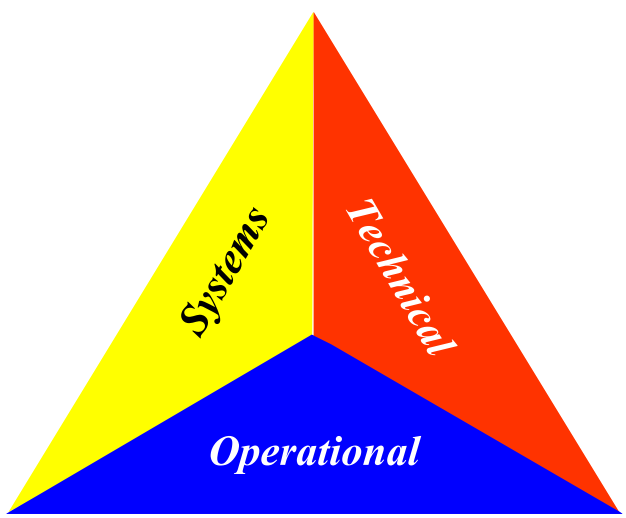
DoD Architecture Framework Working Group

DoD Architecture Framework Version 1.0



Volume II: Product Descriptions

9 February 2004

TABLE OF CONTENTS

SECTION **PAGE**

TABLE OF CONTENTS (Cont)

SECTION PAGE

TABLE OF CONTENTS (Cont)

SECTION **PAGE**

TABLE OF CONTENTS (Cont)

SECTION PAGE

TABLE OF CONTENTS (Cont)

SECTION PAGE

TABLE OF CONTENTS (Cont)

SECTION PAGE

TABLE OF CONTENTS (Cont)

SECTION PAGE

TABLE OF CONTENTS (Cont)

SECTION **PAGE**

TABLE OF CONTENTS (Cont)

SECTION **PAGE**

ANNEX

LIST OF FIGURES

FIGURE **PAGE**

LIST OF FIGURES (Cont)

FIGURE **PAGE**

LIST OF FIGURES (Cont)

FIGURE **PAGE**

LIST OF FIGURES (Cont)

LIST OF TABLES

LIST OF TABLES (Cont)

TABLE **PAGE**

1 INTRODUCTION

1.1 OVERVIEW

The Department of Defense (DoD) Architecture Framework (DoDAF), Version 1.0, defines a common approach for DoD architecture[1] description development, presentation, and integration for both warfighting operations and business operations and processes. The Framework is intended to ensure that architecture descriptions can be compared and related across organizational boundaries, including Joint and multinational boundaries.

An architecture description is a representation of a defined domain, as of a current or future point in time, in terms of its constituent parts, what those parts do, how the parts relate to each other and to the environment, and the rules and constraints governing them. Within the DoDAF, architectures are described in terms of three views: Operational View (OV), Systems View (SV), and Technical Standards View (TV). An architecture description is composed of architecture products that are interrelated within each view and are interrelated across views. Architecture products are those graphical, textual, and tabular items that are developed in the course of gathering architecture data, identifying their composition into related architecture components or composites, and modeling the relationships among those composites to describe characteristics pertinent to the architecture's intended use. Underlying the products is the All-DoD Core Architecture Data Model (CADM), which defines a standard set of architecture data entities and relationships of architecture data.

The term *integrated architecture* refers to an architecture description that has integrated OVs, SVs, and TVs (i.e., there are common points of reference linking the OV and SV and also linking the SV and TV). The Operational Activity to Systems Function Traceability Matrix (SV-5), for example, relates operational activities from the Operational Activity Model (OV-5) to system functions from the Systems Functionality Description (SV-4); the SV-4 system functions are related to systems in the Systems Interface Description (SV-1), thus bridging the OV and SV. An architecture is defined to be 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.

The term *architecture* is generally used both to refer to an architecture description and an architecture implementation. An architecture description is a representation of a current or postulated real-world configuration of resources, rules, and relationships. Once the representation enters the design, development, and acquisition portion of the system development life-cycle process, the architecture description is then transformed into a real implementation of capabilities and assets in the field. The Framework itself does not address this representation-to-implementation transformation process but references policies that are relevant to that process.

---

[1] "An architecture is the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution." [IEEE STD 1471, 2000]

Hereafter in this document, the term *architecture* will be used as a shortened reference to *architecture description*. Occasionally the term *architecture description* is used for emphasis.

1.2 PURPOSE AND INTENDED AUDIENCE

The purpose of this volume is to define, provide a purpose for, and describe in detail each Framework product. This volume is organized with various readers in mind.

For the **manager** who needs to lead architecture development projects and who may need to use an architecture to make acquisition, budgeting, or resourcing decisions, **product definition** and **product purpose** subsections are provided in each product section to:

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

For the **architect and engineering team** who need to develop architecture products for high-level decision makers for use in decision support analysis, a **product detailed description** and an architecture **data element table** subsection are provided in each product section to:

– Enable the architect and his team to identify products to be included in the architecture based on the architecture's intended use (see Figure 2-2, Architecture Products by 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

For the **architecture data modelers, tool developers, and engineers** who are involved with implementing an architecture data repository to store and manipulate Framework architecture data elements, a **CADM support** subsection **(including entity-relationships [ER] diagram)** is provided in each product section.

1.3 ORGANIZATION OF THIS VOLUME

Section 2 provides an overview of basic concepts of the DoD architecture approach. Product descriptions are provided in sections organized by view: Section 3, All-Views; Section 4, Operational View; Section 5, Systems View; and Section 6, Technical Standards View. Section 7 contains details of the architecture data element and product relationships. The product definitions are provided according to the format described below.

1.3.1 Product Definitions Structure

Products for each view are presented individually, with the following separate subsections:

1. A product overview for managers (*what* is it)

2. A brief statement on the purpose of the product for managers and architects (*why* is it useful)

3. A detailed description for architects and engineers who are developing architectures, including:

   – Narrative details about the product and its representation in Structured Analysis (SA) and in Object-Oriented (OO) notation, where applicable

   – One or more generic templates and/or examples (For most of the 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, relationships of these architecture data elements to architecture data elements in other products

4. Descriptive text followed by an ER diagram that describes the corresponding CADM entities. This section is included for the data modelers, tool developers, and engineers who are concerned with implementing a data repository to store and manipulate Framework data elements. A discussion of the correspondence between basic DoDAF data elements and a CADM-like conceptual model is detailed in the section on CADM in the Deskbook. An index of the conceptual entities and their corresponding CADM entities appears at the end of the same Deskbook section.

1.3.2 Architecture Data Element Tables Structure

An architecture data element table for each product provides definitions of the metadata (i.e., the architecture data types that comprise the products). For each architecture data type, attributes are defined to provide added detail about the architecture data type characteristics. The architecture data elements provide structure for storing architecture data about a given architecture that should be captured in the product and stored in the Integrated Dictionary.[2] 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. At a minimum, an integrated architecture consists of Overview and Summary Information (AV-1), Integrated Dictionary (AV-2), Operational Node Connectivity (OV-3), Operational Activity Model (OV-5), Systems Interface Description (SV-1) and Technical Standards Profile (TV-1). For additional products, architecture 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. Figure 2-2 provides a use matrix that provides guidelines on which products should be developed based on intended use.

To assist architects and engineers in developing products, the architecture data element tables are organized 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 (e.g., some icons may be placed *inside* other icons to indicate containment or 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 four categories listed above

---

[2] An integrated architecture database conforming to the CADM would be an example of the Integrated Dictionary storing information about product elements (and their definitions).

2 ARCHITECTURE BASICS - VIEWS, PRODUCTS, AND ARCHITECTURE DATA

2.1 ARCHITECTURE VIEWS

As defined in Volume I, the term *integrated architecture* refers to an architecture description that has an Operational View (OV), Systems View (SV), and Technical Standards View (TV), and the views are integrated (i.e., there are common points of reference linking the OV and SV and also linking the SV and TV). The Operational Activity to Systems Functionality Traceability Matrix (SV-5), for example, relates operational activities from the Operational Activity Model (OV-5) to system functions from the Systems Functionality Description (SV-4); the SV-4 system functions are related to systems in the Systems Interface Description (SV-1); thus bridging the OV and SV. An architecture is defined to be 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. **Figure 2-1** illustrates the major relationships.

