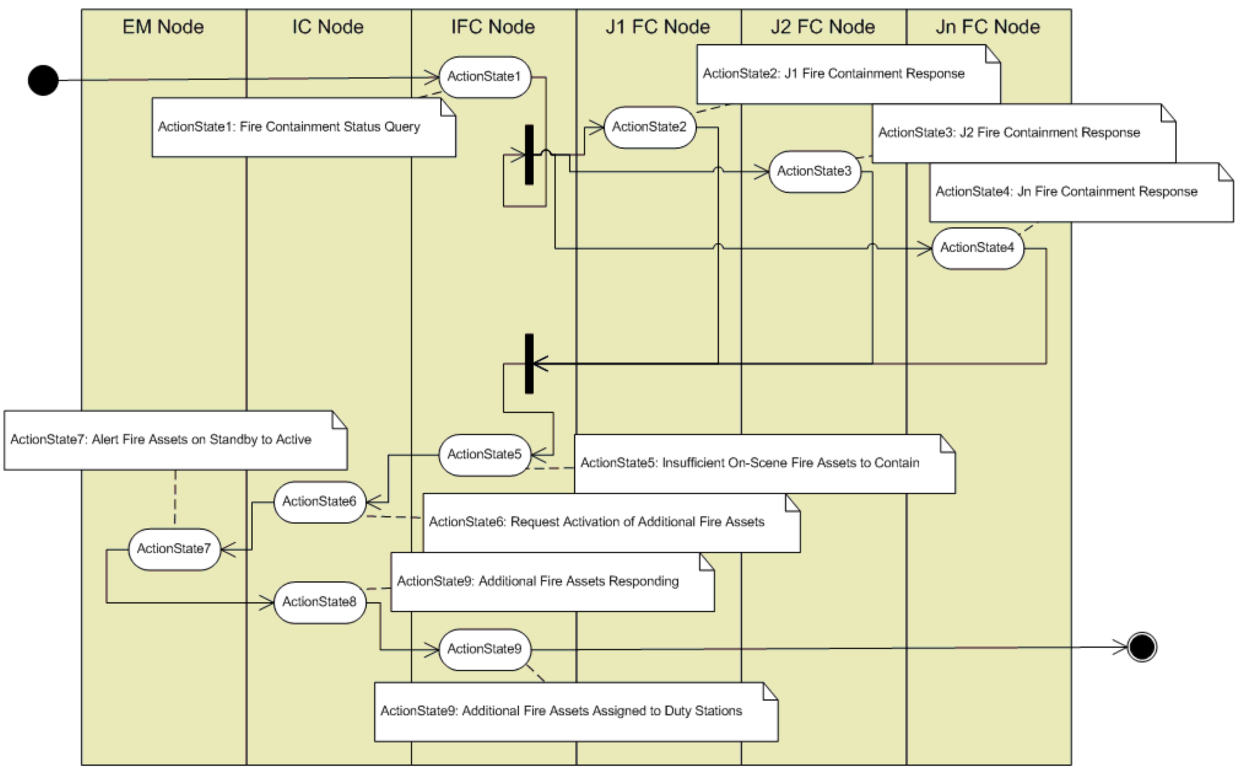


Activity diagrams in UML primarily document the sequential flow between states of an object instance, or a set of object instances. "An activity diagram can be applied to organizational modeling for business process engineering and workflow modeling. In this context, events often originate from inside the architecture, such as the completion of an activity, and also from outside the architecture, such as a customer call." (OMG, 2003)

The UML specification suggests using this diagram to support the description of a set of activities that occur internal to one object (e.g., an actor instance/one operational node). However, activity diagrams are special cases of state machines or state diagrams, and can be useful for presenting another view of the dynamic behavior of an architecture or system. By using the swim lanes variant, reflected by the columns in Figure 19, activity diagrams in UML can focus on the flow of control between activities across a number of objects (in this case, the operational nodes from OV-5). This variant of activity diagrams is recommended for use here. The activities within each swim lane relate to the use cases, and specify the sequence for the use cases that the operational node participates in or collaborates on. The activity diagram is used here to add the sequential flow aspect among operational activities to a use case diagram.

The UML activity flow diagram example for OV-5 is shown in Figure 19. Section 5.1 identifies the incident scenario referenced in Figure 19.

Figure 19: UML Activity Diagram for OV-5 – Example



To ensure model consistency, the activities in the diagram must correspond to, and provide detail for, the use cases (the operational activities of OV-5) appearing in the related use case diagrams. The objects you use to define activity diagram swim lanes must also correspond to the actors in related OV-2 collaboration diagrams — the same actors as in the related OV-5 use case diagrams.

5.6.5 Data Element Definitions

Table 7 describes the architecture data elements for OV-5.

Table 7: Data Element Definitions for OV-5

Data Element	Attributes	Explanation
Graphical Box Types		
Operational Activity*		
	Name*	Name of activity
	Level Identifier	If using hierarchical decomposition of activities, identifier that corresponds to the activity's place in the activity hierarchy; should be unique
	Description*	Description of the activity
	References	Any local policy references that provide further explanation of the activity
	Operational Activity Cost	Cost for activity derived from or used in activity-based cost analysis
Graphical Arrow Types		
Hierarchy Chart Connector		
	Parent Operational Activity*	Name or level identifier of an activity that has a decomposition
	Child Operational Activity	Name or level identifier of child or subordinate activity
Flow Connector*		(For process-oriented models)
	Name*	Label of connector on graphic
	Type*	If using IDEF0: one of input, output, control, mechanism
For Input/Output Flow Connector*		
	Source*	Name of source activity or external
	Destination*	Name of destination activity or external
	Operational Activity I/O Information Name*	Name of the information associated with the flow
For Subtype Control		(For IDEF0 models, if any)

Data Element	Attributes	Explanation
	Source	Name of source activity or External
	Destination	Name of destination activity
	Control Description	Description of the control information
For Subtype Mechanism		(For IDEF0 models, if any)
	Source	Name of source activity, if relevant, otherwise not available
	Destination	Name of destination activity
	Resource Type	Type of resource represented: operational node, systems node, or system
Non-Graphical Types		
Capability		May be defined in terms of the attributes required to accomplish a set of activities to achieve a given mission objective; capability-related attributes may be associated with specific activities
	Name	Name of capability
	Description	Description of the capability
Model		
	Name	Name/identifier of operational activity model
	Type	Operational activity hierarchy, IDEF0-style model, object-oriented, 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)
Facing Page Text		
	Identifier	Identifier/title of a page of text
	Text	Text description of the content of a graphic/diagram and its constituent parts

Data Element	Attributes	Explanation
Operational Activity I/O Information*		
	Name*	Name of input or output information for an activity
	Content*	The information content (information that is input or output by the activity)
Relationships		
Capability is Related to Operational Activity		
	Capability	Name or identifier of a capability
	Operational Activity	Name or identifier of an operational activity
Operational Activity I/O Information is Associated with an Information Exchange		(Information produced or consumed by an activity may be associated with multiple information exchanges)
	Operational Activity I/O Information Name	Name or label of information that is an input or an output of an activity
	Information Exchange Identifier	Identifier for an associated information exchange (from OV-3)
Operational Activity is Performed by Operational Node*		
	Operational Activity Name*	Name/identifier of an operational activity
	Operational Node Name*	Name/identifier of the operational node that performs the operational activity
Capability is Related to Operational Activity		
Operational Node, Systems Node, or Systems is Represented by Mechanism		May be many to many
	Operational Node, Systems Node, or System Name	Name of an operational node, systems node, or system that is represented by the mechanism
	Mechanism Name	Name of mechanism representing the operational node, systems node, or system
Diagram Belongs to Model		
	Diagram Title	Title of a diagram

Data Element	Attributes	Explanation
	Model Name	Name of the model to which the diagram belongs
Facing Page Text Reference Diagram		
	Facing Page Text Identifier	Identifier/title for a page of text
	Diagram Title	Title of the diagram that the text describes
Operational Activity is Contained in Diagram		
	Operational Activity Name	Name/identifier of an activity
	Diagram Title	Title of the diagram on which the activity occurs
Flow Connector is Contained in Diagram		
	Flow Connector Name	Label of connector
	Diagram Title	Title of diagram on which the connector appears
Operational Activity has Input*		
	Operational Activity*	Name or level identifier of an activity
	Operational Activity I/O Information Name*	Name or label of information or product that is an input to the activity
Operational Activity has Output*		
	Operational Activity*	Name or level identifier of an activity
	Operational Activity I/O Information Name*	Name or label of information or product that is an output from the activity
Operational Activity is Decomposed into Sub-Operational Activity		
	Operational Activity	Name or identifier of a decomposed operational activity
	Sub-Operational Activity	Name or identifier of a sub-operational activity
Flow Connector Corresponds to Flow Connector		

Data Element	Attributes	Explanation
	Child Flow Connector Name	Name of flow connector on boundary of child diagram
	Parent Flow Connector Name	Name of the corresponding flow connector on parent diagram

5.6.6 AFDM Support

GJXDM is being proposed for the AFDM. See Section 3.7 for information.

5.7 Operational Activity Sequence and Threads (OV-6)

5.7.1 Product Definition

The Operational Activity Sequence and Threads (OV-6) products describe the dynamic behavior concerning the timing and sequencing of events that capture the operational behavior of a business process or mission thread. OV products discussed in previous sections model the static structure of the architecture elements and their relationships. However, many of the critical characteristics of an architecture are only discovered when you model the dynamic behavior of these elements to incorporate sequencing and timing aspects.

The dynamic behavior referred to here concerns the timing and sequencing of events that capture, for example, the operational behavior of a business process or mission thread. Thus, this behavior is related to the activities of OV-5.

5.7.2 Product Purpose

Behavior modeling and documentation is essential to a successful architecture description, because it is how the architecture behaves that is often crucial. Knowledge of the operational nodes, activities, and information exchanges is crucial; however, knowing when, for example, a response should be expected after sending message X to node Y, can also be crucial to successful operations.

5.7.3 UML Representation

You can use several modeling techniques to refine and extend the architecture's OV to adequately describe the architecture's dynamic behavior and timing performance characteristics, including logical languages such as LDL (Naqvi, 1989) Harel Statecharts (Harel, 1987 a, b), Petri nets (Kristensen, 1998), IDEF3 diagrams (IDEF3, 1995), and UML statechart and sequence diagrams (OMG, 2001). The OV-6 product includes three such models. They are:

- Operational Rules Model (OV-6a)
- Operational State Transition Description (OV-6b)

- Operational Event-Trace Description (OV-6c)

5.7.4 Data Element Definitions

OV-6 products portray some of the same architecture data elements, but each also portrays some unique ones. You can use OV-6b and OV-6c separately or together, as necessary, to describe critical timing and sequencing behavior in the OV. Both types of products are used by a wide variety of different business process methodologies as well as object-oriented methodologies.

OV-6b and OV-6c describe operational activity or business process responses to sequences of events. Events may also be referred to as inputs, transactions, or triggers. Events can be internally or externally generated, and can include such things as the receipt of a message, a timer going off, or conditional tests being satisfied. When an event occurs, the action to be taken may be subject to a rule or set of rules or conditions, as described in OV-6a.

5.7.5 AFDM Support

GJXDM is being proposed for the AFDM. See Section 3.7 for information.

5.8 Operational Rules Model (OV-6a)

5.8.1 Product Definition

The Operational Rules Model (OV-6a) specifies operational or business rules constraints on an enterprise, a mission, operation, business, or an architecture. While other OV products (e.g., OV-1, OV-2, and OV-5) describe the structure of a business — what the business can do — for the most part, they do not describe what the business *must* do, or what it *cannot* do.

At the mission level, OV-6a may consist of guiding principles or policy. At the operation level, rules may include such things as an operational plan. At lower levels, OV-6a describes the rules under which the architecture or its nodes behave under specified conditions. You can express such rules in a textual form, for example, “If (these conditions) exist, and (this event) occurs, then (perform these actions).”

5.8.2 Product Purpose

At a top level, rules should at least embody the concepts of operations defined in OV-1. In addition, they should provide guidelines for the development and definition of more detailed rules and behavioral definitions that will occur later in the architecture definition process.

5.8.3 Product Detailed Description

Rules are statements that define or constrain some aspect of the mission, or the architecture. They are intended to assert operational structure or to control or influence the mission thread. As the product name implies, the rules captured in OV-6a are operational — mission-oriented — not systems-oriented. These rules can include such guidance as the conditions under which operational control passes from one entity to another. They can also include the conditions under which a human role is authorized to proceed with a specific activity.

OV-6a can be associated with the appropriate activities in OV-5. For example, a rule might prescribe the specific set of inputs required to produce a given output. You can also use OV-6a to extend the capture of business requirements by constraining the structure and validity of OV-7 elements.

Detailed rules can become quite complex, and the structuring of the rules themselves can often be challenging. OV-6a extends the representation of business requirements and the concept of operations by capturing, in the form of operational rules expressed in a formal language, both action assertions and derivations. Examples of formal languages include structured English, LDL (Naqvi, 1989), and the Object Constraint Language (OCL) (Warmer, 1999). Action assertions are constraints on the results that actions produce, such as if-then and integrity constraints. Derivations are algorithmically derived facts based on other terms, facts, derivations or action assertions.

Operational rules can be grouped into the following categories:

- Structural Assertions: These rules concern mission or business domain terms and facts that the entities and relationships of entity-relationship models usually capture. They reflect static aspects of business rules that may also be captured in OV-7.
  - Terms: Entities
  - Facts: Association between two or more terms (i.e., relationship)
- Action Assertions: These rules concern some dynamic aspects of the business and specify constraints on the results that actions produce. There are three types of action assertions:
  - Condition: This is a guard, or if portion, of an if-then statement. If the condition is true, it may signal the need to enforce or test additional action assertions.
  - Integrity Constraint: These must always be true (e.g., a declarative statement).
  - Authorization: This restricts certain actions to certain human roles or users.
- Derivations: These rules concern algorithms used to compute a derivable fact from other terms, facts, derivations, or action assertions.

OV-6a can concentrate on the more dynamic action assertions and derivations rules, because the structural assertion rules are usually captured in OV-7. Operational rules are:

- 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
- Mission/business-oriented

You can select the formal language in which to record each architecture. Ensure the formal language you select is referenced and well documented.

Figure 20 illustrates an example action assertion that might be part of OV-6a. The example is given in a form of structured English. Section 5.1 identifies the incident scenario referenced in Figure 20.

Figure 20: OV-6a – Action Assertion Example

Base facts could be:	▪ "An incident-area fire containment assessment cannot be completed until all on-scene fire assets have provided their assigned duty-locations' fire containment responses." ▪ "A finding that on-scene fire assets are insufficient to achieve fire containment must be based on completion of an incident-area fire containment assessment." ▪ "A collateral public safety risk-benefit tradeoff assessment must be based on an incident-area fire containment assessment, a finding that on-scene fire assets are insufficient to achieve fire containment, and an assessment of the probable hazards to the affected public and on-scene first responders, if the fire continues to be uncontained."
A derived fact could be:	▪ "A valid/legal order to activate additional fire assets on standby requires an unsatisfactory incident-area fire containment assessment, a finding that on-scene fire assets are insufficient to achieve fire containment and a collateral public safety risk-benefit tradeoff assessment."
The derivation used to deduce this derived fact would be:	▪ "An order to activate additional fire assets on standby is based on facts contained in the incident-area fire containment assessment, a finding that on-scene fire assets are insufficient to achieve fire containment and a collateral public safety risk-benefit tradeoff assessment."

5.8.4 OV-6a UML Representation

There is no equivalent diagram in UML. The OV-6a is a text-oriented product. It comprises business rules that apply to operational activities and entities of the logical model written in structured English. OV-6a extends the discovery and noting of business requirements and concept of operations for the use cases and activities of OV-5.

If you consider operational rules to be equivalent to complex, nested If-Then-Else and CASE statements, then you can derive these statements unambiguously from UML statechart diagrams. You can specify guard conditions for state transitions. Consequently, you can generate OV-6a via the use of adornments and the inclusion of pre- and post-conditions on the use cases of OV-5 (or guard conditions in statechart diagrams). Operational rules should trace to the constraint relationships identified in OV-5 and to the statechart diagrams for the relevant object classes, if they have been defined.

There is no UML diagram to be produced, but the Integrated Dictionary (AV-2) will contain these rules and constraints, if they are incorporated into the model as pre- and post-conditions or other adornments in the UML diagrams, where applicable.

5.8.5 OV-6a Data Element Definitions

AV-2 should capture information about the rules specified in OV-6a. For example, the dictionary should have information on the type of rule (structural assertion, action assertion, or derivation), the text for the text for the rule, and the relationship between the rules and other architecture product architecture data elements, such as activities from OV-5 or entities from OV-7. Table 8 describes the architecture data elements for OV-6a.

Table 8: Data Element Definitions for OV-6a

Data Element	Attributes	Explanation
Non-Graphical Types		
Rule Model		The model that contains a set of rules
	Rule Model Name	Name of rule model
Rule*		
	Rule Name*	Name/identifier of associated rule (e.g., of type action assertion)
	Type*	One of: structural assertion, action assertion, derivation
	Description	Textual discussion of assertion or derivation
	Text*	Text of assertion of derivation in selected formal language
	Formal Language*	Name of the formal language with which the rule is expressed
Reference Types		
Operational Activity*		See OV-5 Definition Table.
Entity		See OV-7 Definition Table.
Relationships		
Rule Model Includes Rule		
	Rule Model Name	Name of a rule model
	Rule Name	Name of a rule contained in a rule model
Rule Applies to Operational Activity*		(If any rules are related to activities)
	Rule Name*	Name of action assertion or derivation rule

Data Element	Attributes	Explanation
	Operational Activity Name*	Name of activity to which the rule applies
Rule Applies to Entity		See OV-7 Definition Table (if any rules are related to entities).
	Rule Name	Name of structural assertion rule
	Entity Name	Name of entity to which the rule applies

5.8.6 OV-6a AFDM Support

GJXDM is being proposed for the AFDM. See Section 3.7 for information.

5.9 Operational State Transition Description (OV-6b)

5.9.1 Product Definition

The Operational State Transition Description (OV-6b) is a graphical method of describing how an operational node or activity responds to various events by changing its state. The diagram represents the sets of events to which the architecture will respond by taking an action to move to a new state as a function of its current state. Each transition specifies an event and an action.

5.9.2 Product Purpose

The explicit sequencing of activities in response to external and internal events is not fully expressed in OV-5. You can use OV-6bto describe the explicit sequencing of the operational activities. Alternatively, you can use OV-6b to reflect the explicit sequencing of actions internal to a single operational activity, or the sequencing of operational activities with respect to a specific operational node.

5.9.3 Product Detailed Description

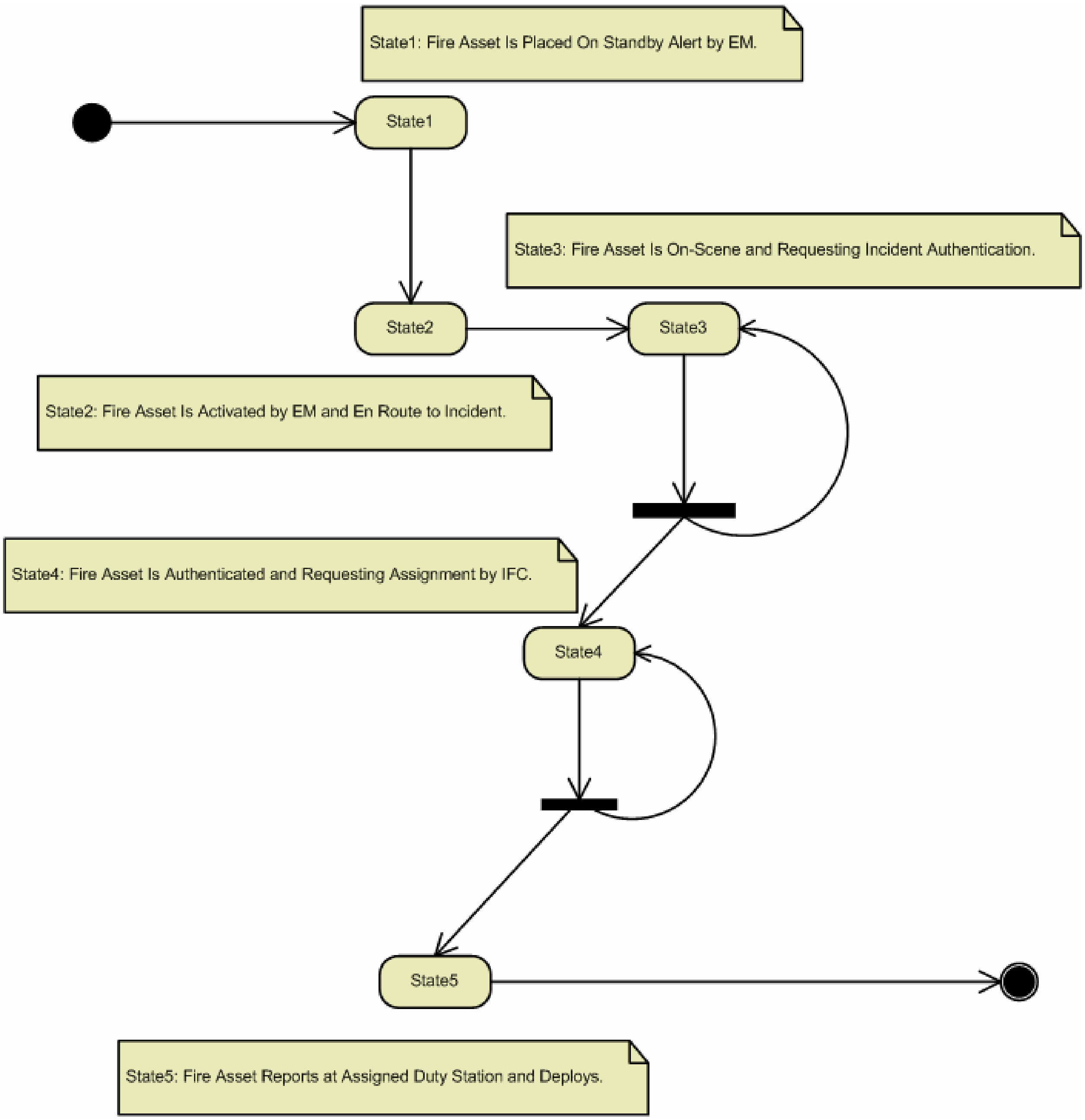
OV-6b is based on the statechart diagram. A state machine is "a specification that describes all possible behaviors of some dynamic model element. Behavior is modeled as a traversal of a graph of state nodes interconnected by one or more joined transition arcs that are triggered by the dispatching of a series of event instances. During this traversal, the state machine executes a series of actions associated with various elements of the state machine." (OMG, 2003)

The product relates states, events, and actions. A state and its associated actions specify the response of an operational activity to events. When an event occurs, the next state may vary, depending on the current state and its associated action, the event, and the rule set or guard conditions. A change of state is called a transition. Each transition specifies the response based on a specific event and the current state. Actions may be associated with a given state or with the transition between states.

You can unambiguously convert statechart diagrams to structured textual rules that specify timing aspects of operational events and the responses to these events, with no loss of meaning. However, the graphical form of the state diagrams can often allow quick analysis of the completeness of the rule set, and detection of dead ends or missing conditions. These errors, if not detected early during the operational analysis phase, can often lead to serious behavioral errors in fielded systems or to expensive correction efforts.

Figure 21 provides an example of a simple OV-6b. The black dot and incoming arrow point to initial states (usually one per diagram), while an outgoing arrow pointing to a black dot with a circle around it identifies terminal states. States are indicated by rounded corner box icons and labeled by name or number and, optionally, any actions associated with that state. One-way arrows indicate transitions between states labeled with an event/action notation that indicates an event-action pair. Semantically, an event-action pair means that when an event occurs, the corresponding action is executed. This notation indicates the event that causes the transition and the ensuing action, if any, associated with the transition. Section 5.1 identifies the incident scenario referenced in Figure 21.

Figure 21: OV-6b – High-Level Example



Another dynamic behavior modeling technique is Colored Petri Net (CPN) (Kristensen, 1998). You can use a CPN as an executable operational architectural model or a business workflow model. CPN executable models provide a description of the activity event sequencing (concurrent or consecutive). They can be used to dynamically simulate an OV-5. With a CPN simulation engine or a discrete event workflow simulation engine, you can achieve parallel event processing and decision support.

The most important reason for using both modeling and simulation tools is to show complex, dynamic organizational interactions that cannot be identified or properly understood using static models. Simulation provides insight into how processes in an enterprise add to the overall cost of a mission thread or scenario. They can make it possible to animate, analyze, and validate these complex relationships. You can use this information to create new alternatives or to compare multiple alternatives until an optimal solution is found.

Most importantly, simulation takes information that traditionally was collected in static reports, and makes it dynamic. Dynamic analysis can provide the following benefits:

- Assess time-dependent process behavior and shared time-dependent resources
- Discover ways to efficiently allocate human and system resources to perform those processes
- Check overall performance
- Identify bottlenecks and human resource overloads caused by insufficient resources or faulty information flows
- Discover and eliminate duplication of effort

In addition to examining behavior over time, you can also assess an overall dynamic mission cost over time in terms of human and system/network resource dollar costs and their processes' dollar costs. Analysis of dollar costs in executable architectures is a first step in an architecture-based investment strategy, where mangers eventually need to align architectures to funding decisions. This is to ensure that investment decisions are directly linked to mission objectives and their outcomes.

State transitions in executable operational architectural models provide for descriptions of conditions that control the behavior of process events in responding to inputs and in producing outputs. A state specifies the response of a process to events. The response may vary, depending on the current state and the rule set or conditions. Distribution settings determine process time executions. Examples of distribution strategies include: constant values, event list, constant interval spacing, normal distribution, exponential distribution, and so forth. Priority determines the processing strategy if two inputs reach a process at the same time. Higher-priority inputs are usually processed before lower-priority inputs.

Processes receiving multiple inputs need to define how to respond. Examples of responses include: process each input in the order of arrival independent of each other, process only when all inputs are available, or process as soon as any input is detected. Processes producing multiple outputs can include probabilities (totaling 100 percent) under which each output would be produced.

Histograms are examples of generated timing descriptions. They are graphic representations over time of processes, human and system resources, and their used capacity during a simulation run. These histograms are used to perform dynamic impact analysis of the behavior of the executable architecture.

5.9.4 UML Representation

You can produce the OV-6b product model in UML using statechart diagrams that contain simple states and composite states. They also contain transitions, which are described in terms of triggers or events (generated as a result of an action) and guard conditions associated with the events, and an action or sequence of actions (see Figure 21) that are executed as a result of the event taking place.

Statechart diagrams specify the reaction of an object to stimuli as a function of its internal state. Guard conditions of a statechart diagram map to the pre-conditions of an OV-5 use case. Activities in an activity diagram (OV-5) should correlate to states of the relevant object classes (operational nodes) and match to the statechart diagram states for the same object classes. In UML 1.4 a state is defined as, "A condition or situation during the life of an object which satisfies some condition, performs some activity, or waits for some event."

Transitions on the activity diagrams should correlate to the transitions on the state diagrams. In essence, an activity diagram for multiple operational nodes, or objects using swim lanes, should be able to map to a set of statechart diagrams for those operational nodes. They also should be able to be mapped to a set of sequence diagrams, where each sequence diagram correlates to a specific use case scenario. Events or triggers associated with the transitions on the state diagrams correlate to the triggering events documented in OV-3. They are the same events shown on the sequence diagrams of OV-6c.

5.9.5 Data Element Definitions

The OV-6b product model describes the detailed time sequencing of activities or workflow in the business process, depicting how the current state of a process or activity changes in response to external and internal events. Table 9 describes the architecture data elements for OV-6b.