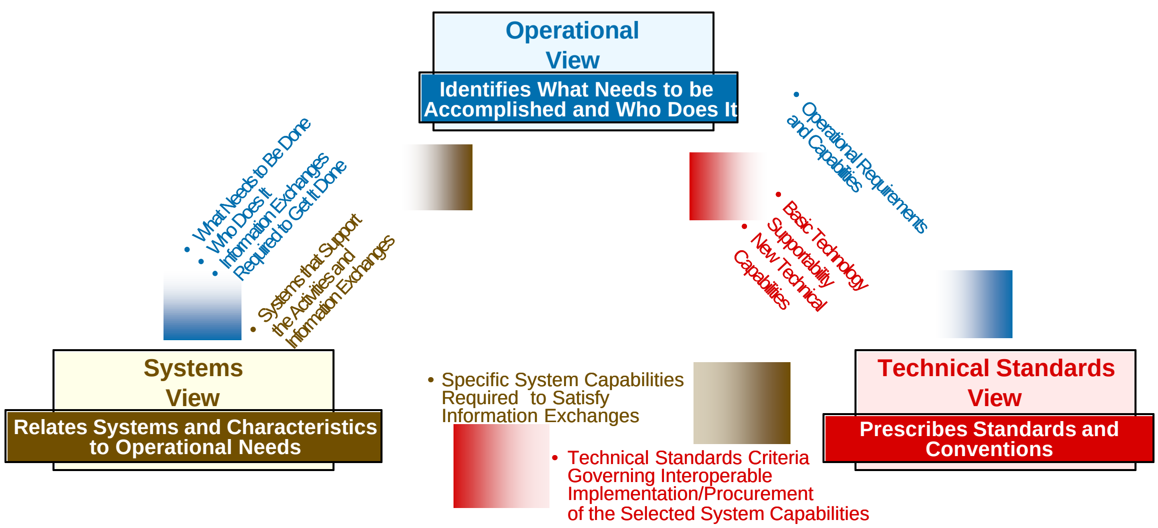


Figure 2-1. Fundamental Linkages Among the Views

2.1.1 Definition of the Operational View

The OV is a description of the tasks and activities, operational elements, and information exchanges required to accomplish DoD missions. DoD missions include both warfighting missions and business processes. The OV contains graphical and textual products that comprise an identification of the operational nodes and elements, assigned tasks and activities, and information flows required between nodes. It defines the types of information exchanged, the frequency of exchange, which tasks and activities are supported by the information exchanges, and the nature of information exchanges.

2.1.2 Definition of the Systems View

The SV is a set of graphical and textual products that describes systems and interconnections providing for, or supporting, DoD functions. DoD functions include both warfighting and business functions. The SV associates system resources to the OV. These system resources support the operational activities and facilitate the exchange of information among operational nodes.

2.1.3 Definition of the Technical Standards View

The TV is the minimal set of rules governing the arrangement, interaction, and interdependence of system parts or elements. Its purpose is to ensure that a system satisfies a specified set of operational requirements. The TV provides the technical systems implementation guidelines upon which engineering specifications are based, common building blocks are established, and product lines are developed. The TV includes a collection of the technical standards, implementation conventions, standards options, rules, and criteria organized into profile(s) that govern systems and system elements for a given architecture.

2.1.4 Definition of the All-Views

There are some overarching aspects of an architecture that relate to all three views. These overarching aspects are captured in the All-Views (AV) products. The AV products provide information pertinent to the entire architecture but do not represent a distinct view of the architecture. AV products set the scope and context of the architecture. The scope includes the subject area and time frame for the architecture. The setting in which the architecture exists comprises the interrelated conditions that compose the context for the architecture. These conditions include doctrine; tactics, techniques, and procedures; relevant goals and vision statements; concepts of operations (CONOPS); scenarios; and environmental conditions.

2.2 ARCHITECTURE PRODUCTS

Architecture products are those graphical, textual, and tabular items that are developed in the course of gathering architecture data, identifying their composition into related architecture components or composites, and modeling the relationships among those composites to describe characteristics pertinent to the architecture purpose. Choosing products to develop for a given architecture description depends on the architecture's intended use.

Table 2-1 lists products. The first column indicates the view applicable to each product. The second column provides an alphanumeric reference identifier for each product. The third column gives the formal name of the product. The fourth column captures the general nature of the product's content. The sequence of products in the table does not imply a sequence for developing the products.

All products, even those whose primary presentation is graphical, should contain explanatory text.

Previous versions of the Framework described essential and supporting products; this version conveys a similar emphasis by specifying that an integrated architecture, as referenced in DoDI 5000.2, DoDI 4630.8, CJCSI 3170.01C, CJCSM 3170.01, and CJCSI 6212.01, consists of AV-1, AV-2, OV-2, OV-3, OV-5, SV-1, and TV-1, at a minimum. This new guidance is intended to ensure that consistent types of architecture data underlying the products are available to support a range of DoD objectives.

Additional products may be developed for a given architecture description depending on the intended use of the architecture (see **Figure 2-2**). The use matrix provides guidelines on what architecture products are applicable to various uses of architecture and emphasizes the development of architectures to support decision making associated with a number of DoD processes (see Volume I for more information on DoD processes). The matrix demonstrates the need for an integrated architecture (across three views) for most uses and the need for integration between the staffs of the operational and acquisition communities. The architecture products appropriate for any individual use case is highly dependent on the specific situation, objectives, and scope of the effect. Therefore, architects should consider the guidelines provided in the use matrix but make decisions based on the specifics of their particular architecture and its intended use.

Table 2-1. List of Products

Applicable View	Framework Product	Framework Product Name	General Description
All Views	AV-1	Overview and Summary Information	Scope, purpose, intended users, environment depicted, analytical findings
All Views	AV-2	Integrated Dictionary	Architecture data repository with definitions of all terms used in all products
Operational	OV-1	High-Level Operational Concept Graphic	High-level graphical/textual description of operational concept
Operational	OV-2	Operational Node Connectivity Description	Operational nodes, connectivity, and information exchange needlines between nodes
Operational	OV-3	Operational Information Exchange Matrix	Information exchanged between nodes and the relevant attributes of that exchange
Operational	OV-4	Organizational Relationships Chart	Organizational, role, or other relationships among organizations
Operational	OV-5	Operational Activity Model	Capabilities, operational activities, relationships among activities, inputs, and outputs; overlays can show cost, performing nodes, or other pertinent information
Operational	OV-6a	Operational Rules Model	One of three products used to describe operational activity—identifies business rules that constrain operation
Operational	OV-6b	Operational State Transition Description	One of three products used to describe operational activity—identifies business process responses to events
Operational	OV-6c	Operational Event-Trace Description	One of three products used to describe operational activity—traces actions in a scenario or sequence of events
Operational	OV-7	Logical Data Model	Documentation of the system data requirements and structural business process rules of the Operational View
Systems	SV-1	Systems Interface Description	Identification of systems nodes, systems, and system items and their interconnections, within and between nodes
Systems	SV-2	Systems Communications Description	Systems nodes, systems, and system items, and their related communications lay-downs
Systems	SV-3	Systems-Systems Matrix	Relationships among systems in a given architecture; can be designed to show relationships of interest, e.g., system-type interfaces, planned vs. existing interfaces, etc.
Systems	SV-4	Systems Functionality Description	Functions performed by systems and the system data flows among system functions
Systems	SV-5	Operational Activity to Systems Function Traceability Matrix	Mapping of systems back to capabilities or of system functions back to operational activities
Systems	SV-6	Systems Data Exchange Matrix	Provides details of system data elements being exchanged between systems and the attributes of that exchange
Systems	SV-7	Systems Performance Parameters Matrix	Performance characteristics of Systems View elements for the appropriate time frame(s)
Systems	SV-8	Systems Evolution Description	Planned incremental steps toward migrating a suite of systems to a more efficient suite, or toward evolving a current system to a future implementation
Systems	SV-9	Systems Technology Forecast	Emerging technologies and software/hardware products that are expected to be available in a given set of time frames and that will affect future development of the architecture
Systems	SV-10a	Systems Rules Model	One of three products used to describe system functionality—identifies constraints that are imposed on systems functionality due to some aspect of systems design or implementation
Systems	SV-10b	Systems State Transition Description	One of three products used to describe system functionality—identifies responses of a system to events
Systems	SV-10c	Systems Event-Trace Description	One of three products used to describe system functionality—identifies system-specific refinements of critical sequences of events described in the Operational View
Systems	SV-11	Physical Schema	Physical implementation of the Logical Data Model entities, e.g., message formats, file structures, physical schema
Technical	TV-1	Technical Standards Profile	Listing of standards that apply to Systems View elements in a given architecture
Technical	TV-2	Technical Standards Forecast	Description of emerging standards and potential impact on current Systems View elements, within a set of time frames

The following legend is used in the figure:

- A light gray cell (■) indicates the product is required in order to have an integrated architecture.
- A dark gray cell (■) indicates the identified product is specified in policy.
- A solid black circle (●) indicates the product is highly applicable to the indicated use (i.e., the product should be developed when the architecture is intended to support the indicated use).
- A white circle with a center black dot (⊙) indicates that the product is often or partially applicable (i.e., consideration should be given to developing the designated product when the architecture is intended to support the indicated use).
- A blank cell indicates that the product is usually not applicable (i.e., there is usually no need to develop the designated product when the architecture is intended to support the indicated use).

The list of uses is not exhaustive; instead, it is intended to provide initial insight into the use of the various architecture products in supporting DoD processes. Future versions of the Framework are expected to expand the uses described. By listing a minimum set of products required for an integrated architecture as prescribed in DoD Directives and Instructions, consistency of products and their constituent architecture data elements across architectures is assured.

APPLICABLE ARCHITECTURE PRODUCTS

RECOMMENDED USES OF ARCHITECTURE:

	All View		Operational View (OV)							Systems View (SV)											Tech Stds View	
	1	2	1	2	3	4	5	6	7	1	2	3	4	5	6	7	8	9	10	11	1	2
Planning, Programming, Budgeting Execution Process																						
Capability-Based Analysis for IT Investment Decisions	●	●	●	●	●	●	●	●	⊙	●	⊙	●	●	●	●	●	●	●	●		●	⊙
Modernization Planning and Technology Insertion/Evolution	●	●	⊙	●	⊙	⊙	●	⊙		●	⊙	⊙	⊙	●	⊙	●	●	●			●	●
Portfolio Management	●	●		●			●	⊙		●			⊙	●		⊙	●				⊙	
Joint Capabilities Integration and Development System																						
JCIDS Analysis (FAA, FNA, FSA)	●	●	●	●	⊙	⊙	●	●		●	⊙		⊙	●							⊙	
ICD/CDD/CPD/CRD	●	●	●	●	●		●	●		●	⊙	⊙	●	●	●	⊙	⊙	⊙	●		●	⊙
Analysis of Alternatives (AoA)	●	●	●	●	⊙		●	●		●	⊙	⊙	●	●	⊙	⊙	⊙	⊙			⊙	⊙
Acquisition Process																						
Acquisition Strategy	●	●	●	●	⊙		●	⊙		●	⊙			●							●	
C4ISP	●	●	●	●	●		●	●		●			⊙		●	⊙					●	⊙
System Design and Development	●	●		●	●		●	●	⊙	●	●	●	●	●	●	●	●	⊙	⊙	⊙	●	⊙
Interoperability and Supportability of NSS and IT Systems	●	●	●	●	●	⊙	●	⊙	⊙	●	⊙	●	●		●	●	⊙	⊙	⊙	⊙	●	⊙
Integrated Test & Evaluation	●	●		●	●	⊙	●	●	⊙	●	●	●	●	●	●	●			●	⊙	●	
Operations (Assessment, Planning, Execution, …)																						
Operations Planning & Execution	●	●	●	●	●	●	●	●	⊙	●	●	⊙	⊙	●	⊙	⊙					⊙	
CONOPS & TTP	●	●	●	●	●	●	●	●		●	⊙	⊙	⊙	⊙							⊙	
Communications Plans	●	●	●	●		⊙	⊙			●	●						⊙	⊙			●	⊙
Exercise Planning & Execution	●	●	●	●	●	●	●	●		●	●	⊙	⊙	⊙	⊙						⊙	
Organizational Design	●	●	●	●	●	●	●	⊙	⊙	⊙	⊙			⊙								
BPR/FPI	●	●	⊙	●	●	●	●	●	⊙													

● = Product is highly applicable

⊙ = Product is often or partially applicable

(dark shading) = Product is specifically addressed in policy

(light shading) = Product is required for an integrated architecture

blank = Product is usually not applicable

Figure 2-2. Architecture Products by Use

2.3 ARCHITECTURE DATA ELEMENTS

As stated earlier, architecture data are the underlying basic elements that comprise a given architecture. The Framework products are those graphical, textual, and tabular items that are developed in the course of gathering architecture data, identifying their composition into related architecture products and modeling the relationships among those products to describe characteristics pertinent to the architecture purpose or intended use.

2.3.1 Architecture Data Model

The heart of interoperability is the preservation of meaning and relationships during architecture data reuse. An architecture data model is a structured representation of the architecture data elements pertinent to an architecture that also defines the relationships among architecture data. Agreement on an architecture data model is essential to the reuse of architecture data, as well as the implementation of architecture databases, regardless of the technology chosen (e.g., relational or object-oriented) for building and managing architecture databases. In addition, a common architecture data model can serve as the basis for defining common Extensible Markup Language tags for architecture data import and export, product extraction, and direct exchange.

2.3.2 Architecture Framework Data Model

The All-DoD Core Architecture Data Model (CADM) was developed cooperatively by representatives of the Office of the Secretary of Defense, Combatant Commands, Military Services, and Defense Agencies as the DoD standard architecture data model for Framework-based architecture data elements. The CADM is built using the Integrated Definition for Data Modeling (IDEF1X) [FIPS 184, 1993] methodology, notation, and forms. More than 95 percent of the entities and attributes from the CADM are approved as DoD architecture data standards. Using relational technology labels, for example, the entities from the CADM provide specifications for tables in a database, and the CADM attributes provide specifications for the fields (architecture data element attributes) in the rows of such tables [All-CADM, 2003a, b, c].

2.4 ARCHITECTURE PRODUCT DEVELOPMENT

The Framework products portray the basic architecture data elements and relationships that constitute an architecture description. In Volume I of the Framework, a process is described for developing architecture descriptions. The first four steps of the architecture development process consist of defining a purpose for building the architecture, defining a scope for the architecture, identifying the key architecture characteristics, and determining which products to build (this step depends on the scope and the purpose of the architecture). These four steps are independent of any methodology[3] that might be used in designing the architecture and require the involvement of few persons—the architect and the users.

2.4.1 Product Development Methodology Support

Step five of the architecture development process (described in Volume I) consists of building the requisite products. The Framework does not advocate the use of any one methodology (e.g., structured analysis vs. object orientation), or one notation over another (e.g., IDEF1X or UML notation) to complete this step, but products should contain the required instances of architecture data elements and relationships (i.e., those marked with an asterisk [*]). However, the need for a well-defined and rigorous methodology is acknowledged and emphasized. There are several candidate methodologies available for consideration, and the choice is ultimately governed by 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.

The actual gathering, analysis, and synthesis of information into an integrated architecture may be conducted using an integrated tool or set of tools that allow the development of the products and accompanying text. The use of an integrated tool or tool suite is highly recommended for developing an integrated architecture, for consistency and version control. The selected tool(s) should allow the architect to produce consistent products/views by performing cross-product checking. The selected tool(s) should also include a mechanism for storing, updating, and retrieving architecture data and their relationships and an ability to automatically generate an integrated dictionary. In addition, the tool should be capable of importing/exporting from a CADM-compliant database.