Table 9: Data Element Definitions for OV-6b

Data Element	Attributes	Explanation
Graphical Box Types		
State*		
	Name*	State name
	Description*	Textual description as necessary
	Type*	Simple, composite, synch state, or sub state, for example
	Number of Partitions	Number of contained state charts
Graphical Arrow Types		
Transition*		
	Label*	Identifier for transition
	Description*	Textual description of transition
	Type*	For example: simple, join, fork
	Source State Name*	Name of state where transition begins
	Target State Name*	Name of state where transition ends
Non-Graphical Types		
Statechart Diagram*		

Data Element	Attributes	Explanation
	Name*	Name/identifier of statechart diagram
	Description*	Textual description of what the statechart diagram represents
	Start State Name*	Name of start state for statechart diagram
Action*		
	Name*	Name/identifier of an action that takes place while the element being modeled by the statechart is in a given state or during a state transition
	Description*	Description for the action
Trigger/Event*		Specifies the event that fires the transition. Event instances are generated as a result of some action, either within the element modeled or in the environment surrounding the element. An event is then conveyed to one or more targets.
	Name*	Name of event
	Description*	Textual description of the event
Guard		A Boolean predicate that provide a fine-grained control over the firing of the transition; must be true for the transition to be fired; evaluated at the time the event is dispatched; correlates to a rule on OV-6a.
	Name	Name/identifier for a Boolean expression that must be true for the associated transition to trigger
	Definition	Expression that defines that guard
Referenced Types		
Operational Activity*		See OV-5 Definition Table.
Rule		See OV-6a Definition Table. (correlates to a guard)
Relationships		
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

Data Element	Attributes	Explanation
State has Associated Action*		
	State Name*	Name of a state
	Action Name*	Name of the action performed while the activity is in a given state
Action is Associated with Transition*		
	Transition Name	Name of a transition
	Action Name*	Name of the action associated with the transition
Event Triggers Transition*		
	Transition Name*	Name/identifier of a transition
	Event Name*	Name of the event that triggers the transition
Transition has Guard		
	Transition Name	Name/identifier for a transition
	Guard Name	Name of associated expression (guard condition) that must be true before transition can be triggered
Transition is Associated with Source		
	Source State Name*	Name of originating state vertex (state or pseudo-state) of the transition
	Transition Name	Name of the transition
Transition is Associated with Target		
	Target State Name*	Name of target state vertex that is reached when the transition is taken
	Transition Name	Name of the transition
Guard is Associated with Rule		
	Guard Name	Name of associated expression that must be true before transition can be triggered
	Rule Name	Name of associated rule (e.g., of type action assertion)
Event Maps to Input, Output and/or		When an event represents information (I/O) or a control, whose receipt triggers a transition,

Data Element	Attributes	Explanation
Control of Operational Activity*		such an event maps to input, output, and/or control of operational activity. May be a many-to-many relationship
	Event Name	Name of event
	Input, Output, or Control Name from an Operational Activity	Name of Input, Output, or Control
Action is Related to Operational Activity*		
	Action Name*	Name/identifier of an action
	Operational Activity Name*	Name/identifier of an activity from the Operational Activity Model
	Relationship Description*	Text description of the relationship (e.g., action is same as activity, action is contained in activity, and action contains activity)

5.9.6 AFDM Support

GJXDM is being proposed for the AFDM. See Section 3.7 for information.

5.10 Operational Event-Trace Description (OV-6c)

5.10.1 Product Definition

The Operational Event-Trace Description provides a time-ordered examination of the information exchanges between participating operational nodes as a result of a particular scenario. Each event-trace diagram should have an accompanying description that defines the particular scenario or situation.

5.10.2 Product Purpose

OV-6c is valuable for moving to the next level of detail from the initial operational concepts. The product helps define node interactions and operational threads. The OV-6c can also help ensure that each participating operational node has the necessary information it needs at the right time to perform its assigned operational activity.

5.10.3 Product Detailed Description

OV-6c allows the tracing of actions in a scenario or critical sequence of events. You can use OV-6c by itself or in conjunction with OV-6b to describe the dynamic behavior of business processes or a mission or operational thread.

An *operational thread* is a set of operational activities, with sequence and timing attributes of the activities, and includes the information needed to accomplish the activities. A particular operational thread may be used to depict a capability. In this context, a *capability* is defined in terms of the attributes required to accomplish a given mission objective by modeling the set of activities and their attributes. The sequence of activities forms the basis for defining and understanding the many factors that affect the capability.

The PSAF does not endorse a specific event-trace modeling methodology. Two such types of models include UML sequence diagrams (OMG, 2003) and IDEF3 (IDEF3, 1995). You can develop the OV-6c product using any modeling notation that supports the layout of timing and sequence of activities along with the information exchanges that occur between operational nodes for a given scenario. Separate diagrams should depict different scenarios.

Figure 22 provides a template for an OV-6c using a UML diagram. The items across the top of the diagram are operational nodes -- usually organizations, organizations types, or human roles -- which take action based on certain types of events. Each operational node has a vertical lifeline associated with it. You should label specific points in time on the lifelines, running down the left-hand side of the diagram. One-way arrows between the node lifelines represent events, and the points at which they intersect the lifelines represent the times at which the nodes become aware of the events. Events represent information passed from one lifeline to another and actions associated with the event. Show labels indicating timing constraints or providing descriptions in the margin or near the event arrow that they label. The direction of the events represents the flow of control from one node to another.

Figure 22: OV-6c – UML-type Template

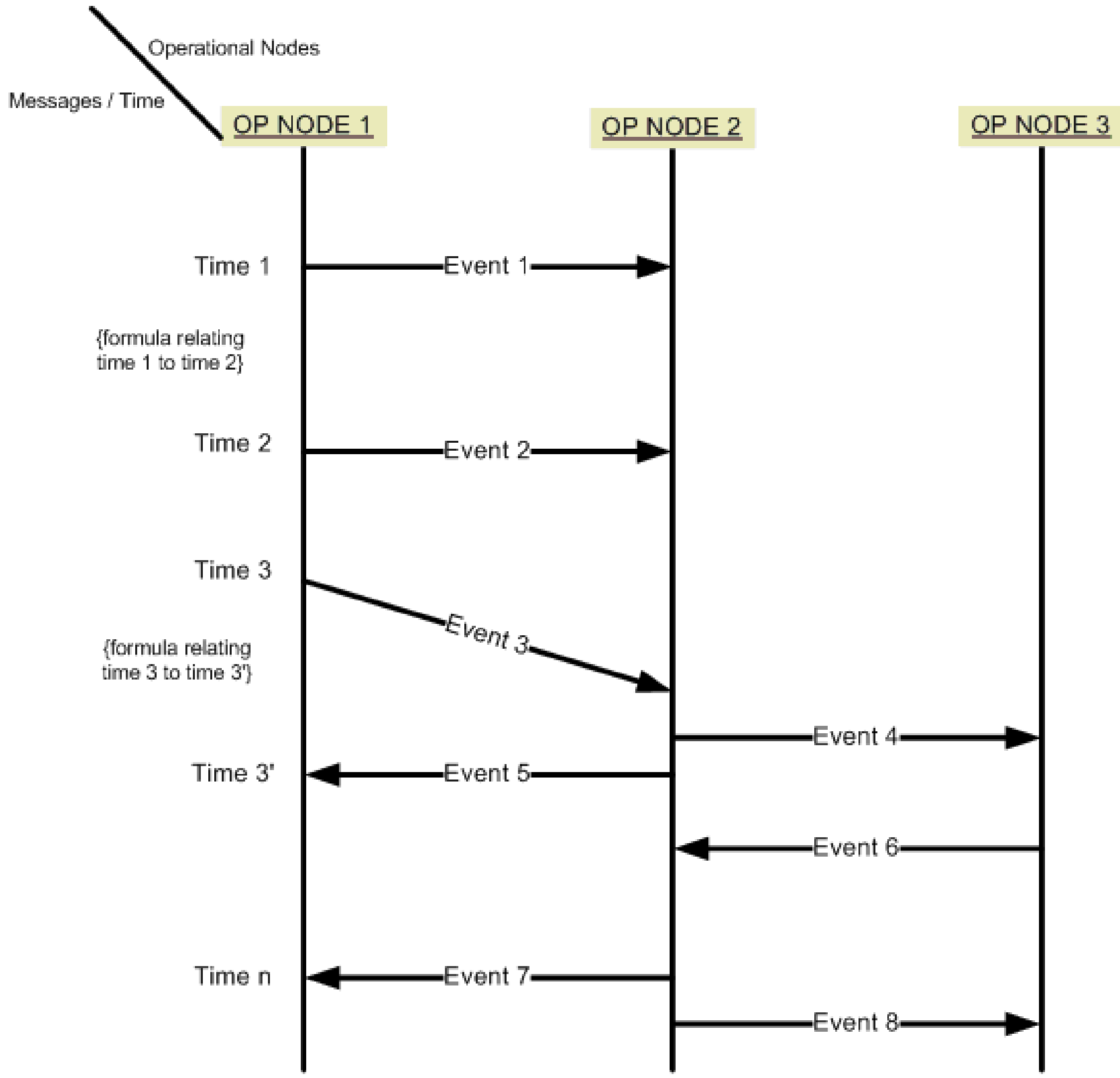
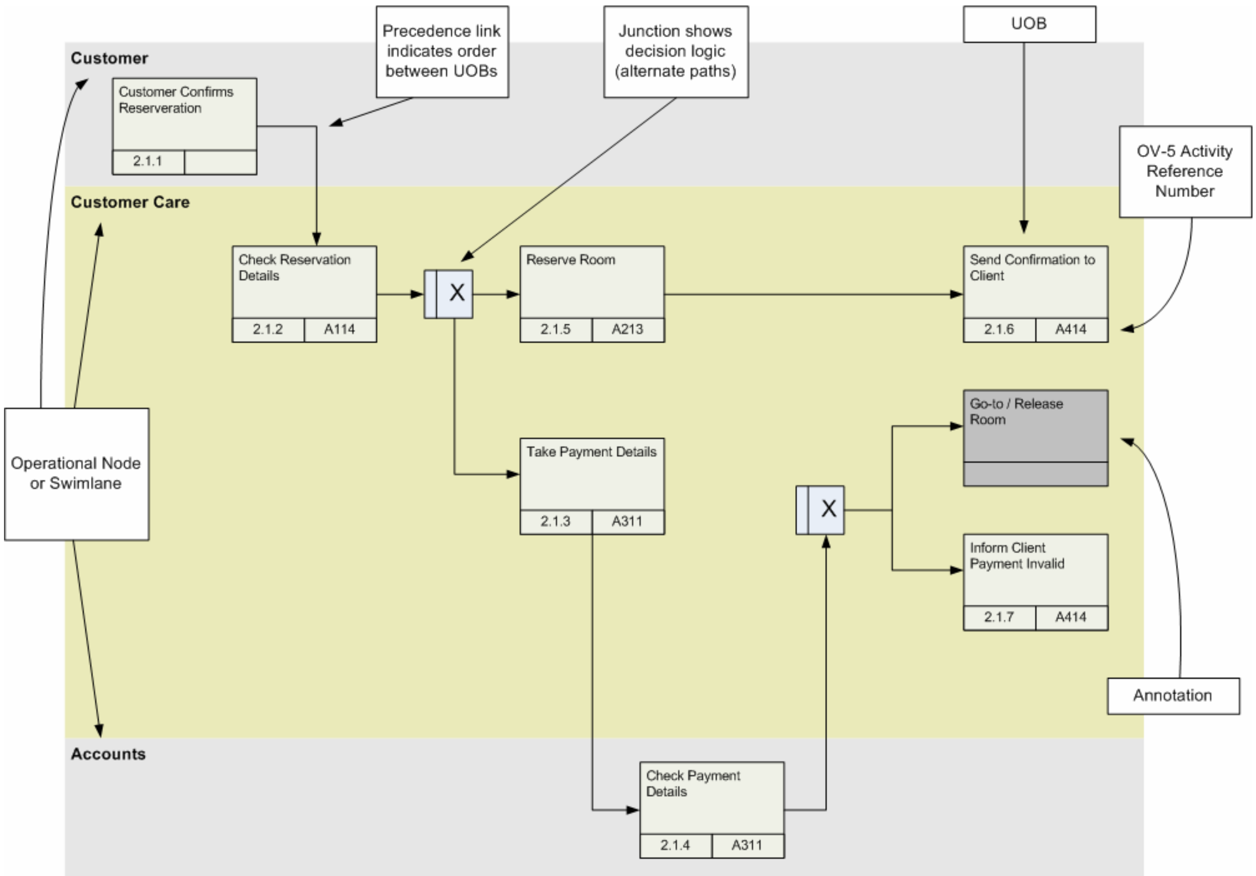


Figure 23 shows an example of an Operational Event-Trace Description using IDEF3 notation. Boxes represent a Unit of Behavior (UOB), which is a step in the process that ties to an operational activity in OV-5 via an IDEF0 reference property. A swim lane represents a horizontal or vertical division of the diagram, and shows what operational nodes perform what processes (UOBs). Links show control flow between UOBs, and not information flow. IDEF3 link types include temporal (sequence, precedence) and logical (guard conditions or business rules).

Figure 23: OV-6c – IDEF3 Example



5.10.4 UML Representation

You can produce the OV-6c product model in UML using sequence diagrams. (see Section 5.10.3 for a description and Figure 22 for a template.) The objects appearing as lifelines in OV-6c are instances of the actors appearing in the use case diagrams of OV-5. They map to the same instances depicted in OV-2 collaboration diagrams. Objects or actor instances referenced in the sequence diagrams should also be traceable to OV-4 class diagrams.

5.10.5 Data Element Definitions

Table 10 describes the architecture data elements for the OV-6c product model.

Table 10: Data Element Definitions for OV-6c

Data Element	Attributes	Explanation
Graphical Box Types		
Lifeline or Swim Lane*		
	Name*	Name of the lifeline or swim lane
	Description*	Text description of any assumptions or scope constraints on the node
Operational Activity		If using IDEF3; see OV-5 Definition Table (referred to as task in IDEF3).
Graphical Arrow Types		
Event*		
	Name*	Event label or name of event
	Description*	Textual description of event
	Originating Node Name*	Name of node where event begins
	Terminating Node Name*	Name of node where event ends
Link		If using IDEF3
	Name	Name of link
	Description	Textual description of link
	Type	Type of link – temporal or logical
	Originating Activity Name	Name of activity where link begins
	Terminating Activity Name	Name of activity where link ends
Non-Graphical Types		
Operational Thread		
	Name	Name/identifier of operational thread or scenario

Data Element	Attributes	Explanation
	Type	Model type used to develop operational thread (e.g., IDEF3-style model, UML sequence diagram, or other type of model)
	Capability Name	Name of capability the operational thread supports
Event Time*		
	Identifier*	Identifier for the time where an event stops or starts
	Position On Lifeline*	Relative position of event on lifeline (e.g., top-most or first, second, etc.)
	Formula	Algebraic formula for calculating the time of event occurrence (i.e., starting or stopping of the event) relative to beginning of node lifeline
Referenced Types		
Operational Node*		See OV-2 Definition Table.
Information Exchange		See OV-3 Definition Table.
Capability		See OV-5 Definition Table.
Action*		See OV-6b Definition Table.
Event*		See OV-6b Definition Table.
Guard		See OV-6b Definition Table.
Relationships		
Diagram Represents Operational Thread		
	Diagram Identifier	Identifier of the diagram that represents the operational thread
	Operational Thread Name	Name of the operational thread or scenario
Operational Thread Defines Capability		
	Operational Thread Name	Name of the operational thread or scenario
	Capability Name	Name of the capability that is enabled or defined by the operational thread

Data Element	Attributes	Explanation
Node Lifeline (or Swim Lane) is Associated with an Operational Node*		
	Node Lifeline Name*	Name of the node lifeline
	Operational Node Name*	Name of the operational node associated with the node lifeline
Event Starts at Time*		
	Event Name*	Label of the event on the graphic
	Starting Event Time Identifier*	Identifier of the time at which the event occurs or starts; gives the relative position of the event on its starting lifeline; may be identical to ending time
Event Ends at Time*		
	Event Name*	Label of the event on the graphic
	Ending Event Time Identifier*	Identifier of the time at which the event ends; gives the relative position of the event on its ending lifeline; value of time should be greater than or equal to the value of the starting time, in terms of lifeline position
Event Originates from Lifeline*		
	Event Name*	Label of the event on the graphic
	Originating Lifeline Name*	Name of the lifeline from which the event originates
Event Terminates at Lifeline*		
	Event Name*	Label of the event on the graphic
	Terminating Lifeline Name*	Name of the lifeline at which the event terminates
Event is Associated with Action*		
	Event Name*	Label of the event
	Action Name*	Name/identifier of an action
Action is Related to Operational Activity*		
	Action Name*	Name/identifier of an action

Data Element	Attributes	Explanation
	Operational Activity Name*	Name/identifier of an activity from OV-5
	Relationship Description*	Text description of the relationship (e.g., action is same as activity, action is contained in activity [activity includes sequence of actions], action contains activity)
Event maps to Input, Output, and/or Control of Operational Activity*		
	Event Name	Name of event
	Input, Output, or Control from an Operational Activity	Name of input, output, or control
Event is Associated with Information Exchange*		
	Event Name*	Label of the event
	Information Exchange Identifier*	Identifier of an information exchange associated with the event
Link Originates from Operational Activity		
	Link Name	Label of the link on the graphic
	Originating Operational Activity	Name of the operational activity from which the link originates
Link Terminates at Operational Activity		
	Link Name	Label of the link on the graphic
	Terminating Operational Activity Name	Name of the operational activity at which the link terminates

5.10.6 AFDM Support

GJXDM is being proposed for the AFDM. See Section 3.7 for information.

5.11 Logical Data Model (OV-7)

5.11.1 Product Definition

The Logical Data Model (OV-7) describes the structure of an architecture domain's system data types and the structural business process rules, defined in the architecture's OV, that govern the system data. It provides a definition of architecture domain data types, their attributes or characteristics, and their interrelationships.

5.11.2 Product Purpose

OV-7, including the domain's system data types or entity definitions, is a key element in supporting interoperability between architectures, since these definitions may be used by other organizations to determine system data compatibility. Often, different organizations may use the same entity name to mean very different kinds of system data with different internal structures. This situation will pose significant interoperability risks, as the system data models may appear to be compatible, each having a First Responder Asset Track data entity — but having different and incompatible interpretations of what First Responder Asset Track means.

An OV-7 may be necessary for interoperability when shared system 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, at a lower level, among systems.

5.11.3 Product Detailed Description

OV-7 defines the architecture domain's system data types or entities and the relationships among the system data types. OV-7 defines each kind of system data type associated with the architecture domain, mission, or business as its own entity, with its associated attributes and relationships. These entity definitions correlate to OV-3 information elements and OV-5 inputs, outputs, and controls.

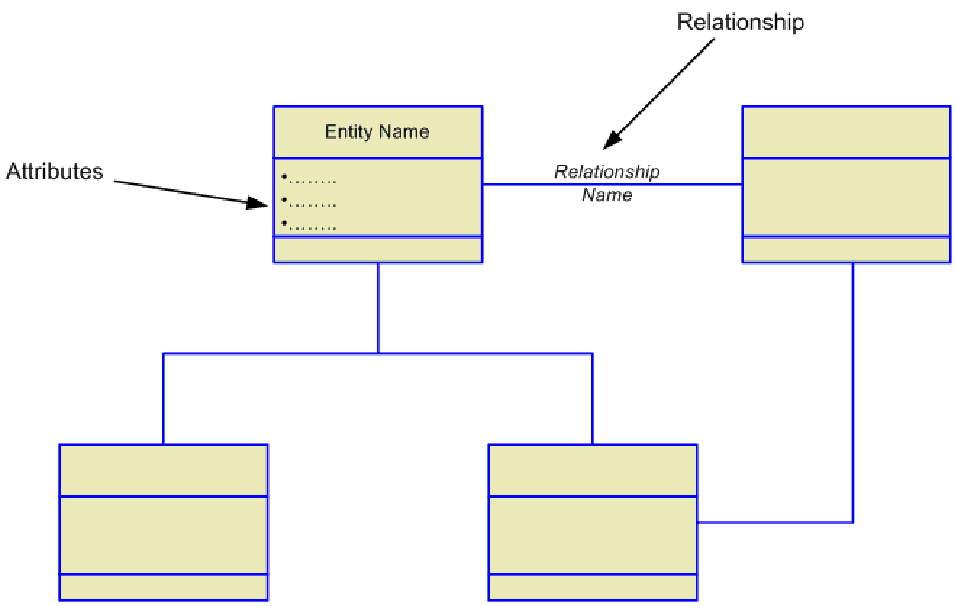
Although they are both called data models, OV-7 should not be confused with the AFDM. OV-7 is an architecture product and describes information about a specific architecture domain. The AFDM is not an architecture product. The AFDM is a database design for a repository of PSAF products and architecture data. AFDM-based repositories can store architecture products, including logical data models, from any PSAF-based architecture project.

The purpose of a given architecture helps to determine the level of detail needed in this product. A formal data model (e.g., the Integrated Definition for Data Modeling (IDEF1X)) (FIPS 184, 1993) that is detailed down to the level of architecture domain system data, their attributes, and their relationships is required for some purposes. An example occurs when validation of completeness and consistency is required for shared data resources. However, for other purposes, a higher-level conceptual data model of the domain of interest will suffice, such as a subject area model or an entity-relation model without attributes. The term *logical data model* is used here in this context, regardless of the level of detail the model exhibits.

The architecture data elements for OV-7 include descriptions of entity, attribute, and relationship types. You can associate attributes with entities and with relationships, depending on the purpose of the architecture.

Figure 24 provides a template for OV-7 with attributes. The format is intentionally generic to avoid implying a specific methodology.

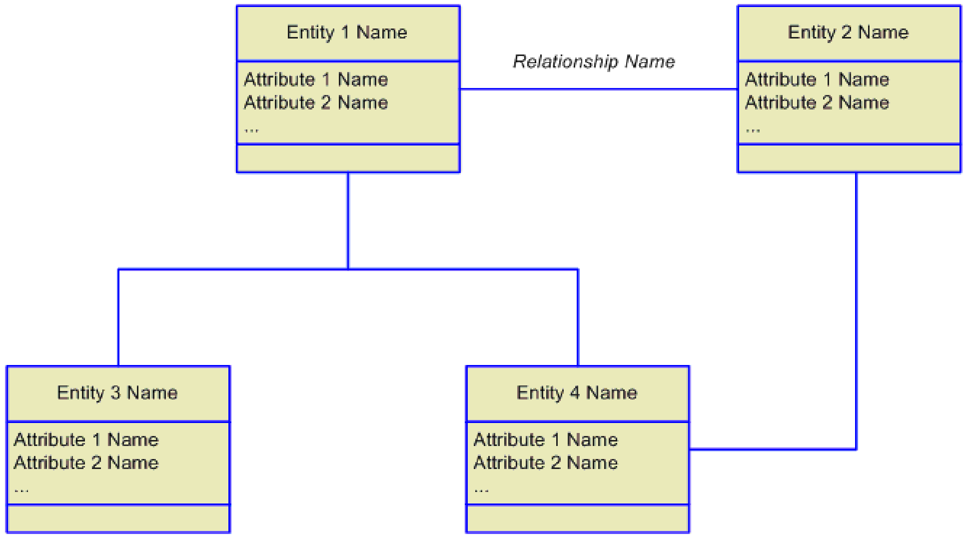
Figure 24: OV-7 – Template



5.11.4 UML Representation

You can model OV-7 in UML using a class diagram (see Figure 25). Class diagrams offer all the UML elements needed to produce entity-relationship diagrams. Class diagrams consist of classes, interfaces, and collaborations, as well as dependency, generalization, association, and realization relationships. The attributes of these classes can be expanded to include associations and cardinality (Booch, 1999). Classes that appear in an OV-7 class diagram correlate to OV-3 information elements and OV-5 inputs, outputs, and controls. The OV-7 class diagram is a separate diagram from the class diagrams that may be developed for other products (e.g., OV-4).

Figure 25: UML Class Diagram for OV-7 – Template



5.11.5 Data Element Definitions

Table 11 describes the architecture data elements for OV-7.

Table 11: Data Element Definitions for OV-7

Data Element	Attributes	Explanation
Graphical Box Types		
Entity Type*		
	Name*	Name of the type of person, place, thing, or message of interest
	Description*	Textual description of the entity type
	Reference	Reference to accepted definition of entity, if one exists
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

Data Element	Attributes	Explanation
	Target Entity Type Name*	Name of the entity type at the target of the relationship
	Cardinality Designation*	Examples: one-to-one, one-to-many
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 (should be an attribute associated with the entity)
	Number of Discriminant Values	Number of different subtypes (if known)
Non-graphical Types		
Attribute Type		
	Name	Name of attribute type
	Definition	Definition of attribute
	Reference	Reference to accepted definition of attribute, if one exists
Data Model Rule		
	Name	Name/identifier of rule
	Type	Examples: Null rule, child delete rule, child update 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
Referenced Types		

Data Element	Attributes	Explanation
Rule		See OV-6a Definition Table.
Standard		See TV-1 Definition Table.
Relationships		
Entity Type is Described by Attribute Type		
	Entity Type Name	Name of entity type
	Attribute Type Name	Name of an associated attribute type
	Role of Attribute	For example: key, foreign key, non-key
Relationship Type is Described by Attribute Type		
	Relationship Type Name	Name of a relationship type
	Attribute Type Name	Name of an associated attribute type
Data Domain Constrains Value 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 Name	Name/identifier of a rule associated with that relationship type See OV-6a Data Element Definition Table.
Entity Type has Rule		
	Entity Type Name	Name of an entity type
	Rule Name	Name/identifier of a rule associated with that relationship type. See OV-6a Data Element Definition Table.
Attribute Type has Rule		
	Attribute Type Name	Name of an attribute type
	Rule Name	Name/identifier of a rule associated with an attribute

Data Element	Attributes	Explanation
Data Model Rule is Associated with a Structural Assertion Type Rule		
	Data Model Rule Name	Name of a data model rule
	Rule Name	Name/identifier of a structural assertion type rule associated with a data model rule
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
Entity is Related to Information Element*		
	Entity Type Name*	Name of an entity type
	Information Element Identifier*	Identifier of an information element
	Description of Relationship*	Text description of the relationship between the entity type and the information element (e.g., same, is subset of, contains)
Entity is Related to Operational Activity Input/Output Information*		
	Entity Type Name*	Name of an entity type
	Operational Activity I/O Information Name*	Name of the information associated with an activity input or output
	Description of Relationship*	Text description of the relationship between the entity and activity input/output (e.g., same, is subset of, contains)
Data Model Conforms to Data Modeling Standard		
	Data Model Name	Name of data model
	Data Model Standard Name	Name of data modeling standard from TV-1

5.11.6 AFDM Support

GJXDM is being proposed for the AFDM. See Section 3.7 for information.

6 Systems View Products

The Systems View (SV) is a set of graphical and textual products that describe systems and interconnections providing for, or supporting, public safety functions. SV products focus on specific physical systems with specific physical or geographical locations. The relationship between architecture data elements across the SV to the OV can be illustrated by example as systems are procured and fielded to support organizations and their operations.

The SV products are:

- Systems Interface Description (SV-1)
- Systems Communications Description (SV-2)
- Systems-Systems Matrix (SV-3)
- Systems Functionality Description (SV-4)
- Operational Activity to Systems Functionality Traceability Matrix (SV-5)
- Systems Data Exchange Matrix (SV-6)
- Systems Performance Parameters Matrix (SV-7)
- Systems Evolution Description (SV-8)
- Systems Technology Forecasts (SV-9)
- Systems Functionality Sequence and Threads (SV-10)
- Systems Functionality and Timing Descriptions
  - Systems Rules Model (SV-10a)
  - Systems State Transition Description (SV-10b)
  - Systems Event-Trace Description (SV-10c)
- Physical Schema (SV-11)

6.1 Systems Interface Description (SV-1)

6.1.1 Product Definition

The Systems Interface Description (SV-1) depicts systems nodes and the systems resident at these nodes. Its aim is to support organizations and human roles represented by operational nodes of the OV-2. SV-1 also identifies the interfaces between systems and systems nodes.

6.1.2 Product Purpose

SV-1 identifies systems nodes and systems that support operational nodes. You can also identify interfaces that cross organizational boundaries or key interfaces in this product. Some systems can have numerous interfaces. Initial versions of this product might only show key interfaces. You might develop detailed versions, as necessary, for use in system acquisition, as part of requirements specifications, and for determining system interoperability at a finer level of technical detail.

6.1.3 Product Detailed Description

SV-1 links together the OV and SV by depicting the assignments of systems and systems nodes, and their associated interfaces, to the operational nodes, and their associated needlines, described in OV-2. OV-2 depicts the operational nodes representing organizations, organization types, or human roles. SV-1 depicts the systems nodes that house operational nodes — for example platforms, units, facilities, and locations — and the corresponding systems resident at these systems nodes and which support the operational nodes.

The term *system* in the PSAF is used to denote a FoS, SoS, nomenclatured system, or a subsystem. An item denotes a hardware or software item. Only systems, subsystems, or hardware/software items and their associated standards are documented in this product, where applicable. Details of the communications infrastructure — for example , physical links, communications networks, routers, switches, communications systems, satellites — are documented in SV-2.

In addition to depicting systems nodes and systems, SV-1 addresses system interfaces. An interface, as depicted in SV-1, is a simplified, abstract representation of one or more communications paths between systems nodes or between systems, including communications systems. It is usually depicted graphically as a straight line. SV-1 depicts all interfaces that are of interest for the architecture purpose.

An SV-1 interface is the systems representation of an OV-2 needline. A single needline shown in the OV may translate into multiple system interfaces. The actual implementation of an interface may take more than one form, for example, multiple physical links. Details of the physical links and communications networks that put the interfaces into effect are documented in SV-2. Characteristics of the interface are described in SV-3. System functions and system data flows are documented in SV-4, and the system data carried by an interface is documented in SV-6.