2.4.2 Architecture Products and Levels of Detail

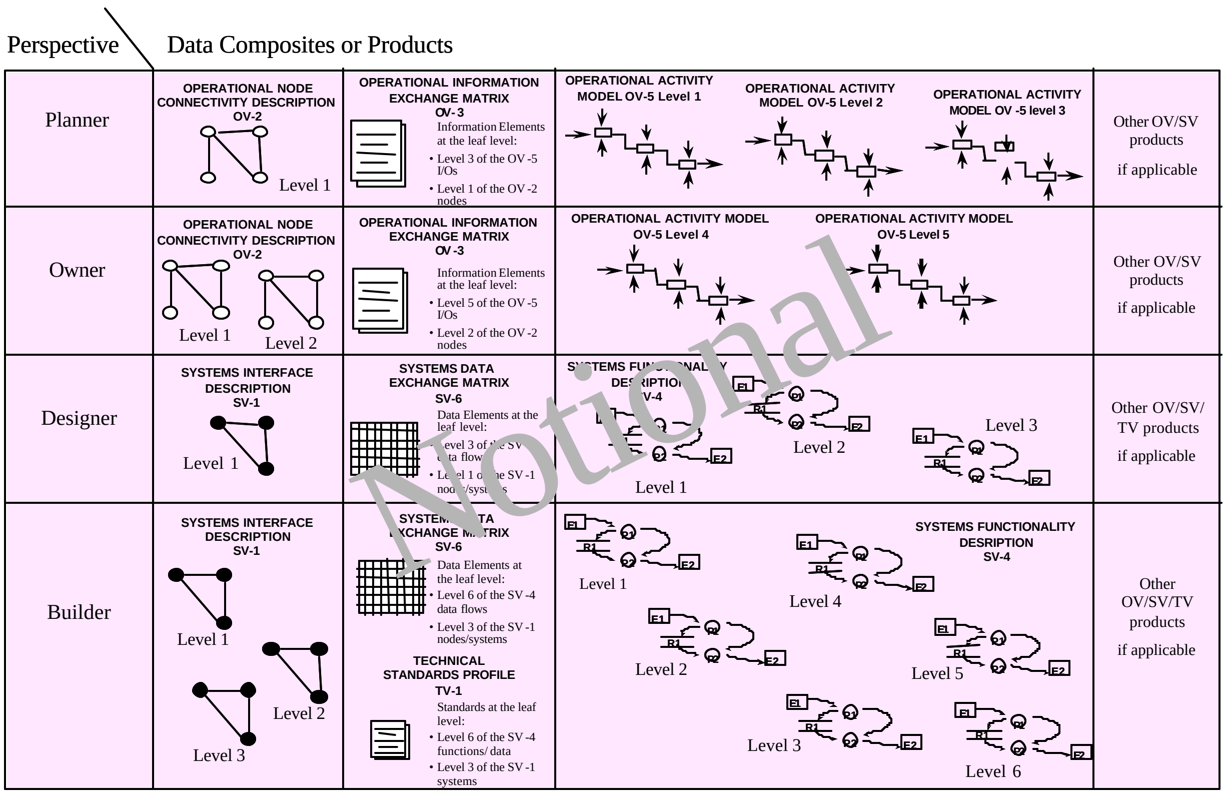
Most graphical products (e.g., OV-2, OV-5, SV-1, and SV-4) permit the modeling of their respective architecture data elements using decomposition (i.e., several diagrams of the same product may be developed for the same architecture, where each diagram shows an increasing level of detail). An example of levels of detail are the various perspectives such as planner, owner, designer, or builder defined by Zachman [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.

---

[3] The 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 Framework defines an approach for developing architecture descriptions, it does not specify a methodology for developing an architecture description.

Within each perspective, all products developed should remain cohesive with respect to the level of detail. For example, if one diagram of OV-2 operational nodes is developed that shows aggregated organizations only, then it is imperative that the corresponding OV-5 product be developed 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 that are used at the operational nodes shown in OV-2 (and not their subordinate operational nodes).

A good guide to tracking the level of detail of an architecture is to always ensure that the information is at the level of detail that is meaningful to the intended user of the architecture. In addition, a good rule of thumb is to restrict decomposition levels for any one type of diagram within the same perspective to no more than three levels because that is generally sufficient to provide the required level of granularity for a stated objective. **Figure 2-3** illustrates some of the decomposition rules of thumb for various perspectives.



No more than 6 levels of decomposition for each type of product within a perspective
All products within a perspective remain cohesive as to level of detail provided in each

Figure 2-3. Perspectives and Decomposition Levels

The products illustrated in Figure 2-3 and the number of decomposition levels shown are not significant but are only examples. The collection of products for each perspective (or level of detail) comprises one model of the architecture. To conduct adequate analyses, an iterative process where multiple architecture models are developed (one for each perspective) is usually needed.

2.4.3 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, the Framework products may be developed by applying an iterative method. 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, different models are developed 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 only a minimum of Framework products are developed to help describe a new concept of operations, a few products may be developed to produce one model of this architecture (denoted *Model A*).

This first model may consist of only highly abstract/generic sets of operational nodes, operational activities, and so forth. Later, when new details need to be added and the architecture is expanded to show more design detail, a new model (*Model B*, consisting of modified Model A products plus additional products as necessary) must be developed.

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 (one that may consist of multiple subordinate operational nodes, but it is deemed unnecessary to show those subordinate nodes at the Model A level). In Model B, the operational node of Model A's product may now be expanded to show the subordinate nodes. No new root-level Framework operational nodes should be introduced at this level that do not trace back to the previous model. For example, if, in the process of model refinement, it is determined 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 Model A's OV-2 needs to be updated 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.

2.4.4 Product Templates

Where applicable, the templates for the Framework products reference industry standard methodologies and techniques, although there is no requirement to comply with the template's chosen standard. Regardless of the technique used to develop the product, the architecture data elements and their relationships, as defined in the architecture data elements tables, must be accurately reflected, including relationships to architecture data elements in other products. Where applicable, the templates for the Framework products reference Structured Analysis (SA) or Object Oriented (OO) standard notation(s).

2.4.5 Object-Orientation and the Unified Modeling Language Support

2.4.5.1 Relationship to the Unified Modeling Language

During the last few years, the Unified Modeling Language (UML) has emerged as the dominant and most prevalent language for OO modeling irrespective of the development process used. The Object Management Group (OMG) characterizes UML as "a general-purpose modeling language for specifying, visualizing, constructing and documenting the artifacts of software systems, as well as for business modeling and other non-software systems" [OMG, 2000].

The UML representation is provided in this version of the Framework 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 Framework products.

It should be noted that this document is not a complete and thorough tutorial on the entire UML and the processes for using that language. There are numerous books written on UML and its applications to software and systems engineering, and, in fact, this is a continuing and ongoing research area. Interested readers can consult these other reference books for additional information on UML and application techniques. For further reading material on UML and OO methodologies, see Annex D, References.

2.4.5.2 Comments on a UML Representation Multi-diagram and Multi-model Approach

The Framework UML representation uses UML notation to model both Framework 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 being modeled in the diagram via the use cases. However, in the Framework'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 and are intended to add information and flow detail to the use cases.

In addition, the same set of diagram types may be used to model several operational products. The UML diagram for each type of operational product will represent a different aspect of the architecture. For example, a UML class diagram is used to model OV-4 organizational charts, as well as to define a Logical Data Model (OV-7). In the case of OV-4, class diagram notation is utilized to allow the modeling of relationships among organizations (as opposed to relationships among classes of data objects).[4] 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 used in OV-7, on the other hand, show classes that relate to OV-5 activities and information flows.

For completeness, definitions of the UML terms used in this section have been included in Annex B, under the subheading *Dictionary of UML Terms*. Where a term used in discussing UML has also been used in the Framework document to convey a slightly different meaning, the term is fully qualified within the UML representation sections. For example, the term *component(s)* used in the UML sections refers to any type of system software (as in systems engineering), while the term *component* as referenced in the Framework denotes DoD organizational units. Within the UML section, the term is qualified as *UML Component*.

---

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

2.5 PRODUCT AND ARCHITECTURE DATA ELEMENT RELATIONSHIPS

There are general relationships that logically interconnect the Framework products from one view to 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 2-4** illustrates some relationships among the architecture data elements for a subset of the products.

Section 7 of this volume contains a detailed description of these product and architecture data element relationships.

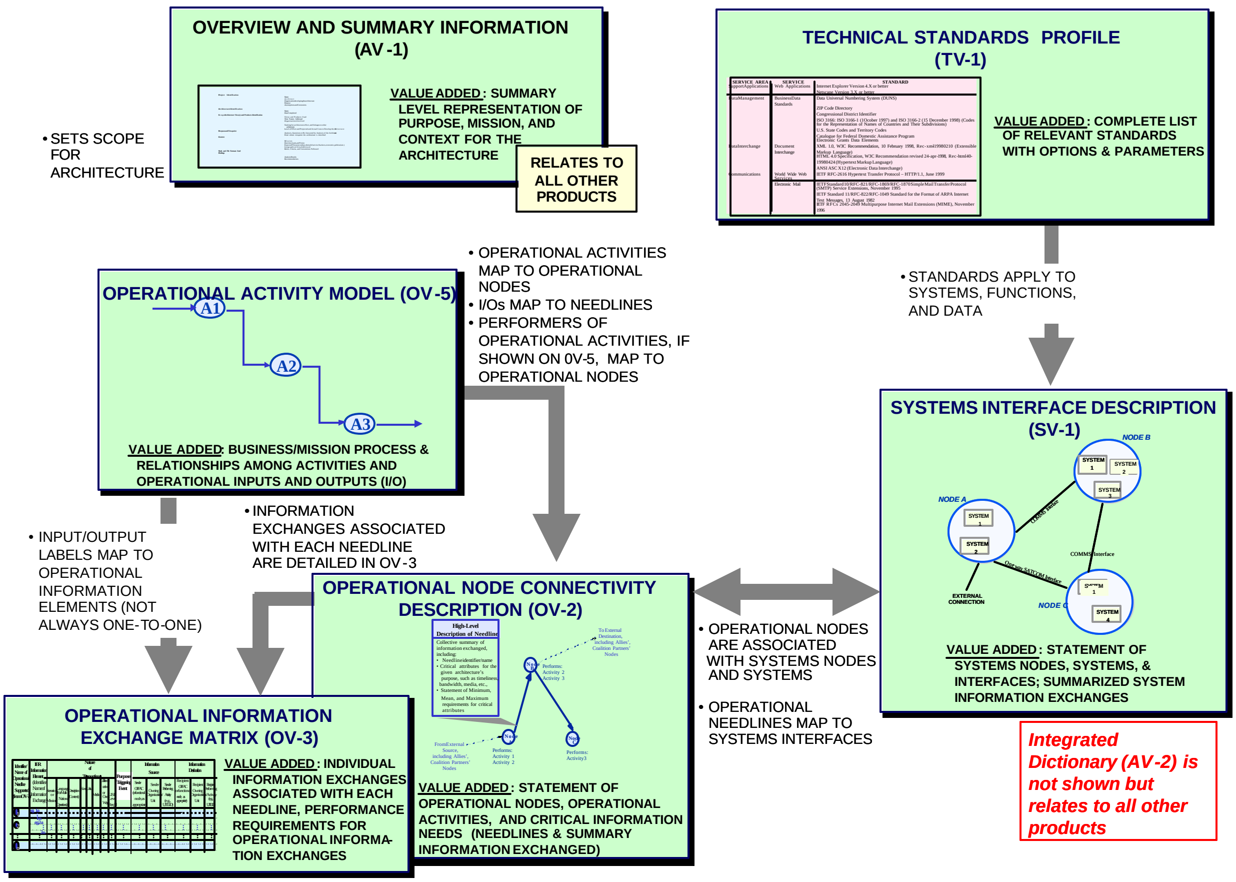


Figure 2-4. Fundamental Linkages Among the Products and Architecture Data Elements

2.6 CADM SUPPORT FOR ARCHITECTURE PRODUCTS[5]

Each instance (including each version or variant used for analysis) of an architecture product is identified and stored as an instance of DOCUMENT in a CADM-conformant database. One of the attributes of DOCUMENT (DOCUMENT ARCHITECTURE PRODUCT TYPE CODE) is used to separate DOCUMENT into subtypes (non-overlapping subsets) as shown in the entity-relationship diagram from the CADM in **Figure 2-5**. In some cases, the same DOCUMENT subtype is used for both an operational and a systems product of the same kind (e.g., OV-5 and SV-4 use essentially the same structures to specify operational activities, processes, and system functionality). Unlike most of the other figures from the CADM for specific architecture products that follow, Figure 2-5 shows attribute detail, providing descriptive values and indicating for some entities (e.g., ACTIVITY-MODEL-SPECIFICATION) associations to entities not shown in the diagram (e.g., Activity Model INFORMATION-ASSET IDENTIFIER and Node Tree DOCUMENT IDENTIFIER) whose reference is used to fully specify an architecture product (e.g., OV-7). As shown in Figure 2-5, when a DOCUMENT subtype is the top-level entity used to characterize a Framework product, the Framework product identifier (e.g., OV-1, OV-2, and SV-1) is included at the end of the entity name.

---

[5] In this section and the other CADM descriptions that follow, bold blue font in the diagram is used for approved entities (and attributes), and blue lines depict approved relationships (already part of the DoD Data Architecture, the data model that structures approved data standards). Other colors are used to identify parts of the CADM that have not yet completed DoD-wide data standardization. Specifically, bold red italic font is used for entities and attributes that are in Candidate status under DoD-wide data standardization.