You can annotate an interface between systems nodes or systems as a key interface (KI). A KI is an interface where one or more of the following criteria are met:

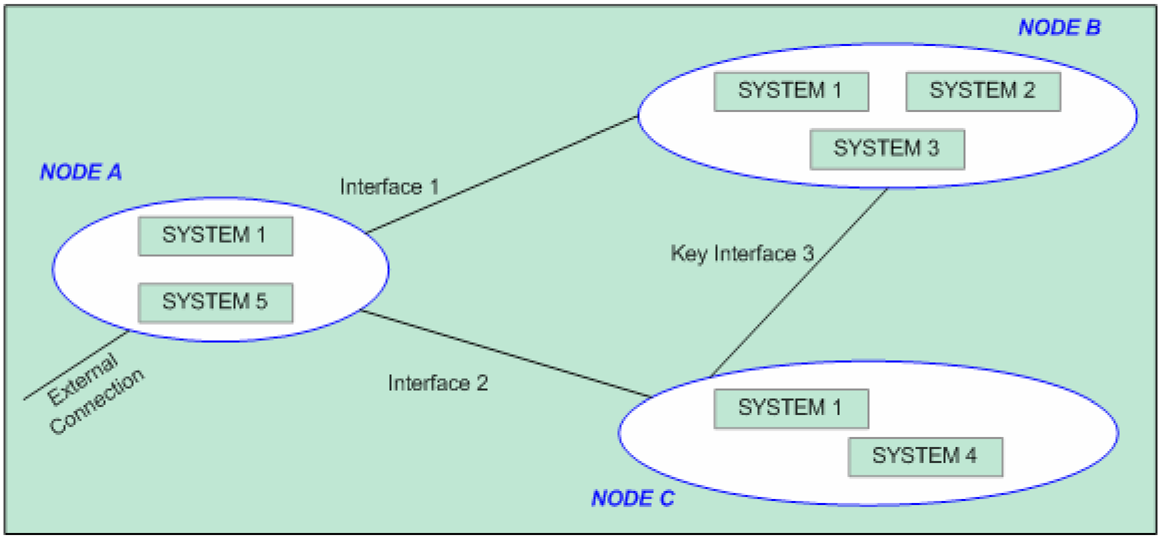
- The interface spans organizational boundaries. (It may be across instances of the same stem, but used by different organizations.)
- The interface is mission-critical.
- The interface is difficult or complex to manage.
- Capability, interoperability, or efficiency issues are associated with the interface.

If you choose, you can add to SV-1 annotations to summarize the system data exchanges carried by an interface.

You can develop several versions of SV-1 to highlight different perspectives of the system interfaces. For example, an internodal version of the product describes systems nodes and the interfaces between them or the systems resident at the systems nodes. An intrasystem version describes subsystems of a single system and the interfaces among them. You can develop other versions, depending on the purposes of the particular architecture.

Figure 26 provides a template of the internodal version, showing system interfaces between nodes from node edge to node edge. The pertinent systems within each node are also shown, but not their specific system-system interfaces. In Figure 26, System 1 is depicted in all three nodes (Node A, Node B, and Node C), which means that the same hardware/software configuration is resident at all three systems nodes. (In this case the notion of same is relative to the purposes of the architecture.) All interfaces are assumed to be two-way unless otherwise noted. The naming convention used for interfaces in Figure 26 is purely for exposition purposes.

Figure 26: SV-1 Internodal Template Showing Systems



You can also show additional information on the graphics. For example, you can add information, such as a system's hardware/software items or the system's functions, as annotations to the system's graphical icon.

Figure 27 provides a notional example in which the system functions have been added for all the systems. This type of SV-1 might be useful in a "to be" architecture, where major system functions have been allocated to systems but other details have not yet been decided.

Figure 27: SV-1 Internodal Version – Node Edge to Node Edge Showing System Functions

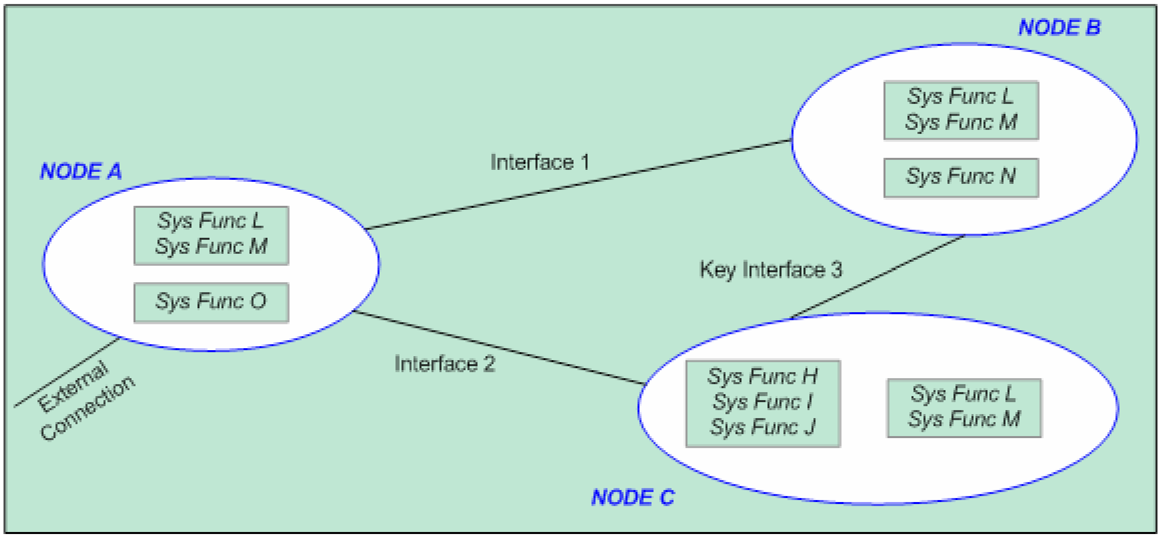
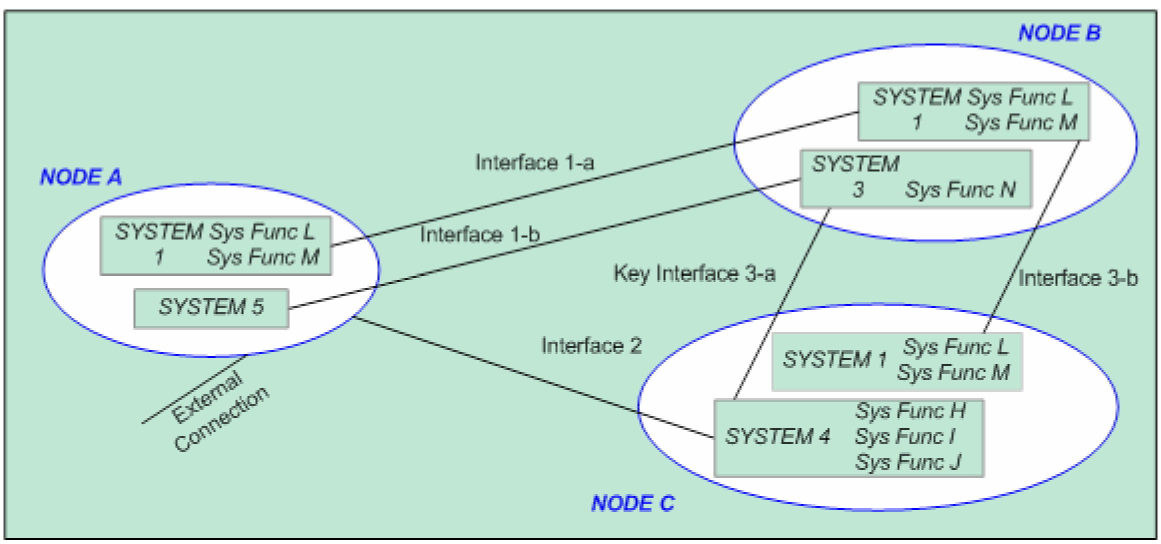


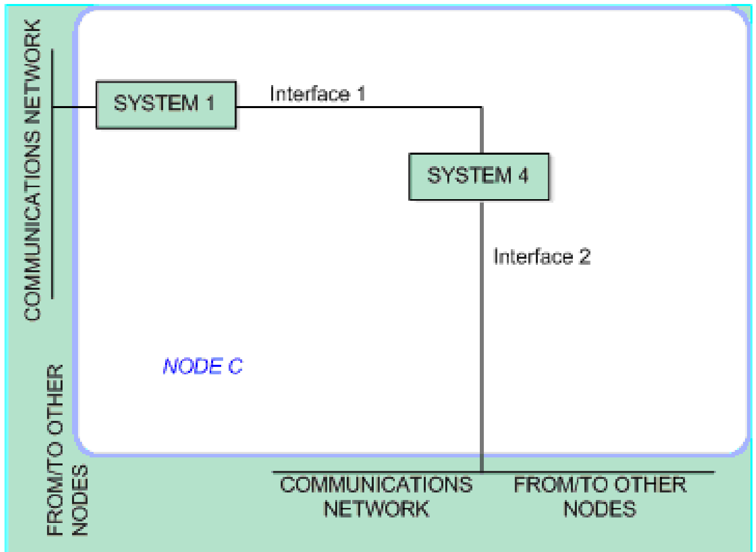
Figure 28 provides a template of the internodal version of SV-1 that extends the node edge connections to specific systems. In Figure 28, the line between System 1 at Node A and System 1 at Node B indicates there is an interface between these two instances of System 1. Similarly, the line between System 5 at Node A and System 3 at Node B indicates that there is a second, logically distinct interface between Node A and Node B, connecting System 3 and System 5.

Figure 28: SV-1 Internodal Version Showing System-System Interfaces – Template



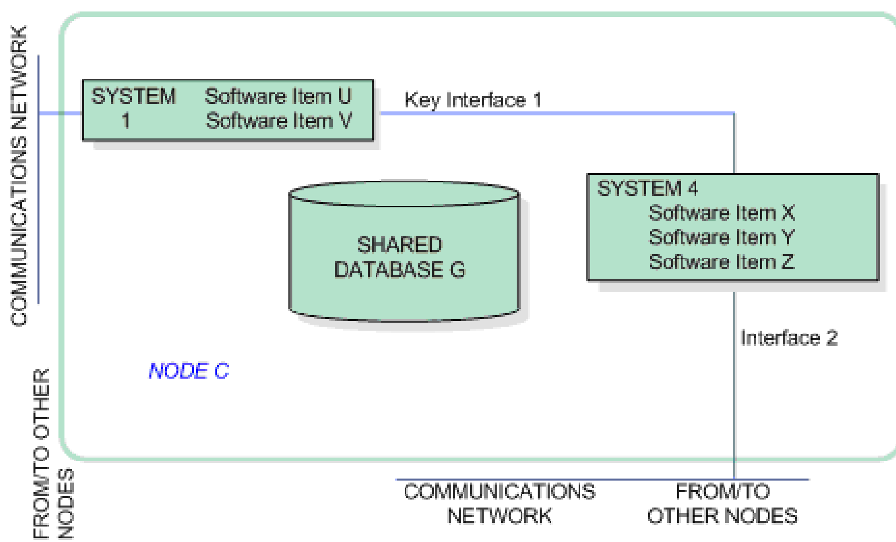
An intranodal version of SV-1 may also be useful to focus analysis on the interfaces between systems within each node and to examine the systems-systems connections within a systems node. This version also continues to show the external connections going out to other systems nodes. Figure 29 provides a template of the intranodal version of SV-1. Note that the naming convention for the interfaces in Figure 29 is purely for exposition purposes. A naming convention for interfaces must assign unique identifiers for interfaces at the various levels of detail for SV-1.

Figure 29: Intranodal Version – Template



You can use the intrasystem version conjunction with the internodal or intranodal versions to analyze and improve the configuration of systems. For example, a purpose of an architecture could be to determine more efficient distribution of software applications. Figure 30 provides a template of the intrasystem version of SV-1 and a notional example that includes a KI and a shared database as a subsystem.

Figure 30: SV-1 Intrasystem Version – Example



6.1.4 UML Representation

SV-1 may be developed in the UML using deployment diagrams to represent SV-1 systems nodes. A deployment diagram is a collection of node symbols connected by lines showing communication associations.

You can use UML to model the SV-1 internodal perspective that show systems by using deployment diagrams that also show UML components. You can accomplish this by defining new stereotypes for UML components.

A stereotype is new type of modeling element that extends the semantics of the metamodel. Stereotypes must be based on certain existing types or classes in the metamodel. Stereotypes may extend the semantics, but not the structure of pre-existing types and classes. Certain stereotypes are predefined in the UML, others may be user-defined. Stereotypes are one of three extensibility mechanisms in UML. Adding new stereotypes is neither necessary nor mandatory, as long as there is consistency in representing the same kind of nodes in each diagram. For example, in any one diagram, all UML nodes are systems nodes, while all UML components are systems that reside on the systems nodes. You can associate components with notes and constraints, the former to specify allocations of resources, such as system functions, to the UML components.

Figure 31 is a template for such a diagram. SV-1 systems nodes trace to operational nodes identified in OV-2. SV-1 interfaces correlate to OV-2 needlines, or use case links.

Figure 31: UML Diagram for Internodal SV-1 Systems – Template

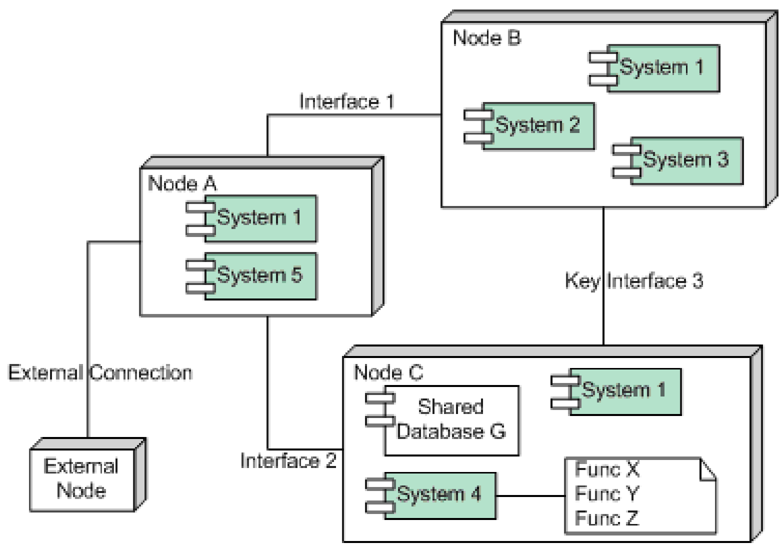
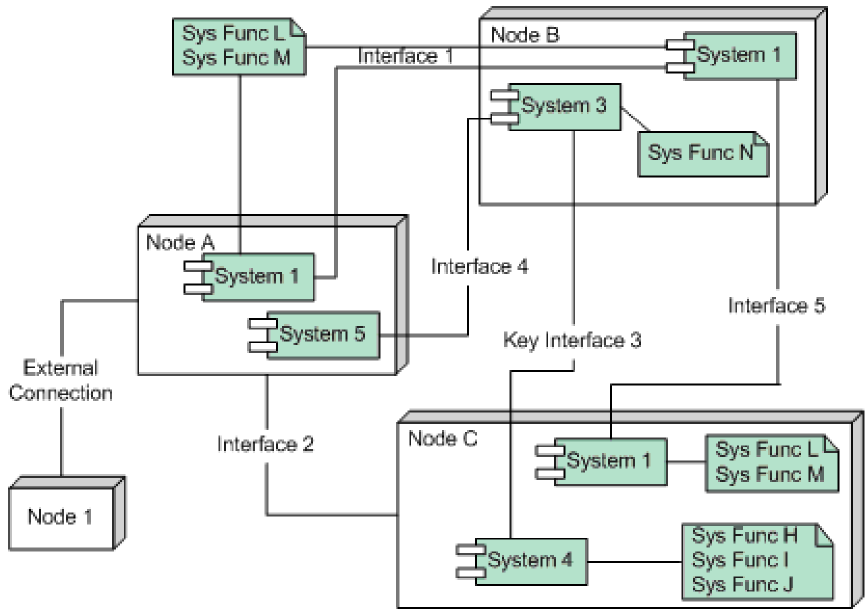


Figure 32 is a template for the SV-1 internodal perspective showing system-to-system interfaces. SV-1 UML components correlate to the systems. In UML, you can assign classes or use cases (SV-4 system functions) to UML components to associate system functions with systems in SV-1.

Figure 32: UML Diagram for Internodal SV-1 System-System Interfaces – Template



You can model the SV-1 intranodal perspective in UML using component diagrams with UML components representing systems and notes to specify to the UML components allocations of resources, such as system functions and hardware/software items.

6.1.5 Data Element Definitions

Table 12 describes the architecture data elements associated with SV-1.

Table 12: Data Element Definitions for SV-1

Data Element	Attributes	Explanation
Graphical Box Types		
Systems Node*		(Appears in internodal and intranodal versions)
	Name*	Name of a systems node, usually denoting a facility where an operational node is located (e.g., platform, unit, facility, location)
	Description*	Text summary description of systems node's role or mission and associated resources (e.g., platforms, units, facilities, and locations) that house the systems that support these roles or missions

Data Element	Attributes	Explanation
System*		Representation of a system (FoS, SoS, subsystem) name/identifier of system or subsystem. (Appears in all versions.)
	Name*	Name/identifier of system or subsystem
	Description*	Text summary of function or set of functions performed and constituent parts contained, for example, shared DB
Graphical Arrow Types		
Interface*		Required in internodal and intranodal versions
	Name*	Name/identifier of interface
	Description*	Text description of interface; may include a discussion of communications systems or communications system elements involved, as well as indications as to whether interface is two-way or one-way only
	Endpoint 1: Name*	Name of entity (systems node, system) that is at one end of the interface. In case of one-way connections, this endpoint is the source endpoint.
	Endpoint 2: Name*	Name of entity (systems node, system) that is at the other end of the interface. In case of one-way connections, this endpoint is the target endpoint.
	Key Interface (KI) Designation	If applicable
	Interface Control Document (ICD) Name	If applicable: Name/reference to the ICD that specified the KI (when developed)
	Key Interface Rationale	Text describing why this is a KI if applicable
Non-Graphical Types		
Hardware/Software Item		Software or hardware items of a system (or subsystem). (Appears in intrasystem version and appears optionally on other versions). Represents a software application or hardware equipment that has a serial number (out of the box, not running as part of a system).
	Name	Name/identifier of system hardware/software item, including model/version number
	Type	Type of hardware/software item (for example, hardware item), platform item (combined

Data Element	Attributes	Explanation
		hardware and system software), system software, or application (mission unique) software
	Description	Text description of functions supported by system hardware/software item
	Vendor/Source	Source of system hardware/software item
System Function (if used)		
	Name	Name/identifier of system function
	Description	Text summary description of system function
Referenced Types		
Operational Node*		See OV-2 Definition Table.
Needline*		See OV-2 Definition Table.
Standard*		See TV-1 Definition Table.
Relationships		
Systems Node Contains System*		
	Systems Node Name*	Name/identifier of a systems node
	Systems Name*	Name/identifier of a system resident at that node
System Contains (Sub-) Systems*		
	System Name*	Name/identifier of a systems node
	(Sub-) Systems Name*	Name/identifier of a system or subsystem contained within system
System Contains Hardware/Software Item		
	System Name	Name/identifier of a system
	Hardware/Software Item Name	Name/identifier of a hardware/software item of the system
System Performs System Function		
	System Name	Name/identifier of a system
	System Function Name	Name/identifier of a system function

Data Element	Attributes	Explanation
		performed by system
Systems Node Supports Operational Node*		
	Systems Node Name*	Name/identifier of systems node
	Operational Node Name*	Name/identifier of operational node supported by automation resident at the systems node
System Supports Operational Node		
	System Name	Name/identifier of system
	Operational Node Name	Name/identifier of an operational node with activities supported by the system
Interface Depicts Automated Portion of Needline*		An interface in SV-1 graphically represents a need to implement an interface between two systems (contained within systems nodes). Implementation detail is depicted in SV-2
	Interface Name*	Name/identifier of interface
	Needline Name*	Name/identifier of needline partially or fully supported by the interface
System Uses Standard*		
	System Name*	Name/identifier of system
	Standard Name*	Name/ID number of a standard used by the system
System Hardware/Software Item Uses Standard		
	System Hardware/Software Item Name	Name/identifier of system hardware/software item
	Standard Name	Name/ID number of a standard used by the system hardware/software item

6.1.6 AFDM Support

GJXDM is being proposed for the AFDM. See Section 3.7 for information.

6.2 Systems Communications Description (SV-2)

6.2.1 Product Definition

The Systems Communications Description (SV-2) depicts pertinent information about communications systems, communications links, and communications networks. SV-2 documents the kinds of communications media that support the systems and put their interfaces into effect, as described in SV-1. Thus, SV-2 shows the communications details of SV-1 interfaces that automate aspects of the needlines represented in OV-2.

6.2.2 Product Purpose

You can use SV-2 to document how interfaces (described in SV-1) are supported by physical media. This kind of communications media support information is critical in performing certain infrastructure and system acquisition decisions.

6.2.3 Product Detailed Description

SV-2 documents the specific communications links or communications networks and the details of their configurations through which systems interface. While SV-1 depicts interfaces between systems or systems nodes, SV-2 contains a detailed description of how each SV-1 interface is implemented (e.g., how each interface is implemented including networking systems, multiple communications links, communications networks, routers, and gateways).

Networking systems include switches, routers, and communications satellites. They are systems whose primary function is to control the transfer and movement of system data as opposed to performing mission application processing.

A communications link is a single physical connection from one system or node to another. A communications path is a connected sequence of communications systems and communications links originating from one system or node and terminating at another system or node.

A communications network may contain multiple paths between the same pair of systems. The term *interface* as used in SV-1, and referenced in SV-2, represents an abstraction of one or more communications paths.

The graphical presentation and supporting text for SV-2 should describe all pertinent communications attributes (e.g., waveform, bandwidth, radio frequency, and packet or waveform encryption methods). Communications standards are also documented in this product, where applicable.

SV-2 depicts the implementation details for the interfaces described in SV-1 by decomposing them into communications systems, communications links, and communications networks. Thus it can present either internodal or intranodal versions. The internodal version details the communications links or communications networks that interconnect systems nodes, node edge to node edge, or specific systems (system-to-system from one node to other nodes, and vice versa).

The intranodal version of SV-2 looks inside each of the represented nodes to illustrate the communications links between specific systems.

Figure 33 provides a template for the internodal version of SV-2. Note that Figure 33 translates the single-line representations of interfaces (as shown in the SV-1 internodal version) into implementation details of the communications infrastructure that provides the connections. The small boxes in Figure 33 represent communications systems, as do the satellite icons. The lines between the communications systems represent communications links and, together with communications systems, can be organized into communications paths or networks.

Figure 33: SV-2 Internodal Version – Template

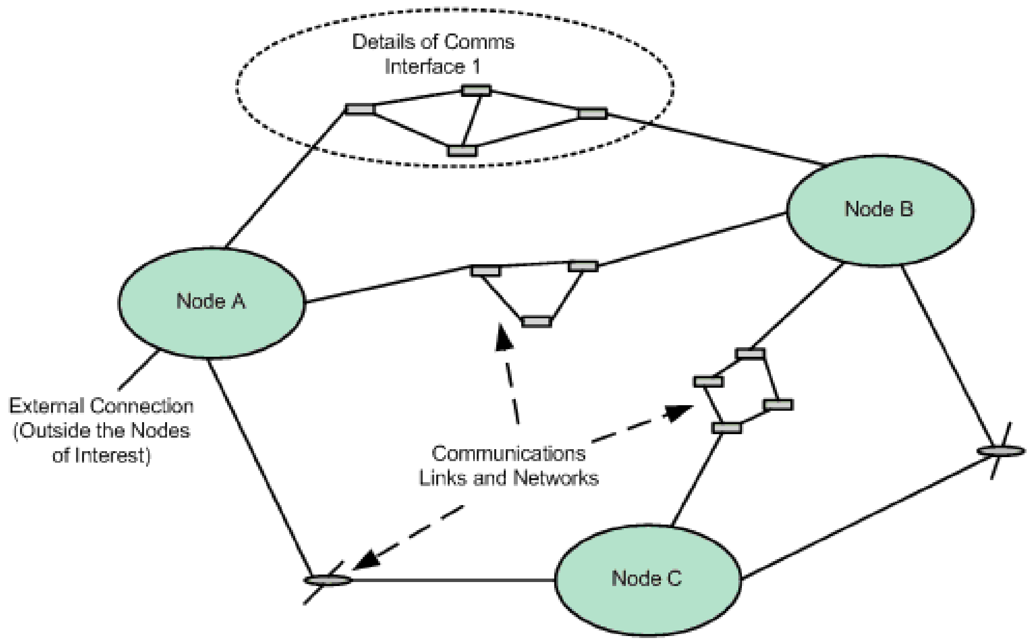
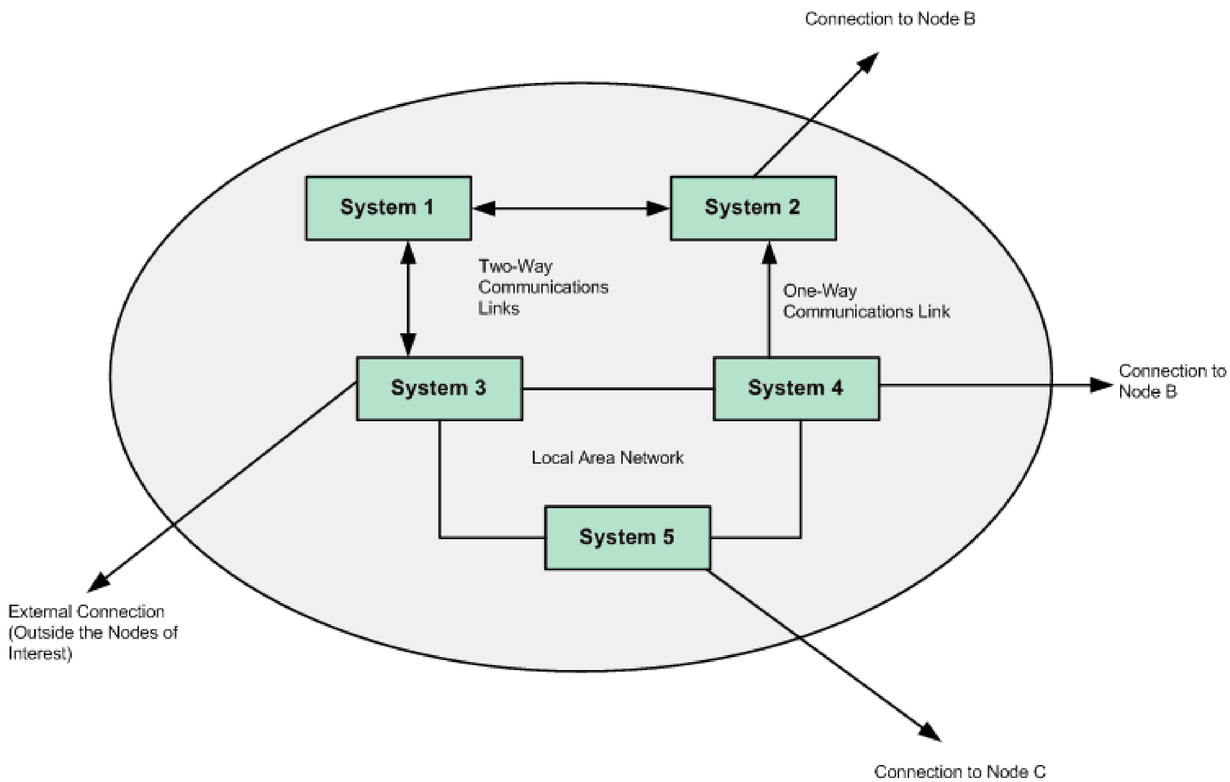


Figure 34 provides a template for the intranodal version of SV-2.

Figure 34: SV-2 Intranodal Version – Template



6.2.4 UML Representation

There is no equivalent to this product in a UML diagram type. If the scope of the architecture is the communications infrastructure, then network and communications diagrams are best described using conventional network diagrams. However, for model consistency across all diagrams, you can describe communications detail in UML using deployment diagrams to represent SV-2 systems nodes. A deployment diagram is a collection of node symbols connected by lines showing communication associations.

You can model SV-2 intranodal perspective in UML by using component diagrams with UML components. The UML components will represent communications systems, and lines between components will represent communications links.

6.2.5 Data Element Definitions

SV-2 depicts the implementation of the interfaces from SV-1 as specific communications systems, communications links, communications paths, or communications networks and the details of their configurations through which the system interface. Table 13 describes the architecture data elements associated with SV-2.

Table 13: Data Element Definitions for SV-2

Data Element	Attributes	Explanation
Graphical Box Types		
Systems Node*		See SV-1 Definition Table.
System*		See SV-1 Definition Table.
Communications System*		Type of system
	Name*	Name/identifier of communications system whose primary function is to control the transfer and movement of system data Examples include communications networks, switches, routers, and communications satellites
	Description*	Text summary description of communications functions of the communications system
	Communications System Standard, Protocols Supported*	For example, TCP/IP, Link-11 See TV-1 or TV-2.
Graphical Arrow Types		
Communications Link*		A communications link connects systems nodes or systems (including communications systems).
	Name*	Name/identifier of a communications link that describes a single physical connection to communicate from one systems node/system to another
	Communication Standard, Protocols Supported*	For example, TCP/IP, Link-11. See TV-1 or TV-2.
	Capacity*	Throughput or bandwidth; channel capacity
	Infrastructure Technology*	Infrastructure technology supporting this communications link. The level of detail contained here will depend on the purposes of the architecture (e.g., fiber, twisted pair, wireless, ATM, Internet, Intranet, Ethernet, VPN, radio plus frequency, encryption (if any)
	Endpoint 1: Systems Node/System Name*	Name of graphic box that is at one end of the communications link on the diagram; in case of one-way connections, this endpoint is the source endpoint. The endpoint of a communications link may also be listed as external. if the endpoint is outside the scope of the architecture or diagram.

Data Element	Attributes	Explanation
	Endpoint 2: Systems Node/System Name*	Name of the graphic box that is at the other end of the communications link on the diagram; in case of one-way connections, this endpoint is the destination endpoint. The endpoint of a communications link may also be listed as external, if the endpoint is outside the scope of the architecture or diagram.
Communications Path*		A series of communications links that support an interface (from SV-1).
	Name*	Name/identifier of a multiple communications path that describes a single way (with no options) to communicate from one systems node/system to another
	Description*	Textual description of communications 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 communications 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 communications path; if path is one-way, this endpoint should be the destination endpoint. May be listed as external.
	Number of Communications Links	Number of communications links or steps in the communications path
LAN		
	Name	Name/identifier of local area network
	Description	Textual description of LAN, including purpose, size, and capability
Communications Network		A collection of communications links, and systems nodes and systems, where applicable
	Name	Name/identifier for a Wide Area Network or Metropolitan Area Network
	Description	Textual description of communications network purpose, size, and capability
	Security Classification	Classification of system data that the communications network is allowed to carry
Referenced Types		

Data Element	Attributes	Explanation
Interface		See SV-1 Definition Table.
Hardware/Software Item		See SV-1 Definition Table.
Standard*		See TV-1 Definition Table; includes standards for OSI layers.
Relationships		
LAN Contains Communications Link		
	LAN Name	Name/identifier of a LAN
	Communications Link Name	Name/identifier of a communications 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 Communications Links*		
	Communications Path Name*	Name/identifier of communications path
	Communications Link*	Name/identifier of the communications link within the communications path
	Communications Link Position in Communications Path*	Position of the communications link in the communications path, given in terms of number of communications link from endpoint 1
Communications Path Implements Interface*		
	Communications Path Name*	Name/identifier of communications path
	Interface Name*	Name/identifier of an interface from SV-1 that is implemented by the communications path
Communications Network Contains LAN		
	Communications Network Name	Name/identifier of a communications network
	LAN Name	Name/identifier of a LAN that is part of the communications network

Data Element	Attributes	Explanation
Communications Network Contains Communications Link		
	Network Name	Name/identifier of a communications network
	Communications Link Name	Name/identifier of a communications link that is part of the communications network
Communications Network Contains Communications System		
	Network Name	Name/identifier of a communications network
	Communications System Name	Name/identifier of a communications system that is part of the communications network
System Is Attached to Communications Network		
	System Name	Name/identifier of a system
	Communications Network Name	Name/identifier of a communications network to which the system is attached
Systems Node is Attached to Communications Network		
	Systems Node Name	Name/identifier of a systems node
	Communications Network Name	Name/identifier of a communications network that is attached to the node (a network to which all systems at the systems node are connected via a common service)
Communications System Contains Hardware/Software Item		
	Communications System Name	Name/identifier of a communications system
	Hardware/Software Item Name	Name/identifier of a hardware/software item that the communications system is composed of
Communications System Uses Standard*		See TV-1 or TV-2.
	Communications System Name*	Name/identifier of communications system
	Standard Name*	Name/ID number of a standard used by the communications system
System Hardware/Software Item		For communications hardware/software item.

Data Element	Attributes	Explanation
Uses Standard*		See SV-1 Definition Table.
	System Hardware/Software Item Name*	Name/identifier of system hardware/software item
	Standard Name*	Name/ID number of a standard used by the system hardware/software item
Communications Link Uses Standard*		See TV-1 or TV-2.
	Communications Link Name*	Name/identifier of communications link
	Standard Name*	Name/ID number of a standard used by the communications link
Communications Network Uses Standard		See TV-1 or TV-2.
	Communications Network Name	Name/identifier of communications network
	Standard Name	Name/ID number of a standard used by the communications network

6.2.6 AFDM Support

GJXDM is being proposed for the AFDM. See Section 3.7 for information.

6.3 Systems-Systems Matrix (SV-3)

6.3.1 Product Definition

The Systems-Systems Matrix (SV-3) provides detail on the interface characteristics described in SV-1 for the architecture, arranged in matrix form.

6.3.2 Product Purpose

SV-3 allows a quick overview of all the interface characteristics presented in multiple SV-1 diagrams. The matrix form can support a rapid assessment of potential commonalities and redundancies or, if fault-tolerance is desired, the lack of redundancies.

You can organize SV-3 in a number of ways — for example, by domain, by operational mission phase — to emphasize the association of groups of system pairs in context with the architecture purpose. SV-3 can be a useful tool for managing the evolution of systems and system infrastructures, the insertion of new technologies or functionalities, and the redistribution of systems and processes in the context of evolving operational requirements.

6.3.3 Product Detailed Description

SV-3 is a summary description of the system-system interfaces identified in SV-1. SV-3 is similar to an N^2-type matrix, where the systems are listed in the rows and columns of the matrix, and each cell indicates a system pair interface, if one exists. You can present many types of interface information in the cells of SV-3, and you can represent the system-system interfaces using a number of different symbols and color codes that depict different interface characteristics. The following are examples:

- Status (e.g., existing, planned, potential, deactivated)
- Purpose (e.g., logistics)
- Classification level (e.g., Type 1 Encryption)
- Means (e.g., P25 radio)
- Standard
- Key Interface

Figure 35 provides a template for SV-3. Each cell may contain several interface characteristics, although Figure 35 does not illustrate this. In an integrated architecture, held in a repository, you can analyze this matrix to gain insight into the precise nature of the interface and even into the data elements traveling between the systems.

Figure 35: SV-3 – Template

	System									
System	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1		■								
2	■		■	■	■	■				
3		■				■				
4		■				■				
5		■				■				
6	■	■	■	■	■		■	■	■	■
7						■				
8						■				
9						■				
10						■				

6.3.4 UML Representation

There is no equivalent to this product in UML. Row and column headings should trace to systems of SV-1 and SV-2.

6.3.5 Data Element Definitions

Table 14 defines the architecture data elements for SV-3.

Table 14: Data Element Definitions for SV-3

Data Element	Attributes	Explanation
Non-Graphical Box Types		
Code Legend Type*		
	Name*	Name of the code legend type. Example: status, purpose, classification level, means
	Description*	Textual description of the code legend type used to categorize interface characteristics
Code Legend*		
	Code*	A unique code representing a certain interface characteristic

Data Element	Attributes	Explanation
	Type Name*	Name of a code legend type this code is categorized under
	Description*	Textual description of the interface characteristic represented by the code
System*		See SV-1 Definition Table.
Interface Entry*		
	Name*	Name/identifier of interface entry. It should be the same as the name of the interface for which this interface entry is detailing the characteristics of the interface
	Description*	Textual summary description of the interface entry
Referenced Types		
Interface*		See SV-1 Definition Table.
Relationships		
Code Legend Represents Characteristics of Interface Entry*		
	Code Legend Code*	Code of a code legend, which implies the code legend type. (Note that the characteristics of an interface entry cannot be represented by more than one code legend with the same code legend type.)
	Interface Entry Name*	Name/identifier of an interface entry for which the characteristics of the interface entry are represented by the code
Code Legend is Categorized by Code Legend Type*		
	Code Legend Name*	Name/identifier of a code legend
	Code Legend Type Name*	Name/identifier of code legend type categorizing the code legend
Interface Entry Details Interface Characteristics*		
	Interface Entry Name*	Name/identifier of interface entry
	Interface Name*	Name/identifier of a interface for which the interface entry is detailing its characteristics

6.3.6 AFDM Support

GJXDM is being proposed for the AFDM. See Section 3.7 for information.

6.4 Systems Functionality Description (SV-4)

6.4.1 Product Definition

The Systems Functionality Description (SV-4) documents system functional hierarchies and system functions, and the system data flows between them. Although there is a correlation between OV-5 or business-process hierarchies and the system functional hierarchy of SV-4, it need not be a one-to-one mapping, mapping. Hence the need for the SV-5, which provides that mapping.

6.4.2 Product Purpose

The primary purposes of SV-4 are to:

- Develop a clear description of the necessary system data flows that are input, or consumed by, and output, or produced by, each system.
- Ensure that the functional connectivity is complete (i.e., that a system's required inputs are all satisfied).
- Ensure that the functional decomposition reaches an appropriate level of detail.

6.4.3 Product Detailed Description

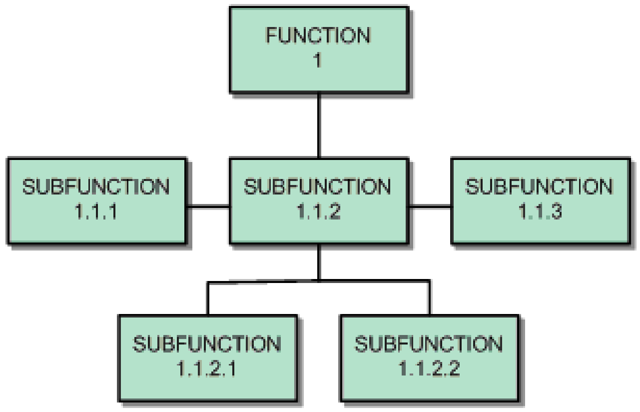
SV-4 describes system functions and the flow of system data among system functions. It is the SV counterpart to OV-5. SV-4 may be represented in a format similar to data flow diagrams (DFDs) (DeMarco, 1979). The scope of this product may be enterprise-wide, without regard to which systems perform which functions, or it may be system-specific. Variations may focus on intranodal system data flow, internodal system data flow, and system data flow without node considerations, function to system allocations, and function to node allocations.

You can identify the system functions you document in the SV-4 using the Service Component Reference Model (SRM),[6] or some other system function taxonomy, and correlate those functions to SV-1 and SV-2 systems. System functions are not limited to internal system functions, and can include Human Computer Interface (HCI) and Graphical User Interface (GUI) functions. They also can include functions that consume or produce system data from or to system functions that belong to external systems. You can use the external system data sources or sinks to represent the human that interacts with the system or external systems. You can use the system data flows between the external system data source or sink (representing the human or system) and the HCI, GUI, or interface function to represent human-system interactions, or system-system interfaces. You also specify in this product standards that apply to system functions, such as HCI and GUI standards.

[6] The SRM is one of the reference models associated with the Federal Enterprise Architecture.

Like OV-5, SV-4 may be hierarchical in nature. It may have both a hierarchy or decomposition model and a system data flow model. The hierarchy model documents a functional decomposition. The functions decomposed are system functions. Figure 36 shows a template for a functional decomposition model of SV-4.

Figure 36: SV-4 – Template (Functional Decomposition)

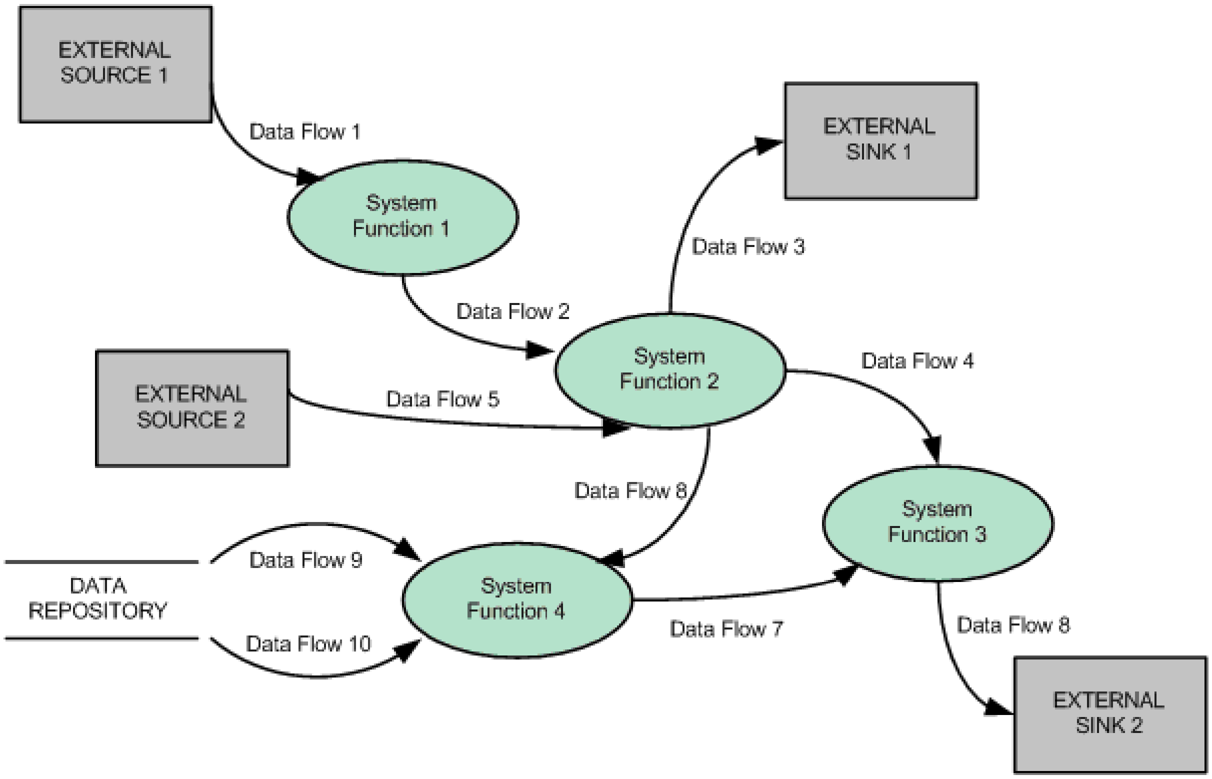


SV-4 documents system functions, the flows of system data between those functions, the internal system data repositories or system data stores, and the external sources and sinks for the system data flows. It may represent external systems or the humans that interact with the system function. External sources and sinks represent sources external to the diagram scope but not external to the architecture scope.

Figure 37 shows a template for the data flow diagram (DFD) of SV-4. The ovals represent system functions at some consistent level of decomposition, the squares are external system data sources and sinks, and the arrows represent system data flows. Parallel lines represent internal system data repositories.

You can arrange SV-4s hierarchically, with each system function at the parent level being decomposed into a child SV-4 at the next level.

Figure 37: SV-4 – Template (Data Flow Diagram)



6.4.4 UML Representation

You can employ use cases, classes, and class operations to represent system functions in SV-4. Use case diagrams model the interactions between humans, external systems, and system functions (use cases) within the scope of the diagram. Systems use cases may be identified as a refinement of the operational use cases. (They have an <<include>> relationship with the operational use cases.) They represent the automated portions of those use cases representing operational activities from OV-5.

Actors in systems use-case diagrams may represent a mix of both the humans supported by the system functions, corresponding to those in OV-5 use case diagrams, and other systems external to the system being described, but which are not necessarily external to the architecture. They are the equivalent of external sources and sinks in DFD notation. In addition, class diagrams can show static relationships between system functions. These diagrams collectively describe the systems support to the operational activities and operational use cases modeled in OV-5. Figure 38 is a sample SV-4 template of a UML use case diagram. Figure 39 shows the corresponding class diagram.

Figure 38: UML Use Case Diagram for SV-4

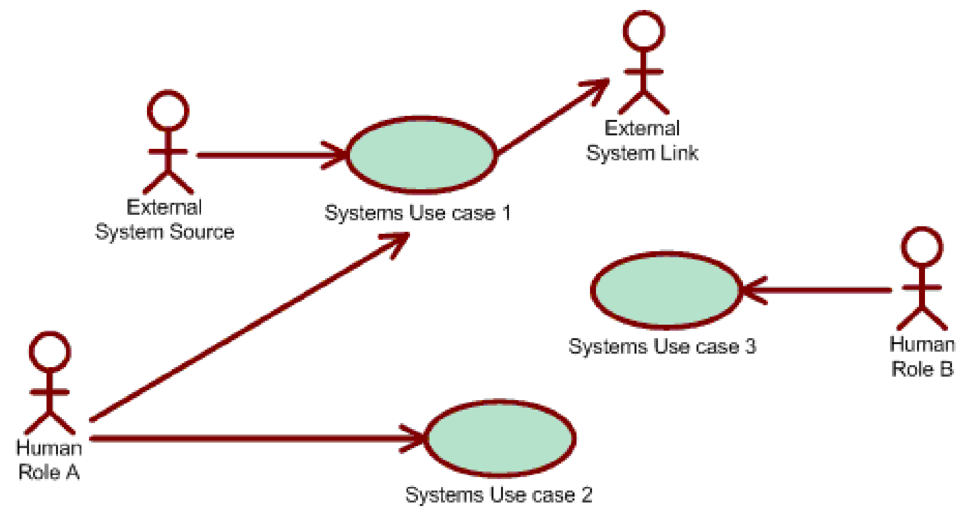
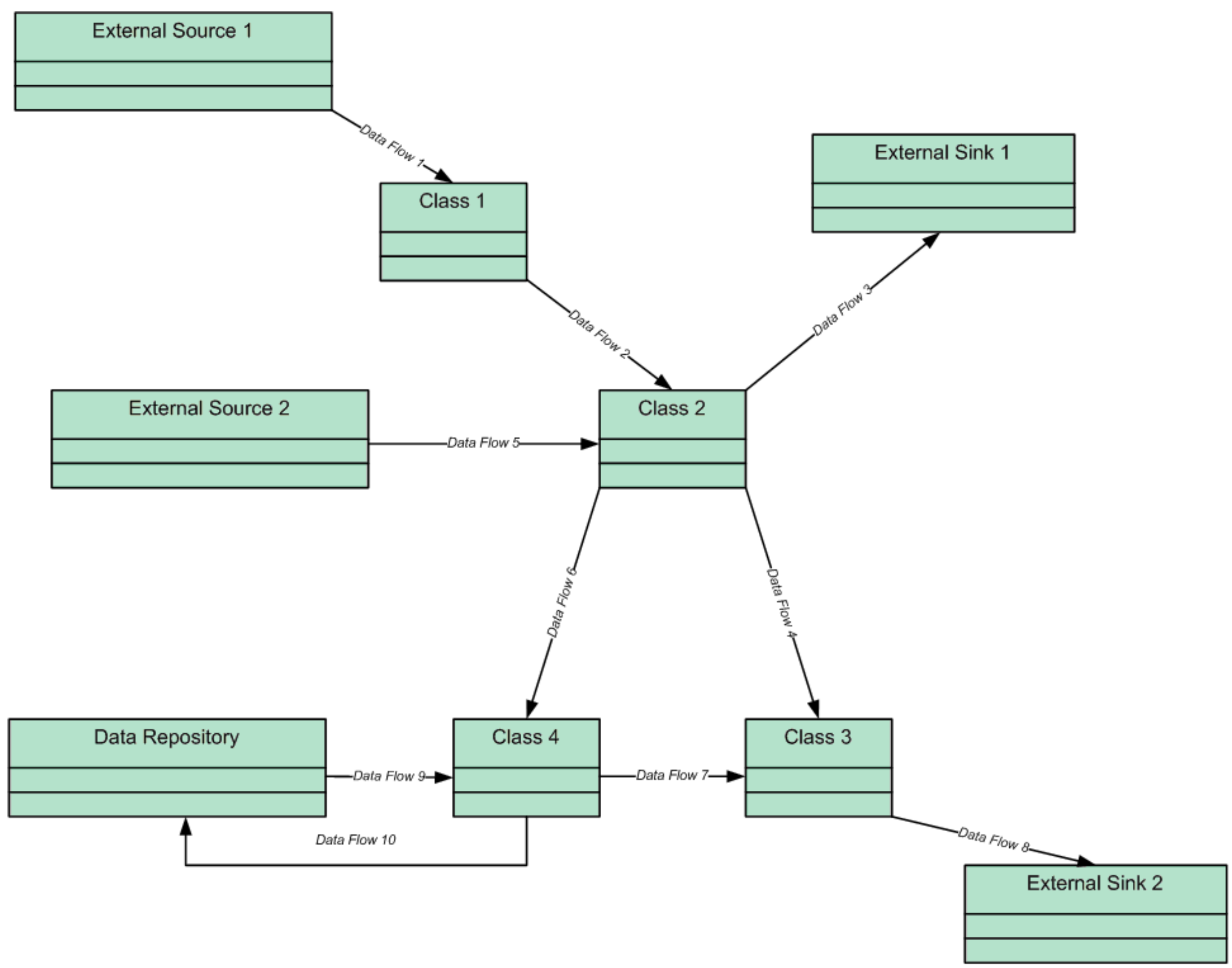


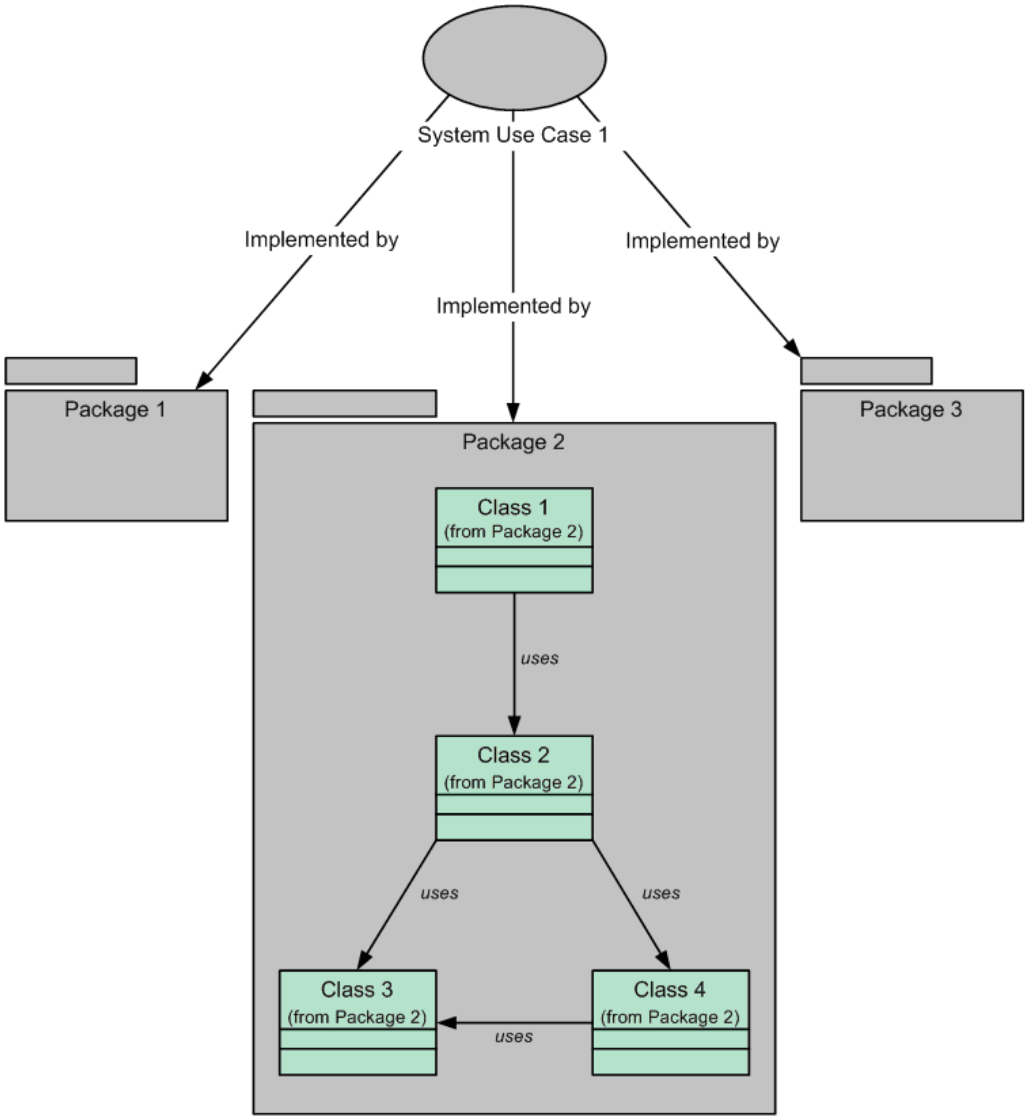
Figure 39: UML Class Diagram Showing System Functions and Relationships



The systems use cases and classes and their relationships represented in these diagrams should trace to the operational activities and operational use cases represented in OV-5 and to the systems identified in SV-1 and SV-2. The mapping between OV-5 to SV-4 use cases is not one-to-one; it is normally many-to-many going both ways. For those automated OV-5 operational activities, the operational activity maps to one or more systems use cases and to the realizations of the systems use cases via classes and class operations. SV-4 systems use cases depict system functional requirements, and actors represent entities that interface with the system functions.

Use case diagrams and class diagrams can show system functional requirements (use cases) and system functions — classes and class operations that realize the use cases. Class diagrams are intended to model the static design perspective of an automated system. As SV-4 products, the classes represent the design perspective of the systems identified in SV-1 that execute the system functions. You can specify or leave out attributes and operations for these classes, which show associated system data types and their permissible operations, depending on the level of implementation detail you need. You can use packages and containment relationships in conjunction with use case and class diagrams to show functional decomposition. Figure 40 depicts relationships between SV-4 use cases and the collection (packages) of classes that implement them.

Figure 40: UML Class Diagram for Systems Functionality Description (SV-4)



6.4.5 Data Element Definitions

Table 15 defines the architecture data elements for SV-4.

Table 15: Data Element Definitions for SV-4

Data Element	Attributes	Explanation
Graphical Box Types		
System Function*		See SV-1 Definition Table.

Data Element	Attributes	Explanation
External System Data Source/Sink*		
	Name*	Name/identifier for a system data source or sink (e.g., node, system, organization, or user) outside the scope of the systems being described
	Description*	Textual description of the external system data source or sink
System Data Repository/ Shared Database		
	Name	Name/identifier of system data store
	Description	Textual summary description of system data store
Graphical Arrow Types		
System Data Flow*		
	Name*	Name/identifier of system data flow (may be the same as a system data element name)
	Description*	Textual description of the system data flow
	From System Function/External System Data Source/System Data Repository*	Name of box entity from which the arrow originates
	To System Function/External System Data Sink/System Data Repository*	Name of box entity at which the arrow terminates
Function Decomposition Connector		
	Super Function	Name/Identifier of function that is being decomposed
Subfunction	Name/Identifier of system subfunction into which the super function decomposes	Subfunction
Referenced Types		
	See SV-1 Definition Table.	
	See SV-1 Definition Table.	
Relationships		
System Data Repository is Sink for System Data Flow		

Data Element	Attributes	Explanation
	System Data Repository Name	Name/identifier of a system data store
	System Data Flow Name	Name/identifier of a system data flow that is input to the system data store
System Data Repository is Source for System Data Flow		
	System Data Repository Name	Name/identifier of a system data store
	System Data Flow Name	Name/identifier of a system data flow that is output from the system data store
External Sink is Sink for System Data Flow*		May be system or human
	External Sink Name*	Name/identifier of an external sink
	System Data Flow Name*	Name/identifier of a system data flow that is input to the external sink
External Source is Source for System Data Flow*		May be system or human
	External Source Name*	Name/identifier of an external source
	System Data Flow Name*	Name/identifier of a system data flow that is output from the external source
System Function Produces System Data Flow*		
	System Function Name*	Name/identifier of system function
	System Data Flow Name*	Name/identifier of system data flow that is output from the system function
System Function Processes (Consumes) System Data Flow*		
	System Function Name*	Name/identifier of system function
	System Data Flow Name*	Name/identifier of system data flow 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

Data Element	Attributes	Explanation
System Function is Allocated to System*		
	System Function Name	Name/identifier of system function
	System Name	Name/identifier of system to which the function has been allocated
System Function Implements Operational Activity		
	System Function Name	Name/identifier of a system function
	Operational Activity Name	Name of an operational activity implemented by the system function
System Contains System Data Repository		
	System Name	Name/identifier of system
	System Data Repository Name	
		Name/identifier of system data store name contained in the system
System Function Uses Standard*		See TV-1 or TV-2
	System Function Name*	Name/identifier of system function
	Standard*	Name of standard

6.4.6 AFDM Support

GJXDM is being proposed for the AFDM. See Section 3.7 for information.