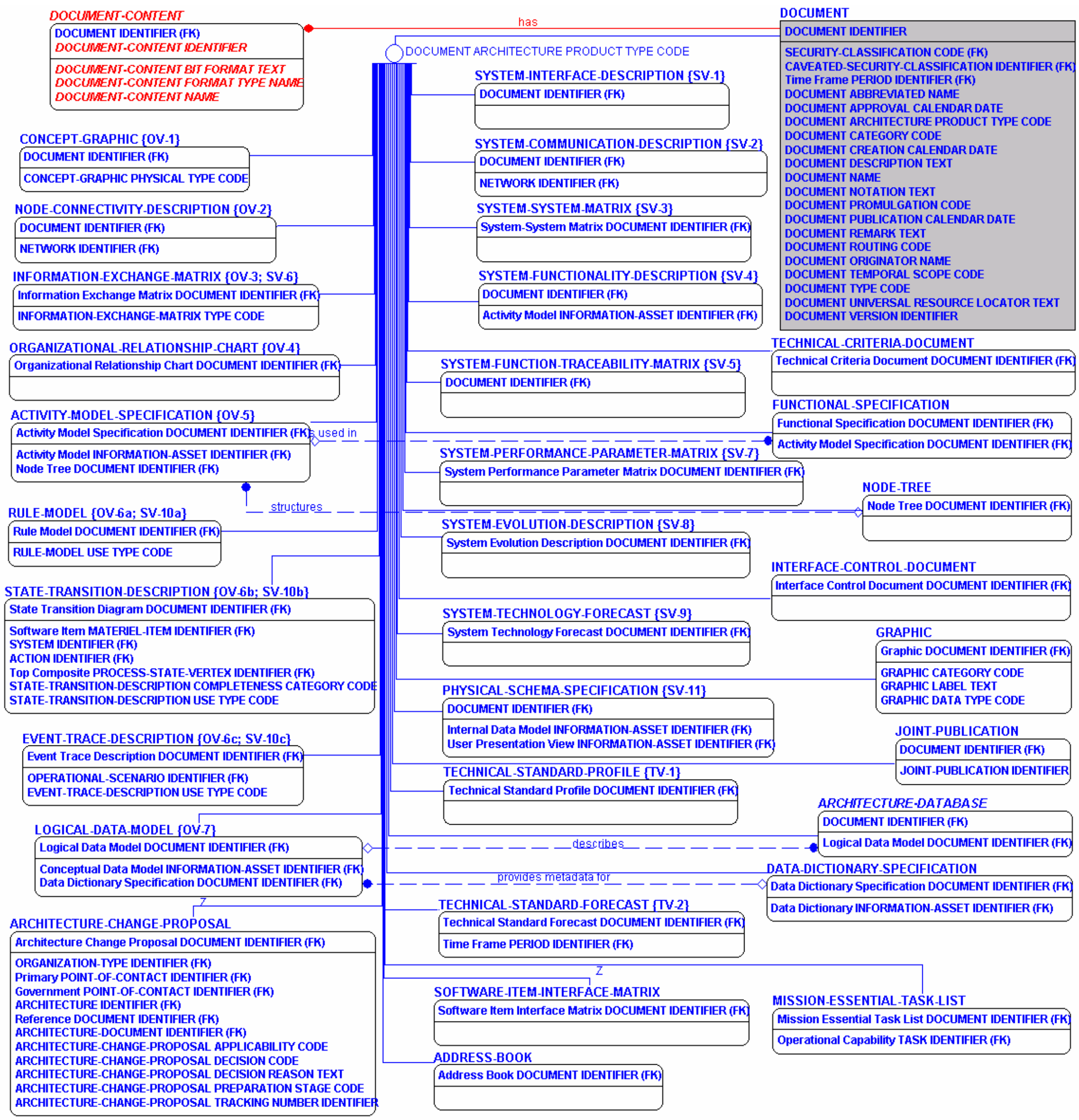


Figure 2-5. CADM Distinctions for Architecture Products

CADM Data Model Diagram Notation. As illustrated in Figure 2-5, boxes represent entities for which architecture data are collected (representing tables when used for a relational database); they are depicted by open boxes with square corners (independent entities) or rounded corners (dependent entities). The entity name is outside and on top of the open box. The lines of text inside the box denote the attributes of that entity (representing columns in the entity table when used for a relational database). The horizontal line in each box separates the primary key attributes (used to find unique instances of the entity) from the non-key descriptive attributes. The symbol with a circle and line underneath indicates subtyping, for which all the entities connected below are non-overlapping subsets of the entity connected at the top of the symbol. Relationships are represented by dotted (non-identifying) and solid (identifying) relationships in which the child entity (the one nearest the solid dot) has zero, one, or many instances associated to each instance of the parent entity (the other entity connected by the relationship line).

3 ALL-VIEWS PRODUCTS

Two products are defined in the All-Views section, Overview and Summary Information (AV-1) and Integrated Dictionary (AV-2).

3.1 OVERVIEW AND SUMMARY INFORMATION (AV-1)

3.1.1 Overview and Summary Information (AV-1) – Product Description

Product Definition. The Overview and Summary Information provides executive-level summary 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.

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

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 (e.g., 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, and what actions are expected to result. The viewpoint from which the architecture is developed is identified (e.g., planner or decision maker).

Context describes the setting in which the architecture exists. It includes such things as mission, doctrine, relevant goals and vision statements, concepts of operation, scenarios, information assurance context (e.g., types of system data to be protected, such as classified or sensitive but unclassified, 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. (See Universal Reference Resources [URR] section in the Deskbook for examples of authoritative sources.) 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 3-1** shows a representative format for the AV-1 product.

- **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
  - Doctrine, 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

Figure 3-1. Overview and Summary Information (AV-1) - Representative Format

3.1.2 UML Representation

The Unified Modeling Language (UML) tool(s) 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. Documentation should be developed for AV-1, which can be input via a documentation field in a UML modeling tool, if desired.

3.1.3 Overview and Summary Information (AV-1) – Data Element Definitions

Table 3-1 defines the data elements related to the AV-1 product.

Table 3-1. Data Element Definitions for Overview and Summary Information (AV-1)[6]

Data Elements	Attributes	Example Values/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., Naval Strike Warfare)
	Architecture Project Description*	Description of the architecture project (e.g., Naval Strike Warfare architecture 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 organization(s) approving the architecture
	Date Completed*	Date on which 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*		Description of the breadth and depth of architecture and the type of operational activities covered by the architecture
	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 together with relevant dates (e.g., Current as of November 1996; Target for 2010, Target 2007-2014)
	Organizations Involved	Free text field listing organization(s) involved
Tasking		
	Name	Name/identifier of the tasking that created the architecture project
	Source	Source of the tasking (e.g., organization, directive, order)
	Description	Text summary of the tasking for 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 view point
	Description	Text indicating from whose viewpoint the architecture is described (e.g., planner)
Development Effort		
	Effort Identifier	Identifier for the level of effort
	Type	Actual/projected
	Level	Text description of the level of effort (actual or projected)
	Cost	Actual or projected cost

[6] 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.

Data Elements	Attributes	Example Values/Explanation
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., land, sea, air)
	Political Situation	Description of political context (e.g., coalition peace enforcement during civil war/internal conflict) for the architecture
Doctrine, Goals, and Vision		
	Name	Name/identifier of document that contains doctrine, goals, or vision relevant to the architecture
	Type	Doctrine, goals, or vision
	Description	Text summary description of contents or relevance of doctrine, goals, or vision to architecture
Rules, Criteria, or Conventions		
	Name	Name/identifier of document that contains rules, criteria, or conventions
	Type	One of: rules, criteria, or conventions
	Description	Text summary description of contents or applicability of rules, criteria, or conventions to architecture development; may include tailoring information for architecture products
Software Tool(s)*		
	Software Tool Name*	Full name of software tool(s) 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 output, or database access/report conventions for database-based tools
	Hardcopy Reference	Name of reference to the hardcopy document (i.e., name, date, etc.) in which product is included; may include primary 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 was performed or completed
	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

Data Elements	Attributes	Example V alues/Explanation
	View Name*	Name of view included in the architecture (e.g., Joint Air Strike Operational Architecture)
View Contains Product*		
	Architecture View*	Name/identifier of architectural view
	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/identifier
	Architecture Purpose Identifier*	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 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 on development of 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 has Context With Respect to Doctrine, Goals, and Vision		
	Architecture Name	Name/identifier of architecture
	Doctrine, Goals, And Vision Name	Name/identifier of doctrine, goals, or vision document relevant to this architecture
Architecture has Context With Respect to Rules, Criteria, or Conventions		
	Architecture Name	Name/identifier of architecture
	Rules, Criteria, Or Conventions Name	Name/identifier of rules, criteria, or conventions documents relevant to this architecture
Architecture Project has Context With Respect to Tasking		
	Architecture Project Name	Name/identifier of architecture project
	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

Data Elements	Attributes	Example Values/Explanation
	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 of 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
	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

3.1.4 CADM Support for Overview and Summary Information (AV-1)

Each architecture should be stored in a Core Architecture Data Model (CADM) structured database as an instance of ARCHITECTURE. **Figure 3-2** provides a high-level diagram from the CADM showing key entities that are used to store architecture data for AV-1 in a CADM-conformant database. Each view of an ARCHITECTURE is specified as a separate instance of ARCHITECTURE, and the collection of views is related to the overall instance of ARCHITECTURE through populating ARCHITECTURE-ASSOCIATION. When an instance of ARCHITECTURE is a specific view (Operational View [OV], Systems View [SV], or Technical Standards View [TV]), the ARCHITECTURE View Type Code is set to the appropriate value, and the appropriate entity from the following list may be populated: OPERATIONAL-ARCHITECTURE, SYSTEM-ARCHITECTURE, or TECHNICAL-ARCHITECTURE (these tables may be left empty if none of the attributes has a value to be stored). The set of SYSTEM instances addressed in an SV is listed by populating SYSTEM-SYSTEM-ARCHITECTURE (populating this table is optional).

Each architecture product for a specific ARCHITECTURE is specified as an instance of DOCUMENT with an appropriate type code (DOCUMENT Architecture Product Type Code) designating the kind of architecture product. All the architecture products for a specific ARCHITECTURE are listed as instances of ARCHITECTURE-DOCUMENT. In addition, ARCHITECTURE-DOCUMENT can include a reference to the OPERATIONAL-SCENARIO(s) relevant to a specific ARCHITECTURE.

Organizations have several possible roles in a specific ARCHITECTURE. These organizations and their roles are listed in ARCHITECTURE-ORGANIZATION. When a specific ORGANIZATION is the subject of an ARCHITECTURE, the associated instance of ARCHITECTURE-ORGANIZATION is given ARCHITECTURE-ORGANIZATION Role Code = "08" (is the focus of).

Figure 3-2 provides a high-level diagram from the CADM showing key entities that are used to store architecture data for this architecture product in a CADM-conformant database.