6.5 Operational Activity to Systems Function Traceability Matrix (SV-5)

6.5.1 Product Definition

The Operational Activity to Systems Function Traceability Matrix (SV-5) is a specification of the relationships between the set of operational activities applicable to an architecture and the set of system functions applicable to that architecture.

6.5.2 Product Purpose

SV-5 depicts the mapping of operational activities to system functions. Thus it identifies the transformation of an operational need into a purposeful action a system performs.

You can extend SV-5 to depict the mapping of capabilities to operational activities, operational activities to system functions, system functions to systems, and thus you can relate the capabilities to the systems that support them. Such a matrix allows decision makers and planners to quickly identify stove-piped systems, redundant or duplicative systems, gaps in capability, and possible future investment strategies, all in accord with the time stamp given the architecture. SV-5 correlates capability requirements that would *not* be satisfied if a specific system is not fielded to a specific organization.

6.5.3 Product Detailed Description

The PSAF uses the terms *activity* in the OVs and *function* in the SVs to refer to essentially the same kind of thing. Both activities and functions are tasks that are performed, that accept inputs, and that develop outputs. The distinction lies in the fact that system functions are executed by automated systems, while operational activities describe business operations that may be conducted by humans, automated systems, or both. Typical systems engineering practices use both of these terms, often interchangeably.

However, given the PSAF's use of activities on the operational side and functions on the systems side, and the fact that operational nodes do not map one-to-one to systems nodes, it is natural that operational activities do not map one-to-one to system functions. Therefore, SV-5 forms an integral part of the eventual complete mapping from operational capabilities to systems requirements.

SV-5 is an explicit link between the OV and SV. The capabilities and activities are drawn from OV-5, OV-6b, and OV-6c. The system functions are drawn from an SV-4. (SV-1 and SV-2 may also define system functions for identified systems.)

The relationship between operational activities and system functions can also be expected to be many-to-many. This means one operational activity may be supported by multiple system functions, and one system function may support multiple operational activities. Figure 41 provides a notional example of SV-5.

Figure 41: SV-5 Matrix

Operational Activities

System Functions	3.11	3.11.3	3.12	3.12.1	3.12.2	3.12.3	3.13	3.14	3.14.1	3.14.2	3.14.3	3.14.4	3.15	3.16	3.17	3.17.1
1	X															
1.1		X														
1.1.1			X													
1.1.1.1	X															
1.1.1.2					X											
1.1.1.3							X									
1.1.2										X						
1.1.2.1				X												
1.1.2.2						X										
1.1.2.3								X								
1.1.3											X					
1.1.3.1													X			
1.1.3.2									X							
1.1.3.3														X		
1.1.3.4														X		

. .

:

A key element of operational requirement traceability is the relation of system functions and operational activities to systems and capabilities. A capability may be defined in terms of the attributes required to accomplish a set of activities — such as the sequence and timing of activities, and the material that support them — to achieve a given mission objective. You can associate capability-related attributes with specific activities or with the information exchange between activities, or both.

You can use a particular operational thread to depict a capability. An operational thread is a set of operational activities, with sequence and timing attributes of the activities, and includes the information needed to accomplish the activities. In this manner, a capability is defined in terms of the attributes required to accomplish a given mission objective by modeling the set of activities and their attributes.

In addition, you can use SV-5 to map capabilities to operational activities, operational activities to system functions, and system functions to systems, and thus relate the capabilities to the systems that support them. First, you relate the activities to the system functions that automate them wholly or partially. Then, you associate a set of activities with a capability (as defined in OV-6c). By labeling the system functions with the systems that execute them (as defined in SV-1, SV-2, and SV-4), you can use SV-5 as a planner's matrix for analyses and decision support. The traceability from capabilities/activities to system/system functions is described by populating the SV-5 matrix.

Figure 42 provides a notional example of the extended SV-5 that provides a mapping between capabilities and systems.

Figure 42: Capability to System Traceability Matrix (SV-5)

		Capability 1			Capability 2			Capability 3			
		Operational Activity A	Operational Activity B	Operational Activity C	Operational Activity D	Operational Activity E	Operational Activity F	Operational Activity A	Operational Activity E	Operational Activity G	Operational Activity H
System1	System Function A	Red		Red				Red		Red	
	System Function B					Green			Yellow		
	System Function C		Red								Red
System 2	System Function B				Green						
	System Function D						Yellow			Yellow	
	System Function E				Red						
	System Function F										Yellow
System 3	System Function G			Green							
	System Function H		Green				Red				
	System Function I			Yellow							

Each mapping between an operational activity to a system function is described by a colored cell to indicate the status of the system support. Red indicates functionality planned but not developed. Yellow indicates partial functionality provided, or full functionality provided but the system has not been fielded. Green indicates full functionality provided and system fielded. A blank cell indicates that there is no system support planned for an operational activity, or that a relationship does not exist between the operational activity and the system function.

In this manner, the association between a certain capability and a specific system can be illustrated via a many-to-many relationship: *many* operational activities contribute to a capability, and *many* system functions are executed by a system.

6.5.4 UML Representation

There is no equivalent for this product in UML. However, the information for an SV-5 matrix may be gathered from OV-5 and SV-4 element descriptors (e.g., <<include>> relationships between OV-5 use cases and SV-4 use cases). You can use a UML modeling tool script to extract the underlying architecture data and convert it to matrix form.

Since SV-4 use cases are refinements of OV-5 use cases, the relationship between the operational activities and associated capabilities on the OV side, and the system functions on the SV side, is already defined. You can also allocate system functions to systems in UML by allocating SV-4 classes and packages to UML components from SV-1, accomplishing the intent of SV-5.

6.5.5 Data Element Definitions

Table 16 describes the architecture data elements for SV-5.

Table 16: Data Element Definitions for SV-5

Data Element	Attributes	Explanation
Graphical Box Types		
System Function*		See SV-1 Definition Table.
External System Data Source/Sink*		
Referenced Types		
Capability		See OV-5 Definition Table.
System		See SV-1 Definition Table.
Operational Activity*		See OV-5 Definition Table.
System Function*		See SV-4 Definition Table.
Relationships		
System Function of a System Supports Operational Activity for a Capability		
	System Function Name	Name/identifier of system function
	Operational Activity Name	Name/identifier of an operational activity that is associated with the capability and supported by the system function
	System Name	Name/identifier of system that executes the system function
	Capability Name	Name/identifier of a capability associated with the operational activity and supported by the system
	Support Status Code	One of red, yellow, or green. Red means functionality planned but not developed; yellow means partial functionality provided or full functionality provided but system has not been fielded; green means full functionality provided and system has been fielded
System Function Implements Operational Activity*		
	System Function Name*	Name/identifier of system function
	Operational Activity Name*	Name/identifier of operational activity implemented, at least partially, by the system function

6.5.6 AFDM Support

GJXDM is being proposed for the AFDM. See Section 3.7 for information.

6.6 Systems Data Exchange Matrix (SV-6)

6.6.1 Product Definition

The Systems Data Exchange Matrix (SV-6) specifies the characteristics of the system data exchanged between systems. This product focuses on automated information exchanges from OV-3 that are implemented in systems. Non-automated information exchanges, such as verbal orders, are captured in the OV products only.

6.6.2 Product Purpose

System data exchanges express the relationship across the three basic architecture data elements of an SV (systems, system functions, and system data flows). They focus on the specific aspects of the system data flow and the system data content. These aspects of the system data exchange can be crucial to the operational mission. They are critical to understanding the potential for overhead and constraints introduced by the physical aspects of the implementation.

6.6.3 Product Detailed Description

SV-6 describes, in tabular format, system data exchanged between systems. The focus of SV-6 is on how the system data exchange is implemented, in system-specific details covering periodicity, timeliness, throughput, size, information assurance, and the security characteristics of the exchange. In addition, the matrix also describes the system data elements, their format and media type, accuracy, units of measurement, and system data standard..

SV-6 relates to, and grows out of, OV-3. The operational characteristics for the OV-3 information exchange are replaced with the corresponding system data characteristics. For example, performance attributes for the operational information exchanges are replaced by the actual system data exchange performance attributes for the automated portions of the information exchange.

On SV-6, each operational needline is decomposed into the interfaces that are the systems equivalents of the needline. SV-1 graphically depicts system data exchanges as interfaces that represent the automated portions of the needlines. The implementation of SV-1 interfaces is described in SV-2, if applicable. The system data exchanges documented in SV-6 trace to the information exchanges detailed in OV-3. They constitute the automated portions of the OV-3 information elements.

Figure 43 shows a template for this product. The data element definition table for SV-6 contains detailed descriptions or references for each matrix column.

Figure 43: SV-6 – Template

Interface Identifier	Data Exchange Identifier	Data Description							Producer	
System Interface Name and ID.	System Data Exch. Name and ID.	Data Element. Name and ID.	Content	Format Type	Media Type	Units	Accuracy	Data Standard.	Send System Name and ID.	Send Sys. Func. Name and ID.

Consumer		Nature of Transaction				Performance Attributes				Information Assurance				
Rec. System Name and ID.	Rec. Sys. Func. Name and ID.	Transaction Type	Triggering Event	Interop Level Required	Criticality	Periodicity	Timeliness	Throughput	Size	Access Control	Availability	Confidentiality	Dissemination Control	Integrity

Note that each system data element exchanged is related to the system function from SV-4 that produces or consumes it -- via the leaf inputs and outputs of the system functions. However, there may not be a one-to-one correlation between system data elements listed in the matrix and the data flows (inputs and outputs) that are produced or consumed in a related SV-4. System data inputs and outputs between system functions performed at the same systems node — that is, not associated with an interface on SV-1 — will not be shown in the SV-6 matrix.

System data inputs and outputs between functions for some levels of functional decomposition may be at a higher level of abstraction than the system data elements in the SV-6 matrix. In this case, multiple system data elements will map to a single function's system data flow. Similarly, the system data flows between functions at a low level of functional decomposition may be at a finer level of detail than the system data elements in the SV-6 matrix. Multiple system data flows may map to a single system data element.

6.6.4 UML Representation

There is no equivalent for this product in UML. However, this matrix product expands on the information associated with SV-1 systems, SV-4 use cases, and system data flows. If you use an automated tool to create the other products in UML, then you can generate SV-6 expanded definitions from descriptors to the applicable SV-1 systems and interfaces and SV-4 system functions and system data flows.

6.6.5 Data Element Definitions

The SV-6 template (Figure 43) illustrates how the end product should look to the user. Table 17 describes the architecture data elements for SV-6.

Table 17: Data Element Definitions for SV-6

Data Element	Attributes	Explanation
Non-Graphical Types		
System Data Exchange*		
	Identifier*	Identifier of system data exchange – usually based on the relevant operational needline, system interface, and information exchange
	Name*	Name of system data exchange – usually based on the relevant operational needline, system interface, and information exchange
	Sending System Name*	Name/identifier of system that produces the system data
	Sending System Function Name*	If used, the identity of the system function producing the system data
	Receiving System Name*	Name/identifier of the system that consumes the system data
	Receiving System Function Name*	The identity of the system function consuming the system data
	Transaction Type*	Descriptive field that identifies the type of exchange
	Triggering Event*	Brief textual description of the events that triggers the system data exchange
	Interoperability Level Achievable	Interoperability level achieved or achievable through the exchange
	Periodicity*	Frequency of system data exchange transmission – may be expressed in terms of worst case or average frequency

Data Element	Attributes	Explanation
	Criticality	The criticality assessment of the information being exchanged in relationship to the mission being performed
	Timeliness*	How much delay this system data can tolerate and still be relevant to the receiving system
	Throughput*	Bits or bytes per time period – may be expressed in terms of maximum or average throughput required
	Size*	Size of system data
	Integrated Architecture — Access Control	The class of mechanisms used to ensure only those authorized can access a specific system data element
	Integrated Architecture — Availability	The relative level of effort required to be expended to ensure the system data can be accessed
	Integrated Architecture — Confidentiality	The kind of protection required for system data to prevent unintended disclosure
	Integrated Architecture — Dissemination Control	The kind of restrictions on receivers of system data based on sensitivity of system data
	Integrated Architecture — Integrity	The kind of requirements for checks that the content of the system data element has not been altered
	Integrated Architecture — Non-Repudiation Consumer	The requirements for definitive knowledge that the system data sent was consumed by the intended recipient
	Integrated Architecture — Non-Repudiation Producer	The requirements for definitive knowledge that the system data received was produced by the stated source
	Timeliness*	How much delay this system data can tolerate and still be relevant to the receiving system
	Protection Type Name	The name for the type of protection
	Protection Duration Code	The code that represents how long the system data must be safeguarded
	Protection Suspense Calendar Date	The calendar date on which the designated level of safeguarding discontinues for a specific system data element
	Classification*	Classification code for the system data element

Data Element	Attributes	Explanation
	Classification Caveat*	A set of restrictions on system data of a specific classification Supplements a security classification with system data on access, dissemination, and other types of restrictions
	Releasability	The code that represents the kind of controls required for further dissemination of system data
	Security Standard	See TV-1/2 Definition Table.
System Data Element*		
	Identifier*	Identifier of system data element. Based on the relevant SV-4 system data flow, correlates to OV-3 information element
	Name*	Name of system data element. Based on the relevant SV-4 system data flow, correlates to OV-3 information element
	Content*	The system data that is carried by the exchange
	Format Type*	Application level format (e.g., XML, EDI, ASCII Text) with parameters and options used, or other relevant protocol
	Media Type*	Type of media
	Accuracy*	Description of the degree to which the system data conforms to actual fact as required by the system or system function
	Units of Measurement*	Units used for system data
	System Data Standard*	For example, Justice Data Dictionary XML (See TV-1/2 Definition Table.)
Referenced Types		
Information Exchange*		See OV-3 Definition Table.
Interface*		See SV-1 Definition Table.
System*		See SV-1 Definition Table.
System Function*		See SV-4 Definition Table.
System Data Flow*		See SV-4 Definition Table.
Triggering Event*		See SV-6c Definition Table.

Data Element	Attributes	Explanation
Standard*		See TV-1 Definition Table.
Relationships		
System Data Exchange is Carried by Interface*		
	System Data Exchange Identifier*	Identifier for a system data exchange
	Interface Name*	Name/identifier for an interface (from SV-1)
System Data Exchange Automates Information Exchange*		
	System Data Exchange Identifier*	Identifier for a system data exchange
	Information Exchange Identifier*	Name/identifier for an information exchange (from OV-3)
System Data Exchange has Sending System*		
	System Name*	Name of the system that sends the system data exchange
	System Data Exchange Identifier*	Identifier of the system data exchange sent
System Data Exchange has Receiving System*		
	System Name*	Name of the system that receives the system data exchange
	System Data Exchange Identifier*	Identifier of the system data exchange received
System Data Exchange has Sending System Function*		
	System Function Name*	Name of the system function, at the originating node of the interface, that sends the system data exchange
	System Data Exchange Identifier*	Identifier of the system data exchange sent
System Data Exchange has receiving System Function*		
	System Function Name*	Name of the system function, at the receiving node of the interface, that receives the system data exchange

Data Element	Attributes	Explanation
	System Data Exchange Identifier*	Identifier of the system data exchange received
System Data Element is Exchanged Via System Data Exchange*		
	System Data Element Identifier*	Identifier of the system data element
	System Data Exchange Identifier*	Identifier of the system data exchange
System Data Exchange has Triggering Event*	Event Name*	Name of the event that triggers the system data exchange
	System Data Exchange Identifier*	Identifier of the system data exchange
System Data Element Uses Standard		See TV-1/2 Definition Table.
	System Data Element Identifier	Identifier for a system data element
	Standard Name	Name/ID number of a standard used by the system data element
System Data Element is Associated with System Data Flow		
	System Data Element Name	Name/identifier of System data element associated with the system data flow
	System Data Flow Name	See SV-4 Definition Table.
System Data Exchange Conforms to Security Standard*		See TV-1/2 Definition Table.
	System Data Exchange Identifier*	Identifier for a system data exchange
	Standard Name*	Name/ID number of a standard used by the system data exchange

6.6.6 AFDM Support

GJXDM is being proposed for the AFDM. See Section 3.7 for information.

6.7 Systems Performance Parameters Matrix (SV-7)

6.7.1 Product Definition

The Systems Performance Parameters Matrix (SV-7) product specifies the quantitative characteristics of systems and system hardware and software items, their functions, and their interfaces. The interfaces refer to system data carried by the interface as well as communications link details that implement the interface..

SV-7 specifies the current performance parameters of each system, interface, or system function, and the expected or required performance parameters at specified times in the future. Performance parameters include all technical performance characteristics of systems for which you can develop requirements and define specifications. The complete set of performance parameters may not be known at the early stages of architecture definition, so you should expect to update this product throughout the system's specification, design, development, testing, and possibly even its deployment and operations life cycle phases.

6.7.2 Product Purpose

One of the primary purposes of SV-7 is to communicate which characteristics are considered most crucial for the successful achievement of the mission goals assigned to the system. These particular parameters can often be the deciding factors in acquisition and deployment decisions. They will figure strongly in systems analyses and simulations done to support the acquisition decision processes and system design refinement.

6.7.3 Product Detailed Description

SV-7 builds on SV-1, SV-2, SV-4, and SV-6 by specifying performance parameters for systems and system hardware and software items and their interfaces (defined in SV-1), communications details (defined in SV-2), functions (defined in SV-4), and system data exchanges (defined in SV-6). The term *system*, as defined for this product and all others in the PSAF, may represent a FoS, SoS, network of systems, or an individual system.

You also describe in this product the performance parameters for system hardware and software items -- the hardware and software elements comprising a system. Performance parameters often relate to a system function being performed. Therefore, this product also may show system functions and their performance attributes. If the future performance expectations are based on expected technology improvements, then you should coordinate the performance parameters and their time periods with a SV-9. If you associate performance improvements with an overall system evolution or migration plan, coordinate the time periods in SV-7 with the milestones in a SV-8.

Figure 44 is a template of SV-7, listing notional performance characteristics with a time period association.

Figure 44: SV-7 – Notional Example

System Name	Performance Range (Threshold and Objective) Measures		
	$Time_0$ (Baseline Architecture Time Period)	**$Time_1$**	**Tim e_n (Target Architecture Time Period)**
Hardware Element 1			
Maintainability			
Availability			
System Initialization Time			
Architecture Data transfer rate			
Program Restart Time			
Software Element 1 / Hardware Element 1			
Automatic Processing Responses (by input type, # of input types			
Operator Interaction Response Times (by type)			
Availability			
Effectiveness			
Mean Time Between S/W Failures			
Organic Training			
Software Element 2 / Hardware Element 1			
Hardware Element 2			

6.7.4 UML Representation

There is no equivalent to this product in UML.

6.7.5 Data Element Definitions

Table 18 describes the architecture data elements for SV-7.

Table 18: Data Element Definitions for SV-7

Data Element	Attributes	Explanation
Non-Graphical Types		
Performance Parameter Set*		
	Name*	Name/identifier of parameter set
	Number of Performance Parameters in Set*	Number of different performance parameters for which measures were or will be taken
Performance Parameter Type*		
	Name*	Name/identifier of performance parameter type (e.g., mean time between failures, maintainability, availability, system initialization time, system data transfer rate, program restart time for platforms; and system data throughput/capacity; response time, effectiveness, mean time between software failures for application software)
	Description*	Textual description of the performance parameters and what their measurements mean
Time*		
	Time Identifier*	Positive integer that identifies the position of the column of measurements (i.e., first column of measurements, second column, etc.)
	Timestamp*	Time when the measurements were or will be taken
Performance Measurement*		
	Performance Parameter Type Name*	Name/identifier of a performance parameter type and unit of measurement
	Time Identifier*	Positive integer identifying the position of the column of a particular past time
	Measured Value*	Value of performance parameter type that was measured at the indicated time
Required Performance Parameter Range*		

Data Element	Attributes	Explanation
	Performance Parameter Name*	Name/identifier of a performance parameter
	Performance Parameter Type Name*	Name/identifier of a performance parameter type for which the performance parameter range is applied
	Time Identifier*	Positive integer identifying the position of the column of a particular future time
	Threshold Value*	Value of performance parameter type that is acceptable at the indicated time
	Objective Value	The objective value for a performance parameter is the desired operational goal, beyond which any gain in utility does not warrant additional expenditure. Advances in technology or changes in joint concepts and integrated architectures may result in future incremental changes to objective values
Referenced Types		
System*		See SV-1 Definition Table.
System Hardware/Software Item		See SV-1 Definition Table.
Communications Systems		See SV-2 Definition Table.
Communications Link		See SV-2 Definition Table.
System Function		See SV-4 Definition Table.
System Data Exchange		See SV-6 Definition Table.
Timed Technology Forecast		See SV-9 Definition Table.
Milestone		See SV-8 Definition Table.
Relationships		
System Contains System Hardware/Software Item		See SV-1 Definition Table.
System has Performance Parameter Set*		
	System Name*	Name/identifier of a system
	Performance Parameter Set Name*	Name of the performance parameter set
System Hardware/Software Item has Performance Parameter Set		

Data Element	Attributes	Explanation
	System Hardware/Software Item Name	Name/identifier for system hardware/software item
	Performance Parameter Set Name	Name of the performance parameter set
Communications System has Performance Parameter Set		
	Communications System Name	Name/identifier for communications system
	Performance Parameter Set Name	Name of the performance parameter set
Communications Link has Performance Parameter Set		
	Communications Link Name	Name/identifier for communications link
	Performance Parameter Set Name	Name of the performance parameter set
System Function has Performance Parameter Set		
	System Function Name	Name/identifier for system function
	Performance Parameter Set Name	Name of the performance parameter set
System Data Exchange has Performance Parameter Set		
	System Data Exchange Name	Name/identifier for system data exchange
	Performance Parameter Set Name	Name of the performance parameter set
Performance Parameter Set Includes Performance Measurement*		
	Performance Parameter Set Name*	Name/identifier of performance parameter set
	Performance Measurement Name*	Name/identifier of performance measurement to be included in performance parameter set
Performance Parameter Set Includes Required Performance Parameter Range*		
	Performance Parameter Set Name*	Name/identifier of performance parameter set
	Performance Parameter Range Identifier*	Identifier of performance parameter range to be included in performance parameter set
Required Performance Range Depends on Timed Technology Forecast		

Data Element	Attributes	Explanation
	Timed Technology Forecast Name*	Name/identifier of a forecast regarding a specific technology
	Performance Parameter Range Identifier*	Identifier of the performance parameter range related to the timed technology forecast
Required Performance Range Supports Evolution Milestone		
	Performance Parameter Range Identifier*	Identifier of the performance parameter range supporting the milestone
	Milestone Name*	Name/identifier of a system evolution milestone

6.7.6 AFDM Support

GJXDM is being proposed for the AFDM. See Section 3.7 for information.

6.8 System Evolution Description (SV-8)

6.8.1 Product Definition

The Systems Evolution Description (SV-8) captures evolution plans that describe how the system, or the architecture in which the system is embedded, will evolve over a lengthy period of time. Generally, the timeline milestones are critical for a successful understanding of the evolution timeline.

6.8.2 Product Purpose

SV-8, when linked with other evolution products such as SV-9 and TV-2, provides a clear definition of how the architecture and its systems are expected to evolve over time. In this manner, you can use the product as an architecture evolution project plan or transition plan.

6.8.3 Product Detailed Description

SV-8 describes plans for *modernizing* system functions 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). The two thrusts will often be combined.

This product builds on other products and analyses in that planned capabilities and information requirements that relate to performance parameters of SV-7, and technology forecasts of SV-9 are accommodated in this product. The template for SV-8 consists of two generic examples. If the architecture describes a communications infrastructure, then you can describe a planned evolution or migration of communications systems, communication links, and their associated standards in this product. Figure 45 illustrates a migration description for Project 25, while Figure 46 illustrates evolution. All entries in the graphics are for illustration only.

Figure 45: SV-8 – Migration

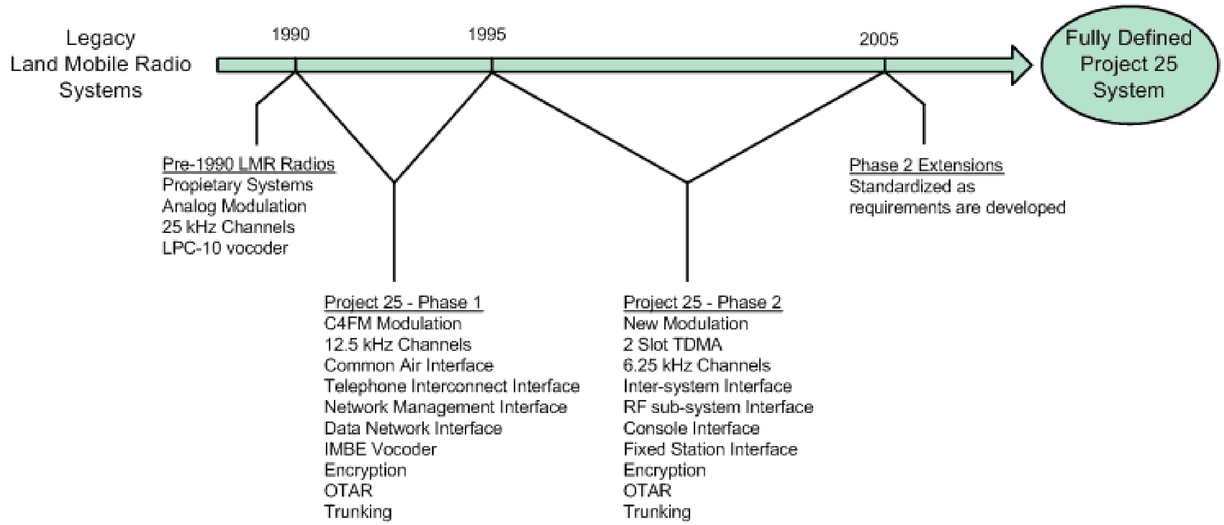
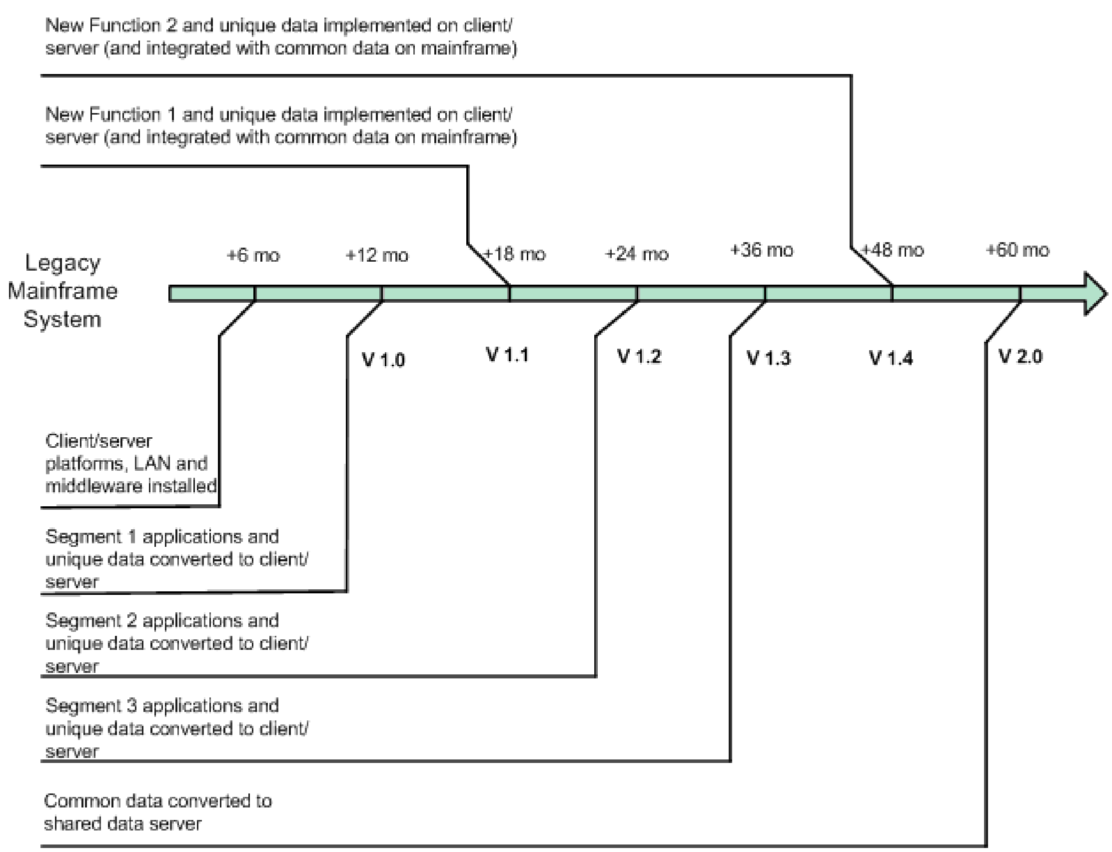


Figure 46: SV-8 – Evolution



6.8.4 UML Representation

There is no equivalent to this product in UML. However, performance attributes may be specified for all system elements where they apply using UML adornments. Furthermore, much of the desired purpose can be achieved by defining sets of UML products for different periods of system evolution, thus allowing a comparison between time periods to highlight the evolutionary changes. Textual descriptions of the evolution steps should be included.

6.8.5 Data Element Definitions

Table 19 describes the architecture data elements for SV-8.

Table 19: Data Element Definitions for SV-8

Data Element	Attributes	Explanation
Non-Graphical Types		
System Group*		
	Name*	Name/identifier for a set of systems, subsystems, or system hardware/software items
	Description*	Textual description of system group
System*		See SV-1 Definition Table.
System Hardware/Software Item		See SV-1 Definition Table.
Communications System		See SV-2 Definition Table.
Communications Link		See SV-2 Definition Table.
Communications Network		See SV-2 Definition Table.
Time Period*		
	Identifier*	Identifier for the time period
	Date*	A specific date in time
Timeline*		
	Name*	Name of timeline
	Description*	Textual description of purpose of timeline, including goals of the migration or evolution process being documented
	Beginning Time Period*	Date of beginning of timeline

Data Element	Attributes	Explanation
	Ending Time Period*	Date of ending of timeline
Milestone*		
	Name*	Name/identifier for milestone
	Time Period Name*	Name of time period indicating a specific time in date
	Description*	Goals to be achieved at milestone
Graphical Arrow Types		
System Grouping Link*		Graphical line that connects a system group to a milestone or timeline
	Milestone or Timeline Name*	Name/identifier of the milestone or timeline when this grouping should complete integration
	System Group Name*	Name/identifier for a set of systems, subsystems, or system hardware/software items
	Number of Constituent Systems, Subsystems or System Hardware/Software Items*	Number of systems, subsystems, or system hardware/software items grouped together
Referenced Types		
Timed Technology Forecast*		See SV-9 Definition Table.
Standard*		See TV-1 Definition Table.
Relationships		
System Group Contains Constituent System, Subsystem, System Hardware/Software Item, Communication Link, Communications Network*		
	System Group Name*	Name/identifier for a set of systems, subsystems, system hardware/software items, communication link, or communications network
	System, Subsystem, or System Hardware/Software Item, Communication Link, or Communications Network Name*	Name of systems, subsystems, or system hardware/software items, communication link, or communications network contained in the system group
Milestone is Completed with System Group*		

Data Element	Attributes	Explanation
	Milestone Name*	Name/identifier of milestone
	System Group Name*	Name/identifier for a set of upgraded systems, subsystems, or system hardware/software items required to complete a milestone
	System Version*	Version name/number for system configuration at the completion of milestone
Timeline has Beginning System Configuration*		
	Timeline Name*	Name/identifier of timeline
	System Name*	Name of a system for system evolution timelines
	System Version*	Version name/number for system configuration at the beginning of timeline for system evolution
Timeline has Ending New System*		
	Timeline Name*	Name/identifier of timeline
	System Name*	Name of new system or subsystems available at end of timeline
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)
Milestone Requires Timed Technology Forecast		
	Milestone Name	Name/identifier of milestone
	Timed Technology Forecast Name	Name/identifier of a timed technology forecast for a specific technology required for supporting the system associated with the milestone. That is, milestone cannot be met if technology forecasted is not available
Milestone Includes Upgrade to Existing Standard		
	Milestone Name	Name/identifier of milestone

Data Element	Attributes	Explanation
	Standard Name	Name/identifier of a standard from TV-1 technical standards profile that will be used in system, subsystem, or system hardware/software items integrated for the milestone

6.8.6 AFDM Support

GJXDM is being proposed for the AFDM. See Section 3.7 for information.

6.9 Systems Technology Forecast (SV-9)

6.9.1 Product Definition

The Systems Technology Forecast (SV-9) defines the underlying current and expected supporting technologies. It is not expected to include predictions of technologies. Expected supporting technologies are those that can be reasonably forecast given the current state of technology and expected improvements. New technologies should be tied to specific time periods, which can correlate against the time periods used in SV-8 milestones.

6.9.2 Product Purpose

SV-9 provides a summary of emerging technologies that affect the architecture and its existing planned systems. The focus should be on the supporting technologies that may most affect the capabilities of the architecture or its systems.

6.9.3 Product Detailed Description

SV-9 provides a detailed description of emerging technologies and specific hardware and software products. It contains predictions about the availability of emerging technological capabilities and about industry trends in specific time periods. The specific time periods selected (e.g., 6-month, 12-month, 18-month intervals) and the technologies being tracked should be coordinated with architecture transition plans (see SV-8). This is because the insertion of new technological capabilities and upgrading of existing systems may depend on or be driven by the availability of new technology.

The forecast includes potential technology impacts on current architectures, and thus influences the development of transition and objective, or target, architectures. You want to focus the forecast on technology areas related to the purpose for which you are describing a given architecture, and identify issues that will affect the architecture. If standards are an integral part of the technologies important to the evolution of a given architecture, then it may be convenient to combine SV-9 with the Technical Standards Forecast (TV-2).

SV-9 is constructed as part of a given architecture and in accordance with the architecture purpose. Typically, this will involve starting with one or more overarching reference models or standards profiles, for example Project 25, to which the architecture is subject to using. Using a reference model or standards profile, the architecture should select the service areas and services relevant to it.

SV-9 forecasts relate to TV-1 in that a timed technology forecast may contribute to the decision to retire or phase out the use of a certain standard in connection with a system element. Similarly, SV-9 forecasts relate to TV-2 standards forecasts in that a certain standard may be adopted depending on a certain technology becoming available (e.g., the availability of Java Script may influence the decision to adopt a new HTML standard).

Table 20 shows a template for SV-9.

Table 20: SV-9 – Notional Example

Service	**Technology Forecasts**		
	Short Term (0-6 Months)	**Mid Term (6-12 Months)**	**Long Term (12-18 Months)**
Application Software			
Support Applications	Microsoft Office 2000 available (for Windows 2000)	Microsoft Office 2000 stable enough for full-scale implementation	Microsoft Office available for Linux E-mail on wireless PDAs commonplace
Application Platform			
Data Management	Oracle 9i available MySQL (Opensource DBMS) available		
Operating System		Next MS Windows server upgrade expected Next Red Hat Linux major release expected	Next MS Windows server upgrade expected
Physical Environment			Intel IA-64 becomes standard processor for desktops initial use of quantum computing technologies
External Environment			
User Interface		Thin screen LCD monitors for PC desktops become price-competitive	Thin screen LCD monitors become price-competitive for desktops. Conventional CRT monitors for desktops become obsolete.
Persistent Storage	5G PCMIA-type card available		Disk storage capacity doubles

Service	Technology Forecasts		
	Short Term (0-6 Months)	**Mid Term (6-12 Months)**	**Long Term (12-18 Months)**
			again.
Communications Networks		Cable modem service available for most telecommuting staff	Fiber optic connections available for most telecommuting staff

Alternatively, SV-9 may relate technology forecasts to SV elements, for example, systems, where applicable. You can include the list of systems potentially affected by the technology directly in SV-9 by specifying a time period in the cell corresponding to the system element and the applicable service area and service. Table 21 is a template showing this variant of SV-9.

Table 21: SV-9 – Template

			SV-1 and SV-2 Systems (includes SOS, FOS, Subsystem, Communications Systems)	SV-1 and SV-2 Hardware or Software Item	SV-2 Communications (Physical) Link	SV-4 System Function	SV-6 System Data Element	OV-7 SV-11 Model Standard or Source
Service Area	Service	Technology Forecast (Summary Prediction)	Application System ID or Name (with Time Period If Applicable)	Hardware or Software ID or Name, Version, (with Time Period If Applicable)	System Data Link ID or Name (with Time Period If Applicable)	System Function ID or Name	System Data Exchange ID or Name (with Time Period If Applicable)	Model Standard or Source ID or Name (with Time Period If Applicable)

6.9.4 UML Representation

There is no equivalent to this product in UML.

6.9.5 Data Element Definitions

Table 22 describes the architecture data elements for SV-9.

Table 22: Data Element Definitions for SV-9

Data Element	Attributes	Explanation
Non-Graphical Types		
Technologies Forecast Profile*		
	Name*	Name/identifier of technologies forecast profile
	Description*	Textual description of purpose of forecast
Timed Technology Forecast*		
	Name*	Name/identifier of a forecast regarding a specific technology
	Technology*	Textual description of a future capability for a specific technology being forecasted
	Time Period*	Time period for which forecast is valid; usually expressed in terms of a future date or months from baseline
	Discussion*	Textual notes regarding technology status, likely commercial market acceptance, and risk assessment of adopting the technology forecasted
Referenced Types		
Reference Model		See TV-1 Definition Table.
Service Area		See TV-1 Definition Table.
Service		See TV-1 Definition Table.
System		See SV-1 Definition Table.
System Hardware/Software Item		See SV-1 Definition Table.
Communications System		See SV-2 Definition Table.
Communications Link		See SV-2 Definition Table.
Communications Network		See SV-2 Definition Table.
Time Period		See SV-8 Definition Table.
Standards Profile		See TV-1 Definition Table.
Standard		See TV-1 Definition Table.

Data Element	Attributes	Explanation
Timed Standards Forecast*		See TV-2 Definition Table.
Relationships		
Technologies Forecast Profile Covers Service Area/Service*		
	Technologies Forecast Profile Name*	Name/identifier of technologies forecast profile
	Service Area/Service Name*	Name/identifier of a TRM service area or service covered by the technologies forecast profile
Service Area/Service has Timed Technology Forecast*		
	Service Area/Service Name*	Name/identifier of a service area or service
	Timed Technology Forecast Name*	Name/identifier of a specific, time-sensitive forecast for technology relevant to the service area or service
Technologies Forecast Profile is Based on Reference Model or Standards Profile*		
	Technologies Forecast Profile Name*	Name/identifier of the technologies forecast profile
	Reference Model or Standards Profile Name*	Name/identifier of a system that could be affected by the technology forecast
Timed Technology Forecast Affects System		
	Timed Technology Forecast Name	Name/identifier of a timed technology forecast
	System	Name/identifier of a system hardware/software item that could be affected by the technology forecast
Timed Technology Forecast Affects System Hardware/Software Item		
	Timed Technology Forecast Name	Name/identifier of a timed technology forecast
	System Hardware/Software Item	Name/identifier of a communications link that could be affected by the technology forecast
Timed Technology Forecast Affects Communications Link		

Data Element	Attributes	Explanation
	Timed Technology Forecast Name	Name/identifier of a timed technology forecast
	Communications Link Name	Name/identifier of a communications network that could be affected by the technology forecast
Timed Technology Forecast Affects Communications Network		
	Timed Technology Forecast Name	Name/identifier of a timed technology forecast
	Communications Network Name	Name/identifier of a communications network that could be impacted by the technology forecast
Timed Technology Forecast Affects Communications System		
	Timed Technology Forecast Name	Name/identifier of a timed technology forecast
	Communications System Name	Name/identifier of a communications system (e.g., gateway, router, satellite) that the technology forecast could affect
Timed Technology Forecast Corresponds to Time Period		
	Timed Technology Forecast Name	Name/identifier of a timed technology forecast
	Standard Name	Name/identifier of a standard
Timed Technology Forecast is Required by Timed Standards Forecast		
	Timed Technology Forecast Name	Name/identifier of a timed technology forecast
	Timed Standards Forecast Name	Name/identifier of a timed standards forecast

6.9.6 AFDM Support

GJXDM is being proposed for the AFDM. See Section 3.7 for information.

6.10 Systems Functionality Sequence and Threads (SV-10)

6.10.1 Product Definition

The Systems Functionality Sequence and Threads (SV-10) products describe the dynamic behavior and performance characteristics of an SV. Many of the critical characteristics of an architecture are only discovered when an architecture's dynamic behaviors are defined and described. These dynamic behaviors concern the timing and sequencing of events that capture system performance characteristics of an executing system (i.e., a system performing the system functions described in SV-4).

6.10.2 Product Purpose

Behavior modeling and documentation are key to a successful architecture description. How the architecture behaves is crucial in many situations. Although knowledge of the functions and interfaces is also crucial, knowing whether, for example, a response should be expected after sending message X to node Y can be crucial to successful overall operations.

6.10.3 UML Representation

The SV-10 product includes three types of models to describe the dynamic behavior and performance characteristics of a SV. These three models are:

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

6.10.4 Data Element Definitions

You can use SV-10b and SV-10c separately or together, as necessary, to describe critical timing and sequencing behavior in the SV. A wide variety of different systems methodologies use both types of diagrams.

Both SV-10b and SV-10c 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 described in SV-10a.

6.10.5 AFDM Support

GJXDM is being proposed for the AFDM. See Section 3.7 for information.

6.11 Systems Rules Model (SV-10a)

6.11.1 Product Definition

Systems rules (SV-10a) are constraints on an architecture, on systems, or system hardware or software items, or on a system functions. While other SV products (for example, SV-1, SV-2, SV-4, SV-11) describe the static structure of the Systems View — what the systems can do — they do not describe, for the most part, what the systems *must* do, or what they *cannot* do.

At the systems or system hardware or software items level, SV-10a describes the rules under which the architecture or its systems behave under specified conditions. At lower levels of decomposition, it may consist of rules that specify the pre- and post-conditions of system functions. You can express such rules in a textual form, for example, "If (these conditions) exist, and (this event) occurs, then (perform these actions)."

6.11.2 Product Purpose

The purpose of this product is to allow understanding of behavioral rules and constraints imposed on systems and system functions.

6.11.3 Product Detailed Description

Rules are statements that define or constrain some aspect of the enterprise. In contrast to OV-6a, SV-10a focuses on constraints imposed by some aspect of operational performance requirements that translates into system performance requirements. At a lower level of detail, it focuses on some aspects of systems design or implementation. Thus, just as you can associate the operational rules with OV-5, you can also associate the systems rules in SV-10a with SV-1 and SV-2 systems and hardware or software items, or with SV-4 system functions.

Systems rules can be grouped into the following categories:

- Structural Assertion: These rules concern the implementation of business domain terms and facts that are usually captured in the file structures or physical database schemas. These assertions reflect static aspects of the implementation of business rules that may be already captured in OV-7. (Sometimes these rules are embedded in application code.)
  - Terms: Entities, records
  - Facts: Association between two or more terms (i.e., relationship)
- Action Assertion: These rules concern some dynamic aspect of system functioning and specify constraints on the results of system functions or applications.
  - Condition: Guard or if portion of If-Then-Else 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 system functions or applications to certain human roles or class of users

- Derivation: These rules concern algorithms to compute a derivable fact from other terms, facts, derivations, or action assertions.

Because the structural assertion rules are frequently captured in the architecture domain system data model, SV-10a usually focuses on the more dynamic action assertions and derivations rules. Additional rule characteristics include:

- 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

Each architecture may select the formal language in which to record its SV-10a. The notation selected should be referenced and well-documented (i.e., there should be text books or articles that describe it and provide examples of its use).

Figure 47 illustrates an example action assertion that might be part of a SV-10a. The assertion is an example of one that might be necessary midway through a system migration, when the databases that support three Forms (FORM-X, FORM-Y, and FORM-Z) have not yet been integrated. Thus, explicit user or application action is needed to keep the related system data synchronized. The following example is in a form of structured English.

Figure 47: SV-10a – Action Assertion Example

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*

6.11.4 SV-10-a UML Representation

There is no equivalent diagram in UML for SV-10a. However, if you consider systems rules to be equivalent to complex, nested If-Then-Else and CASE statements, you can derive these statements unambiguously from UML statechart diagrams for the object classes defined as systems, system functions or system data. You can specify pre- and post-conditions for class operations as well as use cases of SV-4. SV-10a may be generated via the use of adornments, and the inclusion of guard conditions on the statecharts of SV-10b, and pre- and post-conditions on classes and use cases of SV-4.

6.11.5 SV-10a Data Element Definitions

The architecture data elements for SV-10a should capture the type of the rule (e.g., action assertion or derivation) and the text for the rule, as well as the relationship between the rules and other architecture data elements. Table 23 describes the architecture data elements for SV-10a.

Table 23: Data Element Definitions for SV-10a

Data Element	Attributes	Explanation
Non-Graphical Types		
Rule Model		See OV-6a Definition Table.
Rule*		See OV-6a Definition Table.
Referenced Types		
System Function*		See SV-4 Definition Table.
Relationships		
Rule Model Includes Rule		See OV-6a Definition Table.
Rule Applies to System Function*		If any rules are related to activities
	Rule Name*	Name of action assertion or derivation rule
	System Function Name*	Name of system function to which the rule applies

6.11.6 AFDM Support

GJXDM is being proposed for the AFDM. See Section 3.7 for information.

6.12 Systems State Transition Description (SV-10b)

6.12.1 Product Definition

The Systems State Transition Description (SV-10b) is a graphical method of describing a system or system function response to various events by changing its state. The diagram basically represents the sets of events to which the systems in the architecture will respond (by taking an action to move to a new state) as a function of its current state. Each transition specifies an event and an action.

6.12.2 Product Purpose

The explicit time sequencing of system functions in response to external and internal events is not fully expressed in SV-4. You can use SV-10b to describe the explicit sequencing of the system functions. Alternatively, you can use SV-10b to reflect explicit sequencing of the actions internal to a single system function, or the sequencing of system functions with respect to a specific system.

Basically, statechart diagrams can be unambiguously converted to structured textual rules that specify timing aspects of systems events and the responses to these events, with no loss of meaning. However, the graphical form of the state diagrams can often allow quick analysis of the completeness of the rule set, and detection of dead ends or missing conditions. These errors, if not detected early during the systems analysis phase, can often lead to serious behavioral errors in fielded systems, or to expensive correction efforts.

6.12.3 Product Detailed Description

SV-10b is based on the statechart diagram (OMG, 2003). A *state machine* is "a specification that describes all possible behaviors of some dynamic model element. Behavior is modeled as a traversal of a graph of state nodes interconnected by one or more joined transition arcs that are triggered by the dispatching of series of event instances. During this traversal, the state machine executes a series of actions associated with various elements of the state machine." (OMG, 2003)

The product relates states, events, and actions. A state and its associated actions specify the response of a system or system function to events. When an event occurs, the next state may vary depending on the current state (and its associated action), the event, and the rule set or guard conditions.

A change of state is called a transition. Each transition specifies the response based on a specific event and the current state. Actions may be associated with a given state or with the transition between states.

You can use SV-10b to describe the detailed sequencing of system functions described in SV-4. However, the relationship between the actions included in SV-10b and the system functions in SV-4 depends on the purposes of the architecture and the level of abstraction used in the models. The explicit sequencing of system functions in response to external and internal events is not fully expressed in SV-4. You can use SV-10b to reflect explicit sequencing of the system functions, the sequencing of actions internal to a single function, or the sequencing of system functions with respect to a specific system.

Figure 48 provides a template for a simple SV-10b. The black dot and incoming arrow point to initial states (usually one per diagram), while an outgoing arrow pointing to a black dot with a circle around it identifies terminal states. 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 one-way arrows labeled with event/action notation, which indicates an event-action pair, and semantically translates to: when an event occurs, the corresponding action is executed. This notation indicates the event that causes the transition and the ensuing action (if any) associated with the transition.

Figure 48: Systems State Transition Description (SV-10b) – High-Level Template

6.12.4 SV-10b UML Representation

You can produce the SV-10b model in UML using statechart diagrams. Statechart diagrams contain simple states and composite states. They also contain transitions, which are described in terms of triggers or events (generated as a result of an action) and guard conditions associated with the events, and an action or sequence of actions that are executed as a result of the event taking place (see Figure 48). Statechart diagrams specify the reaction of an object to stimuli as a function of its internal state. Guard conditions of a statechart diagram map to the pre- and post-conditions of an SV-4 use case. Events or triggers associated with the transitions on the state diagrams correlate to the triggering events documented in SV-6.

6.12.5 SV-10b Data Element Definitions

The SV-10b model 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. Table 24 describes the architecture data elements for SV-10b.

Table 24: Data Element Definitions for SV-10b

Data Element	Attributes	Explanation
Graphical Box Types		
State*		See OV-6b Definition Table.
Graphical Arrow Types		
Transition*		See OV-6b Definition Table.
Non-Graphical Types		
Statechart Diagram*		See OV-6b Definition Table.
Action*		See OV-6b Definition Table.
Trigger/Event*		See OV-6b Definition Table.
Guard		See OV-6b Definition Table.
Referenced Types		
System		See SV-1/2 Definition Table.
System Function*		See SV-4 Definition Table.

Data Element	Attributes	Explanation
System Data Flow*		See SV-4 Definition Table.
Rule		See OV-6a Definition Table (correlates to a guard).
Relationships		
State Chart has Terminal State*		See OV-6b Definition Table.
State has Associated Action*		See OV-6b Definition Table.
Action is Associated with Transition*		See OV-6b Definition Table.
Event Triggers Transition*		See OV-6b Definition Table.
Transition has Guard		See OV-6b Definition Table.
Transition is Associated with Source		See OV-6b Definition Table.
Transition is Associated with Target		See OV-6b Definition Table.
Guard is Associated with Rule		See OV-6b Definition Table.
Event Maps to System Data Flow Produced or Consumed by System Function*		May be a many-to-many relationship
	Event Name*	Name of event
	System Data Flow Name*	Name of system data flow produced or consumed by system function
Action is Related to System Function*		
	Action Name*	Name/identifier of an action
	System Function Name*	Name/identifier of system function
	Relationship Description*	Text description of the relationship (e.g., action is same as system function, action is contained in system function, action contains system function)

6.12.6 AFDM Support

GJXDM is being proposed for the AFDM. See Section 3.7 for information.

6.13 Systems Event-Trace Description (SV-10c)

6.13.1 Product Definition

The Systems Event-Trace Description (SV-10c) provides a time-ordered examination of the system data elements exchanged between participating external and internal systems, system functions, or human roles as a result of a particular scenario. Each event-trace diagram should have an accompanying description that defines the particular scenario or situation. SV-10c in the Systems View may reflect system-specific aspects or refinements of critical sequences of events described in the Operational View.

6.13.2 Product Purpose

SV-10c products are valuable for moving to the next level of detail from the initial systems design, to help define a sequence of functions and system data interfaces, and to ensure that each participating system, system function, or human role has the necessary information it needs, at the right time, to perform its assigned functionality.

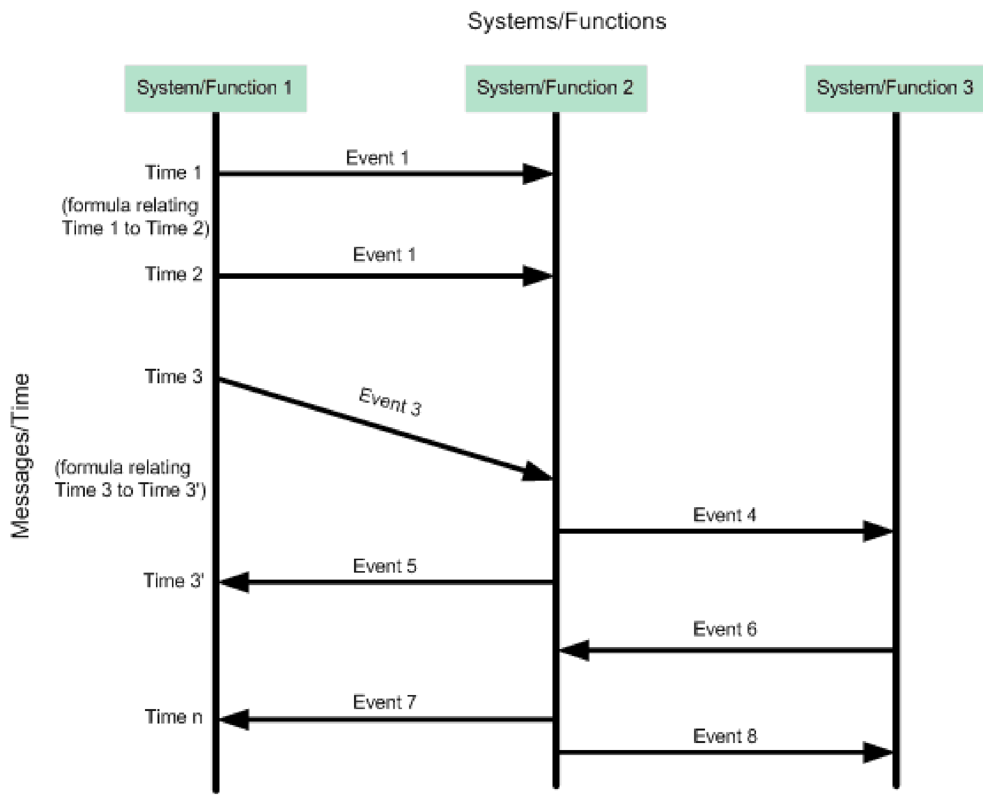
6.13.3 Product Detailed Description

SV-10c allows the tracing of actions in a scenario or critical sequence of events. Time is mapped from the top of the diagram to the bottom. Separate diagrams should depict different scenarios. You can use SV-10c by itself or in conjunction with a SV-10b to describe dynamic behavior of system processes or system function threads.

Figure 49 provides a template for SV-10c. The items across the top of the diagram represent systems, system functions, or human roles that take action based on certain types of events. Each system, function, or human role has a lifeline associated with it that runs vertically. Specific points in time can be labeled on the lifelines, running down the left-hand side of the diagram. An event may occur as a result of an action. An event in a sequence diagram implies the action that produced it.

Labels indicating timing constraints or providing event descriptions can be shown in the margin or near the transitions of the events that they label. One-way arrows between the lifelines represent events, and the points at which they intersect the lifelines represent the times at which the system, function, or role becomes aware of the events. The direction of the events represents the flow of control from one system, function, or role to another based on the event. Each diagram may represent external and internal systems or system functions, but not both in the same diagram. You can also use human roles in the diagram along with either systems or system functions to describe the human interfaces to the systems or system functions.

Figure 49: SV-10c – Template



6.13.4 SV-10c UML Representation

You can use UML sequence diagrams to model SV-10c. Each diagram may represent external and internal systems or system functions, but not both in the same diagram. Figure 49 provides a template for a UML sequence diagram.

6.13.5 SV-10c Data Element Definitions

Table 25 describes the architecture data elements for SV-10c model.

Table 25: Data Element Definitions for SV-10c

Data Element	Attributes	Explanation
Graphical Box Types		
Lifeline*		
	Lifeline Name*	Name of the lifeline representing a system, external system, system function, or human role

Data Element	Attributes	Explanation
	Description*	Text description of any assumptions or scope constraints on the system, external system, system function, or human role associated with the lifeline
Graphical Arrow Types		
Event*		See OV-6c Definition Table.
Non-Graphical Types		
Event Time*		See OV-6c Definition Table.
Referenced Types		
Human Role*		See OV-4 Definition Table.
System*		See SV-1 Definition Table.
System Function		See SV-4 Definition Table.
Action*		See OV-6b Definition Table.
Guard		See OV-6b Definition Table.
Relationships		
Lifeline Represents a Human Role, System, or System Function*		
	Lifeline Name*	Name of the lifeline
	Human Role, System, or System Function Name*	Name of the human role, system, or system function represented by the lifeline
Event Starts at Time*		See OV-6c Definition Table.
Event Ends at Time*		See OV-6c Definition Table.
Event is Associated with Action*		See OV-6b Definition Table.
Action is Related to System/Function*		
	Action Name*	Name/identifier of an action
	System/Function Name*	Name/identifier of a system/function
	Relationship Description*	Text description of the relationship
Event Maps to System Data Produced or Consumed by		

Data Element	Attributes	Explanation
System/Function*		
	Event Name	Name of event
	System Data Produced or Consumed by System/Function Name	Name of system data produced or consumed by system/function
Event is Associated with System Data Exchange*		
	Event Name*	Label of the event
	System Data Exchange Identifier*	Identifier of a system data exchange associated with the event
Event Originates from Lifeline*		See OV-6c Definition Table.
Event Terminates at Lifeline*		See OV-6c Definition Table.

6.13.6 AFDM Support

GJXDM is being proposed for the AFDM. See Section 3.7 for information.

6.14 Physical Schema (SV-11)

6.14.1 Product Definition

The Physical Schema product (SV-11) is one of the architecture products closest to actual system design in the PSAF. The product defines the structure of the various kinds of system data used by the systems in the architecture.

6.14.2 Product Purpose

The product serves several purposes. The product provides as much detail as possible on the system data elements exchanged between systems, thus reducing the risk of interoperability errors. Another purpose is that the product provides system data structures for use in the system design process, if necessary.