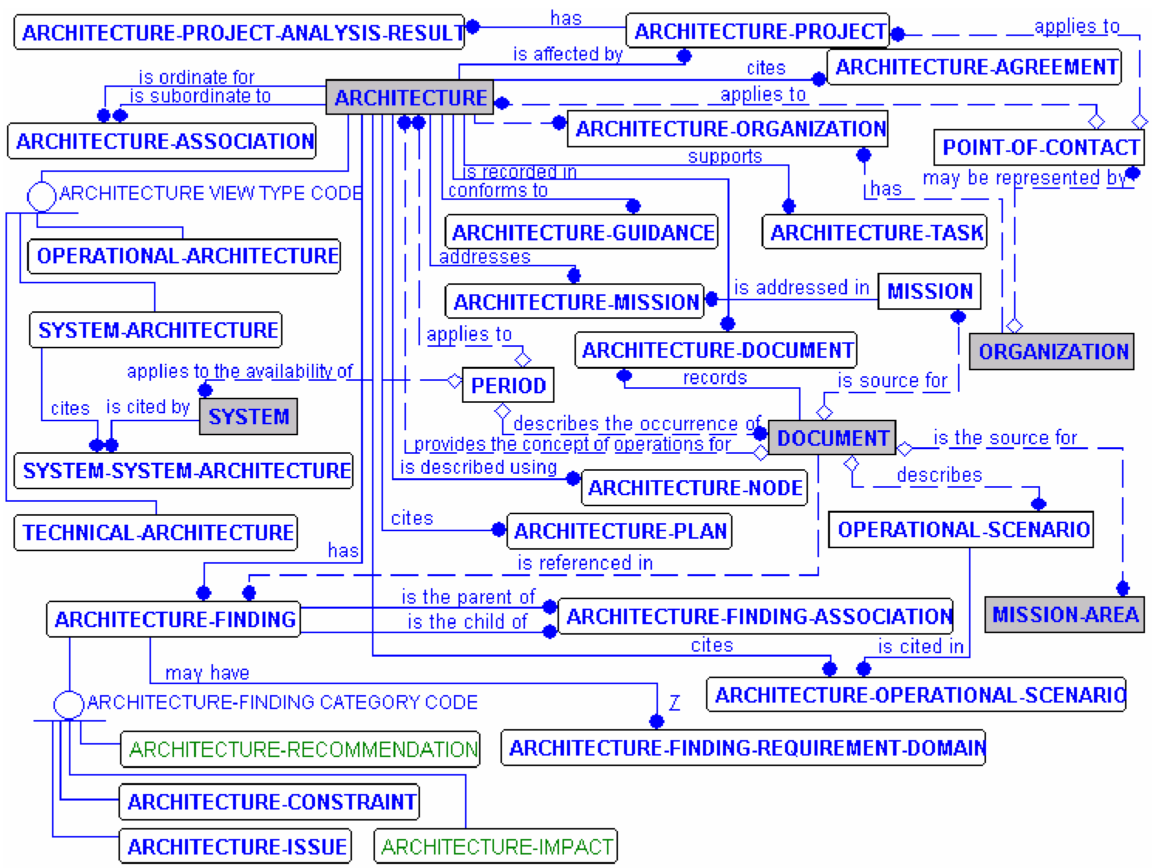


Figure 3-2. CADM ER Diagram for Overview and Summary Information (AV-1)

The association of reference elements other than SYSTEM and ORGANIZATION (noted above) to a specific ARCHITECTURE is identified by populating any of the following as appropriate:

- TASK and ARCHITECTURE-TASK (which can list the elements of the Universal Joint Task List [UJTL] and other recognized task lists germane to a specific ARCHITECTURE)
- GUIDANCE and ARCHITECTURE-GUIDANCE (which can list the information exchange requirements germane to a specific ARCHITECTURE)
- NODE and ARCHITECTURE-NODE (which can list the nodal elements germane to a specific ARCHITECTURE)

Recommendations, constraints, issues, and other types of findings are recorded in ARCHITECTURE-FINDING and its subtypes (e.g., ARCHITECTURE-RECOMMENDATION and ARCHITECTURE-CONSTRAINT). These are related to one another through populating ARCHITECTURE-FINDING-ASSOCIATION. The table ARCHITECTURE-FINDING-REQUIREMENT-DOMAIN can be optionally used to designate any finding as related to one or more of the following: doctrine, training, leadership, organization, materiel, and warfighter (soldier).

3.2 INTEGRATED DICTIONARY (AV-2)

3.2.1 Integrated Dictionary (AV-2) – Product Description

Product Definition. The Integrated Dictionary 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), including 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. In this document, architecture data types are referred to as architecture data elements.

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. The key to long-term interoperability can reside in the accuracy and clarity of these definitions.

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*. A taxonomy of operational nodes applicable to the architecture may be consulted, and the name used for a specific operational node may be chosen from that taxonomy. The AV-2 entry for the node then 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 explanation. The tables form the primary requirements for the CADM. The CADM describes the types and relationships of architecture data in a standard (IDEF1X [FIPS 183, 1993]) form for use in relational, or other, database design and by tool or repository builders. Everything in AV-2 could be stored in a CADM-based repository, just as all Framework architecture products could be stored in a CADM-based architecture modeling tool and/or modeling and repository tool. At a minimum, AV-2 contains the data values for a specific architecture(s), and it is ideally a repository conforming to the CADM.

Architects should use standard terms where possible (i.e., terms from existing, approved dictionaries, glossaries, and lexicons – see Deskbook for URRs). However, when a given architecture is at a lower level of detail than existing architectures or lexicons, or when new concepts are devised for objective architectures, new terms and/or modified definitions of existing terms may be needed. All definitions that originate in existing dictionaries should provide a reference for the source, in addition to providing the definition itself so that architectures are self-contained.

3.2.2 Taxonomies

AV-2 defines the architecture data and their common terms of reference used in creating, maintaining, and using architecture products. The Operational, Systems, and Technical Standards View products are interrelated, sometimes very extensively. Because of this inter-relationship among products and across architecture efforts, it is useful to define 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 DoD 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
- Provides consistency across architectures
- Facilitates 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 that represent Organizations, Organization Types, and Occupational Specialties. The taxonomy minimally consists of names, descriptions, and breakdowns into the parts of the organization, organization type, or human role.
- Operational Activities (or Tasks).[7] The taxonomy minimally consists of names, descriptions, and decomposition into the constituent parts that comprise a process-activity.
- Information Elements. The taxonomy minimally consists of names of information elements exchanged, descriptions, decomposition into constituent parts and subtypes, and mapping to system data elements exchanged.
- 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 consisting of family of systems (FoS), system of systems (SoS), networks of systems, individual systems, and items (e.g., equipment hardware and software). The taxonomy minimally consists of names, descriptions, breakdowns into the constituent parts of the system and categorization of types

[7] Operational Activities defined and standardized by the Joint Staff are in the form of Mission Essential Tasks [CJCSM 3500.04C, 2002]. Operational Activities are also specified (and sometimes standardized) in the form of process activities arising from process modeling. It is sometimes convenient to merge these sets, either as activities or tasks.

of systems. Typing may also address variations across time and systems node installation.

- System Functions. The taxonomy minimally consists of names, descriptions, and decomposition into the constituent parts that comprise a system function.
- Triggers/Events. The taxonomy minimally consists of names, descriptions, and breakdown into constituent parts of the event or trigger and categorization of types of events or triggers.
- Performance Parameters. The taxonomy minimally consists of names, descriptions, units of measure, and conditions that may be applicable to performance parameters.
- Technical Standards. The taxonomy minimally consists of categories of standards (e.g., JTA's Service Areas).
- Technology Areas. The taxonomy minimally consists 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 3-3**. In the table, taxonomy refers to a set of relationships among pairs of instances, often hierarchical. Composition refers to the use of one instance to represent and include as a subset a group of instances. The symbols in the table represent the potential role played by the taxonomy and not by the architecture data elements themselves (e.g., Operational Nodes are important to OV-1, but taxonomies of these nodes are important for products OV-2 through OV-6). 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.

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. (For example, the use of the term *track* refers to very different concepts in the carrier battle group community than in the mine-sweeper community. Yet both of these communities are Navy operational groups and may participate together in littoral warfare task forces.) In these cases, the internal community versions of the architecture products *should* use the vocabulary of the local operational community in order 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.

ARCHITECTURE 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

Figure 3-3. Taxonomies Used in Products

3.2.3 UML Representation

The UML tool(s) being used for analysis and design usually include a data dictionary facility that generates a data dictionary or glossary of terms from the annotated definitions entered by users as they build Framework products. In addition, the collection of products for a specific architecture as stored in the tool comprise the repository of architecture data and the metadata specification.

3.2.4 Integrated Dictionary (AV-2) – Data Element Definitions

As discussed in section 3.2.1, 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 (described in following subsections of this volume) consists 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 into an architecture product). In addition, AV-2 includes a glossary of terms and any taxonomies used. For every architecture data element that is resourced from a pre-defined taxonomy, the AV-2 entry should specify the reference (the source of the taxonomy definition) and the structure.