6.14.3 Product Detailed Description

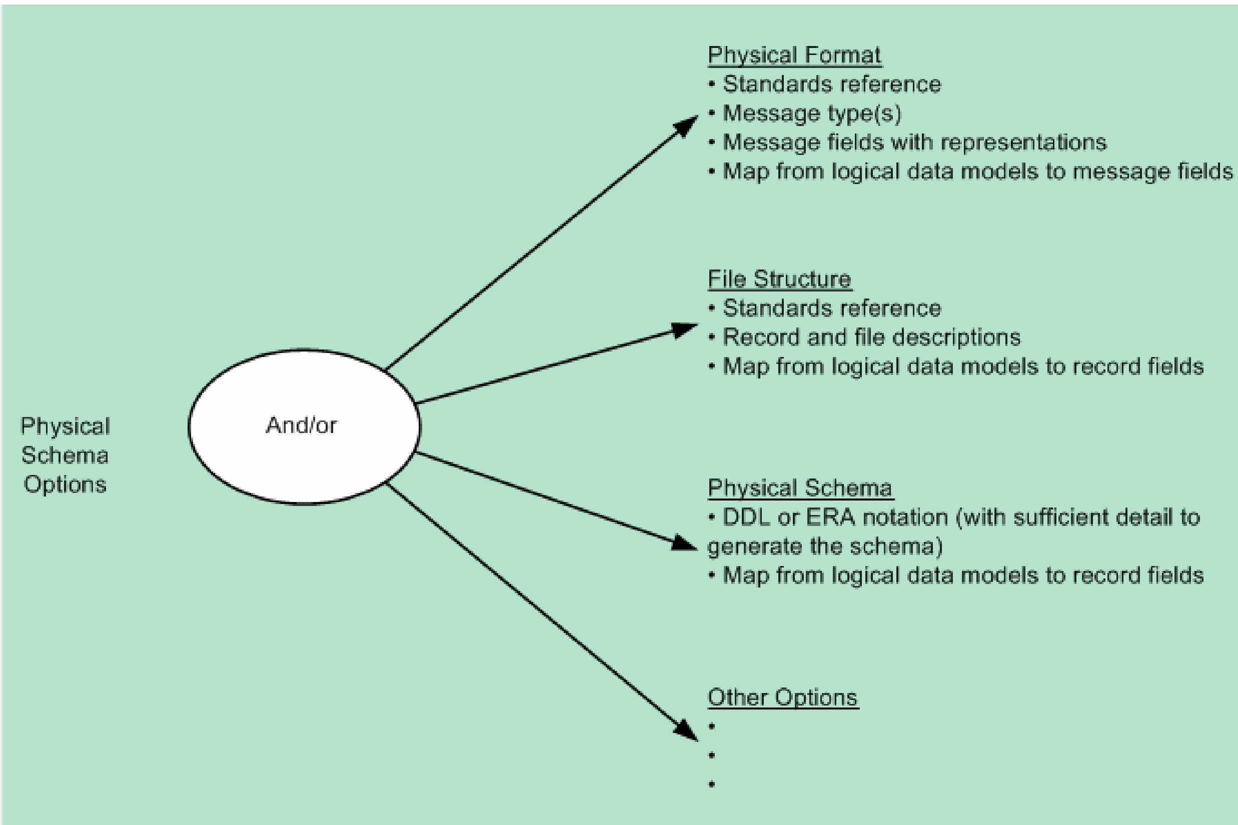
SV-11 is an implementation-oriented data model you use in the Systems View to describe how the information requirements represented in OV-7 are actually implemented. Entities represent one of more of the following:

- System data flows in SV-4

- System data elements specified in SV-6
- Triggering events in SV-10b
- Events in SV-10c

If you use both models, a given OV-7 should be mapped to SV-11. The form of SV-11 can vary greatly, as shown in Figure 50.

Figure 50: SV-11 – Representation Options



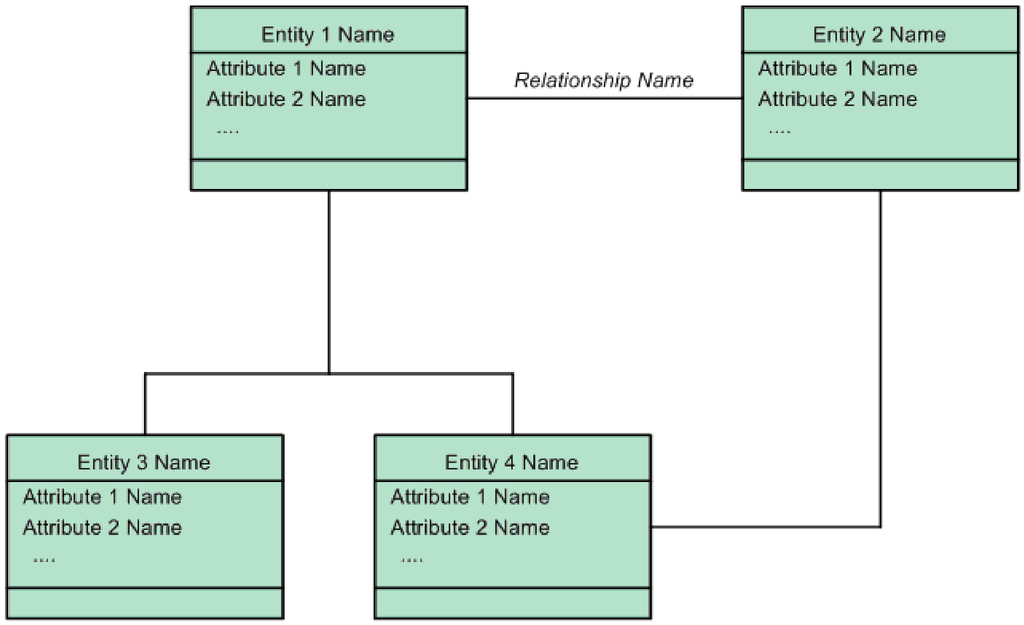
For some purposes, an entity relationship style diagram of the physical database design will suffice. References to message format standards, which identify message types and options to be used, may suffice for message-oriented implementations. Descriptions of file formats may be used when file passing is used to exchange information. A Data Definition Language (DDL) (for example, Structured Query Language (SQL)) may also be used where shared databases integrate the systems. Interoperating systems may employ a variety of techniques to exchange system data; they may have several distinct partitions in their SV-11, with each partition using a different form. Standards associated with entities are also documented in this product.

6.14.4 UML Representation

You can specify SV-11 in UML using a class diagram. Class diagrams offer all the UML elements needed to produce entity-relationship diagrams.

Class diagrams consist of classes, interfaces, and collaborations, as well as dependency, generalization, association, and realization relationships. The attributes of these classes can be expanded to include associations and cardinality (Booch, 1999). The class diagram is the closest parallel to the entity-relationship diagram. Figure 51 is a template of such a use of a UML class diagram.

Figure 51: UML Class Diagram for SV-11



Object classes defined should trace to one of the following:

- System data flows in SV-4
- System data elements specified in SV-6
- Transition triggers in SV-10b
- Events in SV-10c

The SV-11 class diagram is at a lower level of detail than the class diagrams built to support the other SV products. However, the other products cannot be fully defined without reference to this more detailed level.

6.14.5 SV-11 Data Element Definitions

Table 26 describes the architecture data elements for SV-11.

Table 26: Data Element Definitions for SV-11

Data Element	Attributes	Explanation
Non-Graphical Types		

Data Element	Attributes	Explanation
Physical Schema Model*		
	Name*	Name/identifier of physical schema 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
	Number of Constituent Models*	Number of other types of models that make up the physical schema model
System Data/Message Model		
	System Data/Message Standard Name	Name/identifier of system data or messaging standard to be used
	System Data/Message Format Name	Name/identifier of system data or message format used within the message standard
	System Data/Message Type Parameters/Options	Parameter and option values necessary to completely identify system data or message format to be used
File Structure Model		
	File Name	Name/identifier of file used to hold architecture domain system data
	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 structures within the file
Entity Relationship Diagram (ERD)		
	ERD Name	Name/identifier of specific entity-relationship model
	ERD Type	Name of specific form of notation used; may be tool dependent (e.g., system architect)
	Softcopy Reference	Location and file format for softcopy of the specific model (e.g., URL)
Data Definition Language (DDL) Model		
	DDL Model Name	Name/identifier of DDL model

Data Element	Attributes	Explanation
	DDL Name	Name of DDL in which the DDL model is written (e.g., SQL)
	Softcopy Reference	Location and file format for the softcopy of the DDL model (e.g., URL)
Object-Oriented Class Model		
	Diagram Name	Name/identifier of specific OO class model
	Softcopy Reference	Location and file format for softcopy of the specific model (e.g., URL)
Referenced Types		
Standard		See TV-1/2 Definition Table.
Relationships		
Physical Schema Model Contains System Data/Message Model, File Structured Model, ERD Model, DDL Model, or OO Class Model*		
	Physical Schema Model Name*	Name/Identifier of physical schema model
	System Data/Message Model, File Structured Model, ERD Model, DDL Model, or OO Class Mode Name*	Name/Identifier of one of the models that makes up the physical schema model
Logical Model Maps to Physical Schema Model*		
	Logical Model Name*	Name/Identifier of logical data model (see OV-7)
	Physical Schema Model Name*	Name/Identifier of corresponding physical schema model
	Reference to Mapping Document*	Location of hardcopy or softcopy of document containing the detailed mapping between the logical and physical schema models; there is no generic form for this mapping — it can be complex and varies based on the types of physical models used
Physical Schema Model Uses Modeling Standard		
	Physical Schema Model Name	Name/identifier of physical schema model
	Model Standard Name	Name of data modeling standard from TV-1

6.14.6 AFDM Support

GJXDM is being proposed for the AFDM. See Section 3.7 for information.

7 Technical Standards View Products

The TV products provide the technical systems-implementation standards upon which engineering specifications are based, common building blocks are established, and product lines are developed.

The TV includes two products:

- Technical Standards Profile (TV-1)
- Technical Standards Forecast (TV-2)

7.1 Technical Standards Profile (TV-1)

7.1.1 Product Definition

The Technical Standards Profile (TV-1) collects the various systems standards rules that implement and sometimes constrain the choices that can be made in the design and implementation of an architecture.

7.1.2 Product Purpose

Primarily, this product delineates systems standards rules and conventions that apply to architecture implementations. When the standards profile is tied to applicable system elements, TV-1 serves as the bridge between the SV and TV.

7.1.3 Product Detailed Description

TV-1 consists of the set of systems standards rules that govern systems implementation and operation of that architecture. The technical standards generally govern what hardware and software may be implemented and what system data formats may be used. This means the profile delineates which standards may be used to implement the systems, system hardware or software items, communications protocols, and system data formats.

TV-1 is constructed as part of a given architecture and in accordance with the architecture purpose. Typically, this will involve starting with one or more overarching reference models or standards profiles, such as OMB's Technical Reference Model (TRM) (OMB, 2003). Using these reference models or standards profiles, you should select the service areas relevant to the architecture. The identification of relevant services within service areas subsequently points to agreed-upon standards. Cite the source document used for identifying each standard.

The standards are referenced as relationships to the systems, system functions, system data, hardware/software items or communication protocols in SV-1, SV-2, SV-4, SV-6, OV-7, and SV-11 products, where applicable. That is, you should associate each standard listed in the profile with the SV elements that implement or use the standard (e.g., SV-1, SV-2, SV-4, SV-6, OV-7 and SV-11 element standards, where applicable). Standards for OV-7 and SV-11 do not include system data standards such as naming conventions, attribute lists, and field types, but refer to the source for the data entities or the data modeling standard used (e.g., IDEF1X).

Table 27 shows a template for TV-1. The template contains a subset of services by way of example.

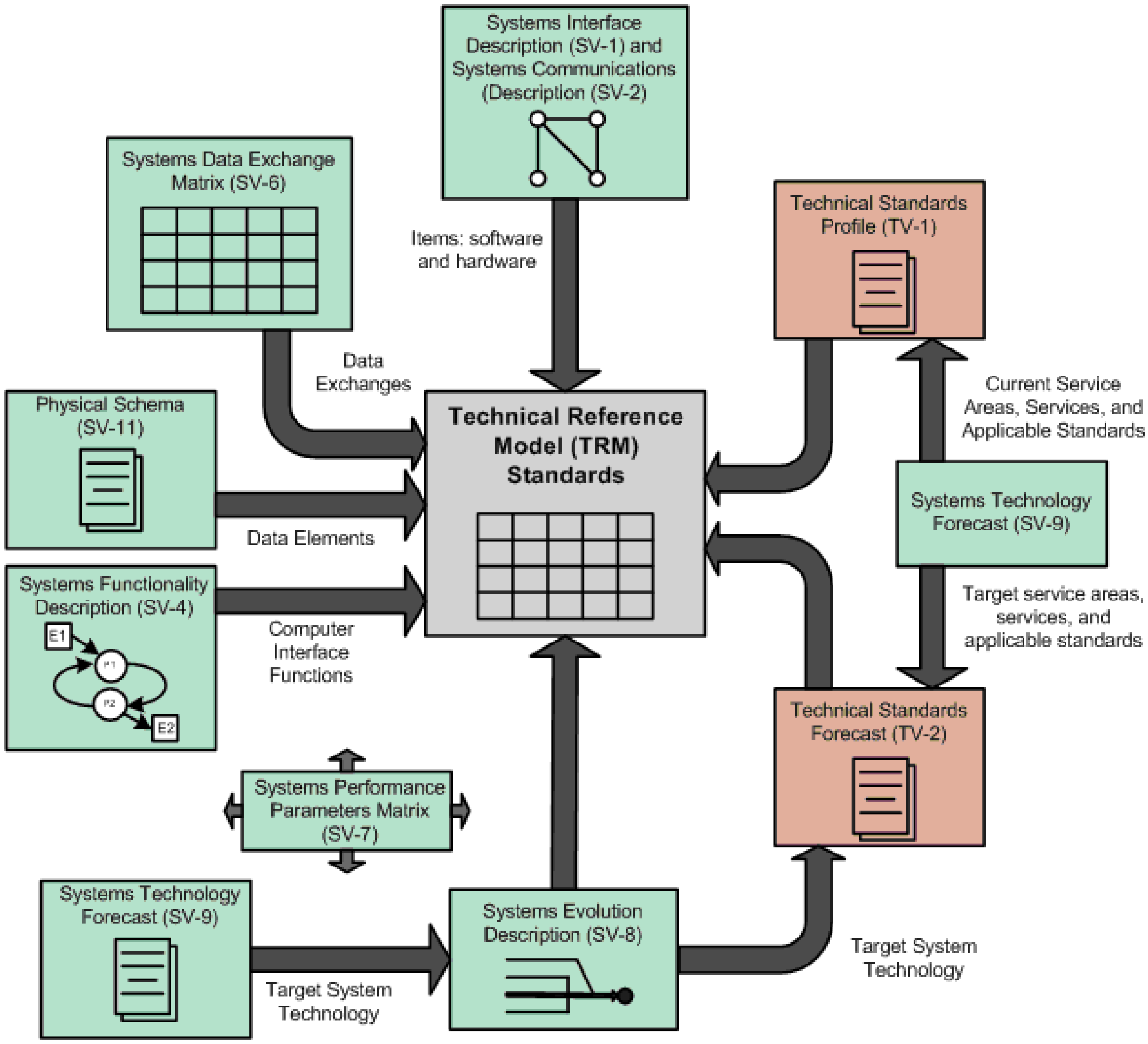
Table 27: TV-1 Template

Service Area	Subset of Service	Standard and Source Document
Information-Processing Standards	Higher Order Languages	
	Software Life Cycle Process	
	Geospatial Data Interchange	
	Motion Imagery Data Interchange — Video	
	Distributed Object Computing	
Information-Transfer Standards	Data Flow Network	
	Command and Control Information (C2I) Network	
	Physical Layer	
	Network Layer	
	Layer Management	
	File Transfer Standards	
	Remote Terminal Standards	
	Network Time Synchronization Standards	
	Web Services Standards	
	Connectionless Data Transfer	
	Transport Services Standards	
Information Modeling, Metadata, and Information Exchange Standards	Activity Modeling	
	Data Modeling	
	Object-Oriented Modeling	
Human Computer Interface	Mandates	
Information Security / Information Infrastructure Standards	Password Security	
	Application Software Entity Security	

Service Area	Subset of Service	Standard and Source Document
	Standards	
	Virtual Private Network Service	
	Intrusion Detection Service	
	Human Computer Interface Security Standards	

Timed technology forecasts from SV-9 may be related to TV-1 standards. When a certain technology becomes available, it may force a need to upgrade to a new version of a TV-1 standard. Similarly, a standard listed in TV-2 may not be adopted until a certain technology becomes available. This is how standards in TV-1, which are applicable to systems elements from SV-1, SV-2, SV-4, SV-6, and OV-7, may be related to future standards listed in TV-2, through the SV-9 product. Figure 52 illustrates the bridge concept.

Figure 52: Systems Products Associated with Standards



Once you develop the standards profile, you can organize another TV-1 product as a matrix that delineates the standards identified in the profile that apply to the relevant system elements. An example bridge matrix template appears in Table 28.

Table 28: TV-1 Template with Corresponding System Elements

			SV-1 and SV-2 Systems (Includes SOS, FOS, subsystem, communications systems)	SV-1 and SV-2 Hardware or Software Item	SV-2 Communications (Physical) Link	SV-4 System Function	SV-6 System Data Element	OV-7, SV-11 Model Standard or Source
Service Area	**Service**	**Standard**	**Applicable System ID or Name (with Time Period if applicable)**	**Hardware or Software ID or Name, Version, (with Time Period if applicable)**	**System Data Link ID or Name (with Time Period if applicable)**	**System Function ID or Name**	**System Data Exchange ID or Name (with Time Period if applicable)**	**Model Standard or Source ID or Name (with Time Period if applicable)**
Information Technology Standards	Operating Environment							
Information Processing Standards	Higher Order Languages							
	Geospatial Data Interchange							
Information Transfer Standards	Data Flow Network							
	Physical Layer							
	Network Interface							
	Narrow-Band Video Conferencing							
Information Modeling, Metadata, and Information Exchange Standards	Object-Oriented Modeling							
Human Computer Interface	Mandates							
Information Security / Information Infrastructure Standards	PKI Certificate Profile Standards							
	Human-Computer Interface Security Standards							
	Web Security Standards							
Physical Services Standards	Chassis							
	Backplanes							
	Circuit Cards							

As Table 29 shows, a separate matrix for each product or product element type may be developed showing a list of the same element type (e.g., system) as columns, with time periods specified in the cells, specifying when to apply the standards, where applicable. If time periods are used, then TV-1 also includes a bridge to the systems and their time periods in SV-8.

Table 29: TV-1 Template for Systems with Corresponding Time Periods

Standards Applicable to SV-1 Systems			System A	System B	System C
Service Area	Service Area	Standard			
Information Technology Standards	Operating Environment	The DII COE as mandated by the service	Current Baseline: Jan. 12, 2003		
	Operating System Standard	ISO/IEC 9945-11996, Information Technology – Portable Operating System Interface (POSIX) 0 Part 1: System Application Program Interface (API) [C language] (Mandated Services).	6 months from Baseline		
	Operating System Standard	ISO/IEC 9945-1:1996: (Thread Extensions) to ISO/IEC 9945-1:1996, Information Technology – Portable Operating System Interface (POSIX) – Part 1: System Application Program Interface (API) [C language] (Thread Optional Services).	12 months from Baseline		
	Operating System Standard	IEEE 1003.3d:1994, POSIX – Part 2: Shell and Utilities – Amendment Batch Environment.			
	Operating System Standard	Win32 APIs, Window Management and Graphics Device Interface, Volume 1 Microsoft Win32 Programmers Reference Manual, 1993 or later, Microsoft Press.			
	Operating System Standard	Win32 APIs, as specified in the Microsoft Platform Software Development Kit (SDK).			

7.1.4 UML Representation

There is no equivalent to this product in UML.

7.1.5 TV-1 Data Element Definitions

Table 30 describes the architecture data elements for TV-1.

Table 30: Data Element Definitions for TV-1

Data Element	Attributes	Explanation
Non-Graphical Types		
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

Data Element	Attributes	Explanation
		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 effects 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, future (i.e., there are current standards for this service or interface to the service; or there are expected to be some standards in the future).
Standard*		
	Standard Name*	Name and identifier for standard, including maintaining organization and relevant revision dates
	Type*	Description of the type of standard (e.g. industry or de facto, organizational or project-specific)
	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
Referenced Types		

Data Element	Attributes	Explanation
System		See SV-1 Definition Table.
System Hardware/Software Item		See SV-1 Definition Table.
Communications System		See SV-2 Definition Table.
Communications Link		See SV-2 Definition Table.
Communications Network		See SV-2 Definition Table.
System Data Element		See SV-6 Definition Table.
System Function		See SV-4 Definition Table.
Time Period		See SV-8 Definition Table.
Timed Technology Forecast		See SV-9 Definition Table.
Logical Data Model		See SV-7 Definition Table.
Physical Schema Model		See SV-11 Definition Table.
Relationships		
Standards Profile is Refinement of Parent Standards Profile		
	Parent Standards Profile Name	Name/identifier of a parent standards profile
	Child Standards Profile Name	Name/identifier of a child standards profile, which is a refinement of the other profile (e.g., 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

Data Element	Attributes	Explanation
Service Area Includes Service*		
	Service Area Name*	Name/identifier of a service area or service
	Service Name*	Name/identifier of a service included in that service area
Standards Profile Includes Service Area*		
	Standards Profile Name*	Name/identifier of a standards profile
	Service Area 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 standards profile
Standard Applies to System or Subsystem at a Time Period		
	Standard Name	Name/identifier of a standard
	System Name	Name/identifier of a system
	Time Period Name	Name/identifier of time period
Standard Applies to System Hardware/Software Item at a Time Period		
	Standard Name	Name/identifier of a standard
	System Hardware/Software Item Name	Name/identifier of a system hardware/software item
	Time Period Name	Name/identifier of time period
Standard Applies to Communications System at a Time Period		
	Standard Name	Name/identifier of a standard
	Communications System Name	Name/identifier of a communications system
	Time Period Name	Name/identifier of time period

Data Element	Attributes	Explanation
Standard Applies to Communications Link at a Time Period		
	Standard Name	Name/identifier of a standard
	Communications Link Name	Name/identifier of a communications link
	Time Period Name	Name/identifier of time period
Standard Applies to Communications Network at a Time Period		
	Standard Name	Name/identifier of a standard
	Communications Network Name	Name/identifier of a communications network
Standard Applies to System Data Element at a Time Period		
	Standard Name	Name/identifier of a standard
	System Data Element Name	Name/identifier of a system data element (see SV-6 Definition Table)
Standard Applies to System Function at a Time Period		
	Standard Name	Name/identifier of a standard
	System Function Name	Name/identifier of a system function (e.g., HCI, HSI, or GUI system function)
	Time Period Name	Name/identifier of time period
Standard Applies to Logical Data Model at a Time Period		
	Standard Name	Name/identifier of a standard
	Logical Data Model Name	Name/identifier of a logical data model
	Time Period Name	Name/identifier of time period
Standard Applies to Systems Physical Schema Model at a Time Period		
	Standard Name	Name/identifier of a standard
	Physical Schema Model Name	Name/identifier of a physical schema model

Data Element	Attributes	Explanation
	Time Period Name	Name/identifier of time period
Timed Technology Forecast Will Retire Standard		
	Standard Name	Name/identifier of a standard
	Timed Technology Forecast Name	Name/identifier of a timed technology forecast

7.1.6 AFDM Support

GJXDM is being proposed for the AFDM. See Section 3.7 for information.

7.2 Technical Standards Forecast (TV-2)

7.2.1 Product Definition

The Technical Standards Forecast (TV-2) contains expected changes in technology-related standards and conventions, which are documented in the TV-1 product. You correlate the forecast for evolutionary changes in the standards against the time periods mentioned in the SV-8 and SV-9 products.

7.2.2 Product Purpose

A prime purpose of this product is to identify critical technology standards, their fragility, and the effect of these standards on the future development and maintainability of the architecture and its constituent elements.

7.2.3 Product Detailed Description

TV-2 lists emerging or evolving technology standards relevant to the systems covered by the architecture. It contains predictions about the availability of emerging standards, and relates these predictions to the Systems View elements and the time periods that are listed in the SV-8 and SV-9.

You coordinate the specific time periods selected (e.g., 6-month, 12-month, 18-month intervals) and the standards you are tracking with architecture transition plans (see SV-8). That is, insertion of new technological capabilities and upgrading of existing systems may depend on, or be driven by, the availability of new standards and products incorporating those standards. The forecast specifies potential standards, and thus affects current architectures and influences the development of transition and objective, or target, architectures. Tailor the forecast to focus on standards areas that relate to the purpose of the architecture you are describing, and identify potential standards that will affect the architecture. If interface standards are an integral part of the technologies important to the evolution of a given architecture, then it may be convenient to combine TV-2 with SV-9. For other projects, it may be convenient to combine all the standards information into one document, combining TV-2 with TV-1.

TV-2 delineates the standards that will potentially affect the relevant system elements (from SV-1, SV-2, SV-4, SV-6, and OV-7) and relates them to the time periods that are listed in the SV-8 and SV-9. You can tie a system's evolution, specified in SV-8, to a future standard listed in TV-2. A timed technology forecast from SV-9 is related to a TV-2 standards forecast in the following manner: a certain technology may be dependent on a TV-2 standard. This means a standard listed in TV-2 may not be adopted until a certain technology becomes available. This is how a prediction on the adoption of a future standard, as applicable to systems elements from SV-1, SV-2, SV-4, SV-6, and OV-7, may be related to standards listed in TV-1 through the SV-9. A template for TV-2 is not shown in this section. You can use the same template (see Table 28) shown in TV-1 to describe TV-2.

7.2.4 UML Representation

There is no equivalent to this product in UML.

7.2.5 TV-2 Data Element Definitions

Table 31 describes the architecture data elements for TV-2.

Table 31: Data Element Definitions for TV-2

Data Element	Attributes	Explanation
Non-Graphical Types		
Standards Forecast Profile*		
	Name*	Name/identifier of standards forecast profile
	Description*	Textual description of purpose of forecast
Timed Standards Forecast*		
	Name*	Name/identifier of a forecast regarding a specific standard
	Standard*	Textual description of a specific standard being forecasted
	Time period*	Time period for which forecast is valid; usually expressed in terms of a future date or months from baseline
	Standard Status*	Expected status based on forecast; for example, one or more of: approved, draft available, updated, replaced, obsolete, COTS implementations available
	Discussion*	Textual notes regarding standard status, likely commercial market acceptance of standard (i.e., likelihood of COT implementations), issues and risks of adopting a new/updated

Data Element	Attributes	Explanation
		standard
Referenced Types		
Reference Model		See TV-1/2 Definition Table.
Service Area		See TV-1 Definition Table.
Service		See TV-1 Definition Table.
System		See TV-1 Definition Table.
System Hardware/Software Item		See TV-1 Definition Table.
Communications System		See SV-2 Definition Table.
Communications Link		See SV-2 Definition Table.
Communications Network		See SV-2 Definition Table.
System Data Element		See SV-6 Definition Table.
Time Period		See SV-8 Definition Table.
Timed Technology Forecast		See SV-9 Definition Table.
Relationships		
Standards Forecast Profile is Based on Reference Model		
	Standards Forecast Profile Name	Name/identifier of standards forecast profile
	Reference Model Name	Name/identifier of reference model used to organize the standards in the forecast
Standards Forecast Profile is Based on Parent Standards Profile		
	Standards Forecast Profile Name	Name/identifier of standards forecast profile
	Parent Standards Profile Name	Name/identifier of parent standards profile used to organize the standards in the forecast
Standards Forecast Profile Covers Service Area/Service		
	Standards Forecast Profile Name	Name/identifier of standards forecast profile
	Service Area Name	Name/identifier of a service area or service covered by the standards forecast profile

Data Element	Attributes	Explanation
Service Area/Service has Timed Standards Forecast		
	Service Area/Service Name	Name/identifier of a service area or service
	Timed Standards Forecast Name	Name/identifier of a specific, time sensitive forecast for a standard relevant to the service area or service
Timed Standards Forecast Impacts System		
	Timed Standards Forecast Name	Name/identifier of a timed standards forecast
	System Name	Name/identifier of a system that could be impacted by the timed standards forecast
Timed Standards Forecast Affects System		
	Timed Standards Forecast Name	Name/identifier of a timed standards forecast
	System Name	Name/identifier of a system that could be affected by the timed standards forecast
Timed Standards Forecast Affects System Hardware/Software Item		
	Timed Standards Forecast Name	Name/identifier of a timed standards forecast
	System Hardware/Software Item Name	Name/identifier of a system hardware/software item that could be affected by the timed standards forecast
Timed Standards Forecast Affects Communications System		
	Timed Standards Forecast Name	Name/identifier of a timed standards forecast
	Communications System Name	Name/identifier of a communications system (for example, gateway, router, satellite) that could be affected by the timed standards forecast
Timed Standards Forecast Affects Communications Link		
	Timed Standards Forecast Name	Name/identifier of a timed standards forecast
	Communications Network Name	Name/identifier of a communications network that could be affected by the timed standards forecast

Data Element	Attributes	Explanation
Timed Standards Forecast Affects System Data Element		
	Timed Standards Forecast Name	Name/identifier of a timed standards forecast
	System Data Element Name	Name/identifier of a system data element that could be affected by the timed standards forecast
Timed Standards Forecast Corresponds to a Time Period		
	Timed Standards Forecast Name	Name/identifier of a timed standards forecast
	Time Period Name	Name/identifier of a time period
Timed Standards Forecast is Dependent on Timed Technology Forecast		
	Timed Standards Forecast Name	Name/identifier of a timed standards forecast
	Timed Technology Forecast Name	Name/identifier of timed technology forecast on which the timed standards forecast is dependent

7.2.6 AFDM Support

GJXDM is being proposed for the AFDM. See Section 3.7 for information.

Appendix A: Glossary of Acronyms

Note: The following acronyms are derived from several sources, including NIMS, NRP, UML, and public safety PSAF and PS SoR documents. The next release of this document will identify and use these acronyms consistently.

A

A&I
Architecture and Interoperability

ACL
Access Control List

AFDM
Architecture Framework Data Model

AFWG
Architecture Framework Working Group

API
Application Program Interface

ARS
Architecture Repository System

ATM
Asynchronous Transfer Mode

AV
All-Views

AV-1
Overview and Summary Information

AV-2
Integrated Dictionary

B

BRM
Business Reference Model

C

COTS
Commercial Off-The-Shelf

C4FM
Compatible Four-Level Frequency Modulation

D

DB
Database

DBMS
Database Management System

DDL
Data Definition Language

DFD
Data Flow Diagram

DoDAF
DoD Architecture Framework

DRM
Data Reference Model

E

EAN
Extended Area Network

EM
Emergency Management

EMS
Emergency Medical Services

EPA
Environmental Protection Agency

ERD
Entity Relationship Diagram

F

FBI
Federal Bureau of Investigation

FEA
Federal Enterprise Architecture

FEMA
Federal Emergency Management Agency

FIPS
Federal Information Processing Standard

FoS
Family of Systems

FRP
Federal Response Plan

G

GUI
Graphical User Interface

GJXDM
Global Justice XML Data Model

H

HCI
Human-Computer Interface.

HR
Human Resources

HTML
Hypertext Markup Language

I

IAN
Incident Area Network

IC
Incident Commander or Incident Command

ICD
Interface Control Document

ICP
Incident Command Post

ICS
Incident Command System

IDEF
Integration Definition Methods

IDEF0
IDEF Function Modeling Method

IDEF1X
IDEF Data Modeling Method

IDEF3
IDEF Process Description Capture Method

IFC
Incident Fire Commander

ILEC
Incident Law Enforcement Commander

IMBE
Improved multiband excitation

I/O
Input and Output

IP
Internet Protocol

IT
Information Technology

J

JAN
Jurisdiction Area Network

K

KI
Key Interface

KIP
Key Interface Profile

L

LAN
Local Area Network

LEC
Law Enforcement Commander

LMR
Land Mobile Radio

LPC
Linear predictive coding

M

M&S
Modeling and Simulation

N

NHTSA
National Highway Traffic Safety Administration

NIMS
National Incident Management System

NRP
National Response Plan

O

OCL
Object Constraint Language

OMB
Office of Management and Budget

OMG
Object Management Group

OO
Object-Oriented

OTAR
Over-The-Air Rekeying

OV
Operational View

OV-1
High-Level Operational Concept Graphic

OV-2
Operational Node Connectivity Description

OV-3
Operational Information Exchange Matrix

OV-4
Organizational Relationships Chart

OV-5
Operational Activity Model

OV-6
Operational Activity Sequence and Timing Descriptions

OV-6a
Operational Rules Model

OV-6b
Operational State Transition Description

OV-6c
Operational Event-Trace Description

OV-7
Logical Data Model

P

PAN
Personal Area Network

POC
Point of Contact

PDA
Personal Digital Assistant

PRM
Performance Reference Model

PS
Public Safety

PSAF
Public Safety Architecture Framework

P25
Project 25

R

R&D
Research and Development

ROI
Return On Investment

S

SA
Structured Analysis

SF
System Function

SoS
System of Systems

SRM
Service Component Reference Model

SV
Systems View

SV-1
Systems Interface Description

SV-2
Systems Communications Description

SV-3
Systems-Systems Matrix

SV-4
Systems Functionality Description

SV-5
Operational Activity to Systems Functionality Traceability Matrix

SV-6
Systems Data Exchange Matrix

SV-7
Systems Performance Parameters Matrix

SV-8
Systems Evolution Description

SV-9
Systems Technology Forecasts

SV-10
Systems Functionality and Timing Descriptions

SV-10a
Systems Rules Model

SV-10b
Systems State Transition Description

SV-10c
Systems Event-Trace Description

SV-11
Physical Schema

T

TCP
Transmission Control Protocol

TDMA
Time division multiple access

TRM
Technical Reference Model

TV
Technical Standards View

TV-1
Technical Standards Profile

TV-2
Technical Standards Forecast

U

UC
Unified Command

UML
Unified Modeling Language

UOB
Unit of Behavior

URR
Universal Reference Resources

V

VPN
Virtual Private Network

W

WAN
Wide Area Network

X

XML
Extensible Markup Language

Appendix B: Dictionary of Terms

A

Architecture Data Element

One of the data elements that make up the PSAF products. Also referred to as architecture data type. (PSAF)

C

Command

The act of directing, ordering, or controlling by virtue of explicit statutory, regulatory, or delegated authority (NIMS)

Communications Medium

A means of data transmission

D

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 Model

A representation of the data elements pertinent to an architecture, often including relationships among the elements and their attributes or characteristics. (PSAF)

Data-Entity

The representation of a set of people, objects, places, events, or ideas that share the same characteristic relationships.

F

Family of Systems

A set or arrangement of independent systems that can be arranged or interconnected in various ways to provide different capabilities.

Format

The arrangement, order, or layout of data. (Derived from IEEE 610.5)

Functional Area

A major area of related activity.

I

Incident

An occurrence or event, natural or human-caused, that requires an emergency response to protect life or property. Incident can, for example, include major disasters, emergencies, terrorist attacks, terrorist threats, wildland and urban fires, floods, hazardous materials spills, nuclear accidents, aircraft accidents, earthquakes, hurricanes, tornadoes, tropical storms, war-related disasters, public health and medical emergencies, and other occurrences requiring an emergency response. (NIMS)

Incident Command Post

The field location at which the primary tactical-level, on-scene incident command functions are performed. The ICP may be collocated with the incident base or other incident facilities and is normally identified by a green rotating or flashing light. (NIMS)

Incident Command System

A standardized on-scene emergency management construct specifically designed to provide for the adoption of an integrated organizational structure that reflects the complexity and demands of single or multiple incidents, without being hindered by jurisdictional boundaries. ICS is the combination of facilities, equipment, personnel, procedures, and communications operating within a common organizational structure, designed to aid in the management of resources during incidents. It is used for all kinds of emergencies and is applicable to small as well as large and complex incidents. ICS is used by various jurisdictions and functional agencies, both public and private, to organize field-level incident management operations. (NIMS)

Incident Commander

The individual responsible for all incident activities, including the development of strategies and tactics and the ordering and the release of resources. The IC has overall authority and responsibility for conducting incident operations and is responsible for the management of all incident operations at the incident site. (NIMS)

Independent Systems

A communications system that serves some communications service requirements for a public safety agency but does not or cannot provide standardized interfaces to other agencies' communication systems.

Information

The refinement of data through known conventions and context for purposes of imparting knowledge.

Information Element

Information that is passed from one operational node to another. Associated with an information element are such performance attributes as timeliness, quality, and quantity values. (PSAF)

Information Exchange

The collection of information elements and their performance attributes such as timeliness, quality, and quantity values. (PSAF)

Information Exchange Requirement

A requirement for information that is exchanged between nodes.

Information Technology

Any equipment, or interconnected system or subsystem of equipment, that is used in the automatic acquisition, storage, manipulation, management, movement, control, display, switching, interchange, transmission, or reception of data or information by the public safety agency.

Integrated Architecture

An architecture description that has integrated Operational, Systems, and Technical Standards Views with common points of reference linking the Operational View and the Systems View and also linking the Systems View and the Technical Standards View. An architecture description is an *integrated architecture* when products and their constituent architecture data elements are developed such that architecture data elements defined in one view are the same (i.e., same names, definitions, and values) as architecture data elements referenced in another view.

Interoperability (Communications)

Communications interoperability is the ability of public safety agencies to talk across disciplines and jurisdictions via radio communications systems, exchanging voice and data with one another on demand, in real time, when needed. (PS SoR)

J

Jurisdiction

A range or sphere of authority. Public agencies have jurisdiction at an incident related to their legal responsibilities and authority. Jurisdictional authority at an incident can be political or geographical (e.g., city, county, tribal, State, or Federal boundary lines) or functional (e.g., law enforcement, public health). (NIMS)

L

Liaison

A form of communication for establishing and maintaining mutual understanding and cooperation. (NIMS)

Liaison Officer

A member of the Command Staff responsible for coordinating with representatives from cooperating and assisting agencies. (NIMS)

Link

A representation of the physical realization of connectivity between systems nodes.

Local Government

A county, municipality, city, town, township, local public authority, school district, special district, intrastate district, council of governments (regardless of whether the council of governments is incorporated as a nonprofit corporation under State law), regional or interstate government entity, or agency or instrumentality of a local government; an Indian tribe or authorized tribal organization, or in Alaska a Native village or Alaska Regional Native Corporation; a rural community, unincorporated town or village, or other public entity. See Section 2 (10), Homeland Security Act of 2002, Pub. L. 107-296, 116 Stat. 2135 (2002). (NIMS)

M

Mission Area

The general class to which an operational mission belongs.

Note: Within a class, the missions have common objectives.

Mission

An objective together with the purpose of the intended action.

Note: Multiple tasks accomplish a mission.

Multi-Jurisdictional Incident

An incident requiring action from multiple agencies that each have jurisdiction to manage certain aspects of an incident. In the Incident Command System (ICS), these incidents will be managed under Unified Command. (NIMS)

N

National Incident Management System (NIMS)

A system mandated by HSPD-5 that provides a consistent nationwide approach for Federal, State, local, and tribal governments; the private-sector, and nongovernmental organizations to work effectively and efficiently together to prepare for, respond to, and recover from domestic incidents, regardless of cause, size, or complexity. To provide for interoperability and compatibility among local, state, tribal and Federal capabilities, the NIMS includes a core set of concepts, principles, and terminology. HSPD-5 identifies these as the ICS; multi-agency coordination systems; training; identification and management of resources (including systems for classifying types of resources); qualification and certification; and the collection, tracking, and reporting of incident information and incident resources. (NIMS)

National Response Plan (NRP)

A plan mandated by HSPD-5 that integrates Federal domestic prevention, preparedness, response, and recovery plans into one all-discipline, all-hazards plan.

Needline

A requirement that is the logical expression of the need to transfer information among nodes.

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.

O

Operational Activity Model

A representation of the actions performed in conducting the business of an enterprise. The model is usually hierarchically decomposed into its actions, and usually portrays the flow of information (and sometimes physical objects) between the actions. The activity model portrays operational actions not hardware/software system functions. (PSAF)

Operational Activity

An activity is an action performed in conducting the business of an enterprise. It is a general term that does not imply a placement in a hierarchy (e.g., it could be a process or a task as defined in other documents and it could be at any level of the hierarchy of the Operational Activity Model). It is used to portray operational actions not hardware/software system functions. (PSAF)

Operational Node

A node that performs a role or mission. (PSAF)

Organization

An administrative structure with a mission.

P

Platform

A physical structure that hosts systems or system hardware or software items.

Process

A group of logically related activities required to execute a specific task or group of tasks. Note: Multiple activities make up a process.

Product

Data elements you depict graphically, textually, and tabularly to identify architecture components and model their relationships. Architecture products describe characteristics pertinent to the architecture's intended use. (PSAF)

R

Report

A combination of architecture data elements from one or more products combined with additional information. Reports provide a different way of looking at architecture data. (PSAF)

Requirement

A need or demand.

Role

A function or position (Webster's Dictionary)

Rule

Statement that defines or constrains some aspect of the enterprise.

S

Service

A distinct part of the functionality that is provided by a system on one side of an interface to a system on the other side of an interface. (Derived from IEEE 1003.0)

System

Any organized assembly of resources and procedures united and regulated by interaction or interdependence to accomplish a set of specific functions. (PSAF)

System Data Element

The architecture data element or type that stores data from the architecture domain (i.e., it has a value) that is produced or consumed by a system function and that has system data exchange attributes as specified in the Systems Data Exchange Matrix. (PSAF)

System Data Exchange

The collection of System Data Elements and their performance attributes such as timeliness, quality, and quantity values. (PSAF)

Systems Node

A node with the identification and allocation of resources (e.g., platforms, units, facilities, and locations) required to implement specific roles and missions. (PSAF)

System of Systems

A set or arrangement of independent systems that are related or connected to provide a given capability. The loss of any part of the system will degrade the performance or capabilities of the whole.

T

Task

A unit of discrete work, not specific to a single organization, system, or individual, that enables the accomplishment of missions or functions.

Note: Multiple processes accomplish a single task; a single process may support multiple tasks.

Task Force

Any combination of resources assembled to support a specific mission or operational need. All resource elements within a Task Force must have common communications and a designated leader. (NIMS)

Tribal

Any Indian tribe, band, nation, or other organized group or community, including any Alaskan Native Village as defined in or established pursuant to the Alaskan Native Claims Settlement Act (85 stat. 688) (43 U.S.C.A. and 1601 et seq.), that is recognized as eligible for the special programs and services provided by the United States to Indians because of their status as Indians. (NIMS)

U

Unified Command

An application of ICS used when there is more than one agency with incident jurisdiction or when incidents cross political jurisdictions. Agencies work together through the designated members of the UC, often the senior person from agencies and/or disciplines participating in the UC, to establish a common set of objectives and strategies and a single IAP. (NIMS)

Universal Reference Resources

Reference models and information standards that serve as sources for guidelines and attributes that must be consulted while building architecture products. (PSAF)

Appendix C: Dictionary of UML Terms

A

Abstract Class

A class that cannot be directly instantiated. Contrast: concrete class.

Abstraction

1. The act of identifying the essential characteristics of a thing that distinguishes it from all other kinds of things. Abstraction involves looking for similarities across sets of things by focusing on their essential common characteristics. An abstraction always involves the perspective and purpose of the viewer; different purposes result in different abstractions for the same things. All modeling involves abstraction, often at many levels for various purposes.

2. A kind of dependency that relates two elements that represent the same concept at different abstraction levels.

Action

The specification of an executable statement that forms an abstraction of a computational procedure. An action typically results in a change in the state of the system and can be realized by sending a message to an object or modifying a link or a value of an attribute.

Action Sequence

An expression that resolves to a sequence of actions.

Action State

A state that represents the execution of an atomic action, typically the invocation of an operation.

Activation

The execution of an action.

Active Class

A class whose instances are active objects.

Active Object

An object that owns a thread and can initiate control activity. An instance of active class.

Activity Graph

A special case of a state machine that is used to model processes involving one or more classifiers.

Actor (Class)

A coherent set of roles that users of use cases play when interacting with these use cases. An actor has one role for each use case with which it communicates.

Actual Parameter

Synonym: argument.

Adornments

Textual or graphical items that are added to an element's basic notation and are used to visualize details from the element's specification. (One of two annotation mechanisms in UML)

Aggregate (Class)

A class that represents the whole in an aggregation (whole-part) relationship.

Aggregation

A special form of association that specifies a whole-part relationship between the aggregate or whole and a component part.

Annotation Mechanisms

Annotations of existing items in a UML diagram. The two annotation mechanisms are specifications and adornments.

Architecture

The organizational structure and associated behavior of a system. An architecture can be recursively decomposed into parts that interact through interfaces, relationships that connect parts and constraints for assembling parts. Parts that interact through interfaces include classes, components and subsystems.

Artifact

A piece of information that is used or produced by a software development process, such as an external document or a work product. An artifact can be a model, description or software.

Association

The semantic relationship between two or more classifiers that involves connections among their instances.

Attribute

A named property of a class that describes a range of values that instances of the property may hold.

B

Building Blocks

Things, relationships, and diagrams that compose something.

C

Class

A description of a set of objects that share the same attributes, operations, methods, relationships and semantics. A class may use a set of interfaces to specify collections of operations it provides to its environment.

Class Diagram

A diagram that shows a collection of declarative or static model elements, such as classes, types, and their contents and relationships.

Collaboration

The specification of how an operation or classifier, such as a use case, is realized by a set of classifiers and associations playing specific roles used in a specific way. The collaboration defines an interaction.

Collaboration Diagram

A diagram that shows interactions organized around the structure of a model, using either classifiers and associations or instances and links. Unlike a sequence diagram, a collaboration diagram shows the relationships among the instances. Sequence diagrams and collaboration diagrams express similar information, but show it in different ways.

Component

A modular, deployable and replaceable part of a system that encapsulates implementation and exposes a set of interfaces. A component is typically specified by one or more classifiers (e.g., implementation classes) that reside on it and may be implemented by one or more artifacts (e.g., binary, executable or script files).

Component Diagram

A diagram that shows the organizations and dependencies among components.

Concrete Class

A class that can be directly instantiated. Contrast: abstract class.

Constraint

A semantic condition or restriction. Certain constraints are predefined in the UML; others may be user-defined. Constraints are one of three extensibility mechanisms in the UML.

Container

1. An instance that exists to contain other instances and that provides operations to access or iterate over its contents (e.g., arrays, lists, sets).

2. A component that exists to contain other components.

Containment Hierarchy

A namespace hierarchy consisting of model elements and the containment relationships that exist between them. A containment hierarchy forms a graph.

Context

A view of a set of related modeling elements for a particular purpose, such as specifying an operation.

D

Dependency

A relationship between two modeling elements, in which a change to one modeling element (the independent element) will affect the other modeling element (the dependent element).

Deployment Diagram

A diagram that shows the configuration of run-time processing nodes and the components, processes, and objects that live on them. Components represent run-time manifestations of code units.

Derivation

A relationship between an element and another element that can be computed from it. Derivation is modeled as a stereotype of an abstraction dependency with the keyword Derive.

Derived Element

An element that can be computed from other elements and is included for clarity or for design purposes, even though it adds no semantic information.

Diagram

A graphical presentation of a collection of model elements, most often rendered as a connected graph of arcs (relationships) and vertices (other model elements). The UML supports the following diagrams: class diagram, object diagram, use case diagram, sequence diagram, collaboration diagram, statechart diagram, activity diagram, component diagram and deployment diagram.

E

Effect

Specifies an optional procedure to be performed when the transition fires.

Element

An atomic constituent of a model.

Entry Action

An action executed upon entering a state in a state machine, regardless of the transition taken to reach that state.

Event

The specification of a significant occurrence that has a location in time and space. In the context of statechart diagrams, an event is an occurrence that can trigger a transition.

Exit Action

An action executed upon exiting a state in a state machine, regardless of the transition taken to exit that state.

Extend

A relationship from an extension use case to a base use case, specifying how the behavior defined for the extension use case augments (subject to conditions specified in the extension) the behavior for the base use case. The behavior is inserted at the location defined by the extension point in the base use case. The base use case does not depend on performing the behavior of the extension use case.

F

Fire

See "Transition."

G

Guard

A Boolean predicate that provides fine-grained control over the firing of the transition. It must be true for the transition to fire. It is evaluated at the time the event is dispatched. There can be at most one guard per transition.

Generalizable Element

A model element that may participate in a generalization relationship.

Generalization

A taxonomic relationship between a more general element and a more specific element. The more specific element is fully consistent with the more general element and contains additional information. An instance of the more specific element may be used where the more general element is allowed. See: "Inheritance."

I

Inheritance

The mechanism by which more specific elements incorporate structure and behavior of more general elements related by behavior.

Instance

An individual entity with its own identity and value.

Interaction

A specification of how stimuli are sent between instances to perform a specific task. The interaction is defined in the context of a collaboration.

Interaction Diagram

A generic term that applies to several types of diagrams that emphasize object interactions. These include collaboration and sequence diagrams.

Interface

A named set of operations that characterize the behavior of an element.

L

Link

A semantic connection among a tuple of objects. An instance of an association.

Link End

An instance of an association end.

M

Message

A specification of the conveyance of information from one instance to another, with the expectation that activity will ensue. A message may specify the raising of a signal or the call of an operation.

Metadata

The architecture data types, possibly expressed in the form of a physical schema.

Model

A semantically complete abstraction of a system.

N

Node

A node is a classifier that represents a run-time computational resource, which generally has at least a memory and often processing capability. Run-time objects and components may reside on nodes.

Notes

Notes may contain any combination of text or graphics. A note that renders a comment has no semantic impact; it does not alter the meaning of the model to which it is attached. Notes are used to specify requirements, observations, reviews and explanations, in addition to rendering constraints.

O

Object

An entity with a well-defined boundary and identity that encapsulates state and behavior. State is represented by attributes and relationships; behavior is represented by operations, methods and state machines. An object is an instance of a class.

Object Diagram

A diagram that encompasses objects and their relationships at a point in time. An object diagram may be considered a special case of a class diagram or a collaboration diagram.

Operation

An operation is the implementation of a service that can be requested from any object of the class to affect behavior.

P

Package

A package is a general-purpose mechanism for organizing elements into groups.

Postcondition

A constraint that must be true at the completion of an operation.

Precondition

A constraint that must be true when the operation is invoked.

R

Realization

The relationship between a specification and its implementation; an indication of the inheritance of behavior without the inheritance of structure.

Refinement

A relationship that represents a more complete specification of something that has already been specified at a certain level of detail. For example, a design class is a refinement of an analysis class.

Relationship

A semantic connection among model elements. Examples include: dependency, association, generalization, and realization.

S

Sequence Diagram

A diagram that shows object interactions arranged in a time sequence. In particular, it shows the objects participating in the interaction and the sequence of messages exchanged. Unlike a collaboration diagram, a sequence diagram includes time sequences but does not include object relationships. A sequence diagram can exist in a generic form (describes all possible scenarios) and in an instance form (describes one actual scenario). Sequence diagrams and collaboration diagrams express similar information, but show it in different ways.

Signal

The specification of an asynchronous stimulus communicated between instances. Signals may have parameters.

Specification

A declarative description of what something is or does.

Source

A source designates the originating state vertex of the transition.

State

A condition or situation during the life of an object during which it satisfies some condition, performs some activity or waits for some event.

State Machine

A behavior that specifies the sequences of states that an object or interaction goes through during its life in response to events, together with its responses and actions.

Statechart Diagram

A diagram that shows a state machine.

Stereotype

A new type of modeling element that extends the semantics of the metamodel. Stereotypes must be based on certain existing types or classes in the metamodel. Stereotypes may extend the semantics, but not the structure of pre-existing types and classes. Certain stereotypes are predefined in the UML, others may be user defined. Stereotypes are one of three extensibility mechanisms in the UML.

Stimulus

The passing of information from one instance to another, such as raising a signal or invoking an operation. The receipt of a signal is normally considered an event.

Swim Lane

A partition on an activity diagram for organizing the responsibilities for actions. Swim lanes typically correspond to organizational units in a business model.

T

Tagged Value

An extensibility mechanism that adds a new property to things.

Target

Designates the target state vertex that is reached when the transition is taken.

Things

The abstractions that are first-class citizens in a model; relationships tie these things together; diagrams group collections of things. There are four kinds of things in the UML: structural things, behavioral things, grouping things and annotational things.

Thread (of Control)

A single path of execution through a program, a dynamic model or some other representation of control flow. Also, a stereotype for the implementation of an active object as a lightweight process.

Time Event

An event that denotes the time elapsed since the current state was entered.

Time Expression

An expression that resolves to an absolute or relative value of time.

Trace

A dependency that indicates an historical or process relationship between two elements that represent the same concept without specific rules for deriving one from the other.

Transient Object

An object that exists only during the execution of the process or thread that created it.

Transition

A relationship between two states indicating that an object in the first state will perform certain specified actions and enter the second state when a specified event occurs and specified conditions are satisfied. On such a change of state, the transition is said to fire.

Trigger

Specifies the event that fires the transition. There can be at most one trigger per transition.

Type

A stereotyped class that specifies a domain of objects together with the operations applicable to the objects, without defining the physical implementation of those objects. A type may not contain any methods, maintain its own thread of control or be nested. However, it may have attributes and associations. Although an object may have at most one implementation class, it may conform to multiple different types.

U

Use Case (Class)

The specification of a sequence of actions, including variants, that a system (or other entity) can perform, interacting with actors of the system.

Use Case Diagram

A diagram that shows the relationships among actors and use cases within a system.

Use Case Instance

The performance of a sequence of actions being specified in a use case. An instance of a use case.

Use Case Model

A model that describes a system's functional requirements in terms of use cases.

Appendix D: References

Booch, 1999	Booch, G., J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language User Guide*, Addison-Wesley Publishing Company, Reading, Massachusetts, April 1999.
DeMarco, 1979	DeMarco, Tom, *Structured Analysis and Systems Specification*, Prentice-Hall, Englewood Cliffs, New Jersey, 1979.
DoDAF Working Group, 2004	DoD Architecture Framework Working Group, *DoD Architecture Framework Version 1.0*, "Volume I: Definitions and Guidelines," February 2004.
DoDAF Working Group, 2004	DoD Architecture Framework Working Group, *DoD Architecture Framework Version 1.0*, "Volume II: Product Descriptions," February 2004.
FIPS, 1993	*Integration Definition for Function Modeling* (IDEF0) method, 1993, pp. 183
FIPS, 1993	*Integration Definition for Function Modeling* (IDEF1X) method, 1993, pp. 184
Harel, 1987a	Harel, D., "Statecharts: A Visual Formalism for Complex Systems," *The Science of Computer Programming*, 1987, 8, pp. 231-274.
Harel, 1987b	Harel, D., A. Pnueli, J.P. Schmidt, and R. Sherman, *On The Formal Semantics of Statecharts*, Proceedings, Second IEEE Symposium, Logic Computer Science, Dorset House, New York, 1987, pp. 54-64.
IEEE STD 1471, 2000	Institute of Electrical and Electronics Engineers, IEEE STD 1471, *Recommended Practice for Architectural Description of Software-Intensive Systems*, The Institute of Electrical and Electronics Engineers, Inc., New York, New York, 2000.
IEEE 610.12, 1990	Institute of Electrical and Electronics Engineers, IEEE STD 610.12, *Standard Glossary of Software Engineering Terminology*, The Institute of Electrical and Electronics Engineers, Inc., Piscataway, New Jersey, 1990.
Kristensen, 1998	Kristensen, Lars M., S. Christensen, and Kurt Jensen, The Practitioner's Guide to Coloured Petri Nets, Springer-Verlag, 1998.
Naqvi, 1989	Naqvi, Shamim, and Shalom Tsur, *LDL: A Logical Language for Data Knowledge Bases*, Computer Science Press, Rockville, Maryland, 1989.
Neill, 2002	Neill, C.J., and J.D. Holt, "Adding Temporal Modeling To The UML To Support Systems Design," *Systems Engineering*, Vol. 5, No. 3, Fall 2002.
OMB, 2003	Office of Management and Budget, *Business Reference Model (BRM) v2.0, Service Component Reference Model (SRM) v1.0, Technical Reference Model (TRM) v1.0,* Released June 12, 2003, *Performance Reference Model (PRM)*, Released July 2003.
OMB, 2000	Office of Management and Budget, *Circular A-130: Management of Federal Information Resources*, 30 November 2000.
OMG, 2003	Object Management Group, *Unified Modeling Language Specification*, Version 1.5, Framingham, Massachusetts, March 2003, Available: http://www.omg.org.
OMG, 2001	Object Management Group, UML Primer 2000, *What Is OMG-UML and Why Is It Important?*, Framingham, Massachusetts, Available: http:/.www.omg.org/news/pr97/umlprimer.html, 2000.
Rumbaugh, 1991	Rumbaugh, J., M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen, *Object-Oriented Modeling and Design*, Prentice Hall, Englewood Cliffs, New Jersey, 1991.
Rumbaugh, 1999	Rumbaugh, J., I. Jacobson, and G. Gooch, *The Unified Modeling Language Reference Manual*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1999.
Warmer, 1999	Warmer, Jos B., and Anneke G. Kleppe, *The Object Constraint Language*, Addison-Wesley, 1999.
Zachman, 1987	Zachman, J.A., A Framework for Information Systems Architecture, IBM Systems Journal, 26(3): 276-291, 1987, Available: http://www.zifia.com/.