3.2.5 CADM Support for Integrated Dictionary (AV-2)

AV-2 may consist of a CADM-conformant database or repository, specifically in such reference tables as MATERIEL, MATERIEL-ITEM, ORGANIZATION, ORGANIZATION-TYPE, INFORMATION-ELEMENT, and PROCESS-ACTIVITY that are developed with an architecture or set of architectures. More than one dictionary can be maintained, but only one is needed for each architecture. Each dictionary is stored as an instance of DOCUMENT with a category code designating it as a DATA-DICTIONARY-SPECIFICATION (a subtype of DOCUMENT).

When an architecture database or repository is not provided or deemed adequate, two additional types of data dictionaries are possible:

- An offline dictionary whose terms are stored in a document. Reference to it is stored in DOCUMENT using DOCUMENT Name to store the filename and DOCUMENT Universal Resource Locator Text to store its location.
- An online dictionary whose terms are included in a CADM-structured database. The identity of the dictionary is stored in DATA-DICTIONARY (as an INFORMATION-ASSET); and each term is identified, named, and described as an instance of DATA-DICTIONARY-ELEMENT. Reference to the DATA-DICTIONARY is included in DATA-DICTIONARY-SPECIFICATION through the use of the attribute, Data Dictionary INFORMATION-ASSET Identifier, in that entity.

All architectures using a specific DATA-DICTIONARY-SPECIFICATION are specified by populating ARCHITECTURE-DOCUMENT.

Figure 3-4 provides a high-level diagram from the CADM showing key entities that are used to store architecture data in the form of glossaries, documents, and data dictionaries for this architecture product in a CADM-conformant database.

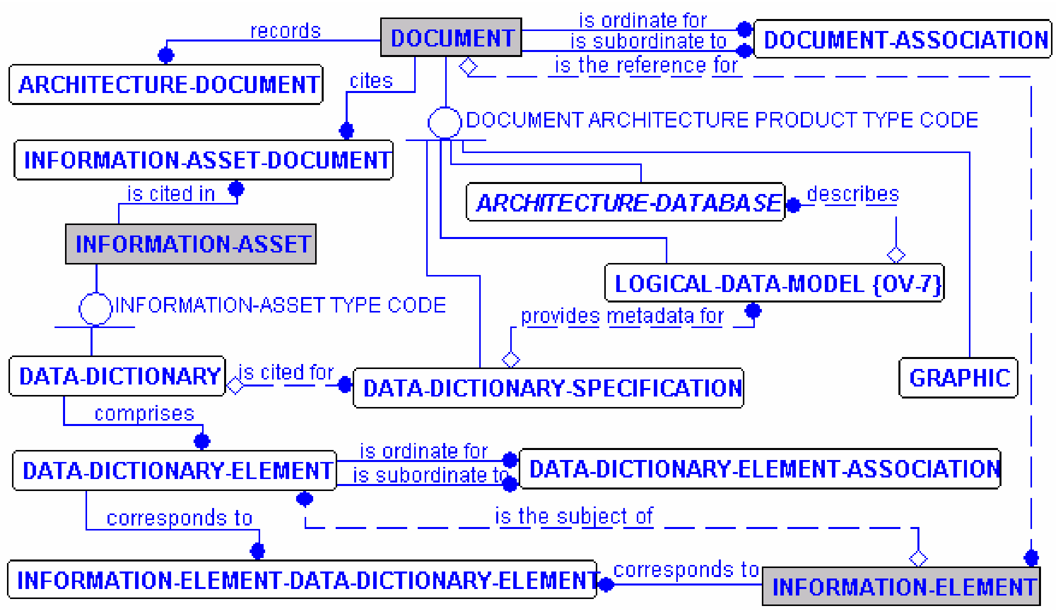


Figure 3-4. CADM ER Diagram for Integrated Dictionary (AV-2)

Taxonomies relate two instances, often hierarchically (tree diagram); they may be viewed as a set of folders and subfolders. The CADM supports one or more taxonomies through the use of a double associative relationship such as the one shown in Figure 3-4 between DATA-DICTIONARY-ELEMENT and DATA-DICTIONARY-ELEMENT-ASSOCIATION. The latter entity identifies all the elements at one level below a specific DATA-DICTIONARY-ELEMENT. The following identifies the CADM structures used to support the taxonomy requirements noted above:

- Operational Nodes:
  - ORGANIZATION-ASSOCIATION and ORGANIZATION-TYPE-ASSOCIATION (for various kinds of operational elements)
  - NODE-ASSOCIATION (for specific nodes, each of which represents an operational unit, command post, command post cell, operational facility, command element, etc.)
- System and Other Physical Nodes:
  - NODE-ASSOCIATION (for specific nodes, each of which represents a platform, system, etc.)
  - MILITARY-PLATFORM-ASSOCIATION
  - FACILITY-ASSOCIATION
  - ORGANIZATION-ASSOCIATION (each organizational unit is assigned a unique ORGANIZATION IDENTIFIER)
  - NODE-ASSOCIATION (when NODE denotes specific location)
  - FEATURE-ASSOCIATION (when NODE denotes specific a geospatial feature)
- Systems:
  - SYSTEM-TYPE-ASSOCIATION (among general classes)
  - SYSTEM-ASSOCIATION (among versions of a SYSTEM)
  - NODE-SYSTEM-ASSOCIATION (among instances of NODE-SYSTEM, each specifying a SYSTEM at a specific NODE)
- Operational (Process) Activities and Tasks:
  - For two instances of PROCESS-ACTIVITY: PROCESS-ACTIVITY-ASSOCIATION
  - For TASK (e.g., in UJTL): TASK-ASSOCIATION (in the Global Information Grid [GIG] and the Department of the Navy Integrated Architecture Database [DIAD]), each TASK corresponds to a unique PROCESS-ACTIVITY using PROCESS-ACTIVITY-TASK
  - For ACTION: ACTION-ASSOCIATION
  - For ACTIVITY-MODEL: ACTIVITY-MODEL-PROCESS-ACTIVITY-ASSOCIATION (e.g., for IDEF0 node trees)

- Information Elements for Information Exchanges:
  - INFORMATION-ELEMENT-ASSOCIATION
  - AGREEMENT-ASSOCIATION (since MESSAGE-STANDARD and STANDARD-TRANSACTION are in subtype hierarchy of AGREEMENT)
  - GUIDANCE-ASSOCIATION (since INFORMATION-REQUIREMENT, INFORMATION-EXCHANGE-REQUIREMENT, EXCHANGE-NEED-LINE-REQUIREMENT, and PROCESS-ACTIVITY-EXCHANGE-REQUIREMENT are all subtypes of INFORMATION-TECHNOLOGY-REQUIREMENT, which in turn, is a subtype of GUIDANCE)
- System Functions:
  - PROCESS-ACTIVITY-ASSOCIATION (in conjunction with SYSTEM-PROCESS-ACTIVITY if a specific system is cited), because each SYSTEM-FUNCTION is a subtype of PROCESS-ACTIVITY
- Triggers/Events:
  - EVENT-ASSOCIATION
- Performance Parameters:
  - CAPABILITY-ASSOCIATION (resources such as organizations, types of organizations, systems, materiel, materiel items, etc., each have capabilities defined by association with CAPABILITY)
- Technical Standards:
  - AGREEMENT-ASSOCIATION (since INFORMATION-TECHNOLOGY-STANDARD is a subtype of AGREEMENT)
  - INFORMATION-TECHNOLOGY-STANDARD-CATEGORY
- Technology Areas:
  - TECHNOLOGY-ASSOCIATION

The following entities provide or otherwise have references in the sense of a relationship to DOCUMENT:

- ACTION-DOCUMENT
- AGREEMENT-DOCUMENT
- ARCHITECTURE-DOCUMENT
- ARCHITECTURE-FINDING
- CAPABILITY-DOCUMENT
- EVENT-DOCUMENT
- FEATURE-DOCUMENT
- GUIDANCE-DOCUMENT
- INFORMATION-ASSET-DOCUMENT
- INFORMATION-ELEMENT

- INFORMATION-ELEMENT-DOCUMENT
- MATERIEL-ITEM-DOCUMENT
- MISSION
- MISSION-AREA
- NETWORK-DOCUMENT
- NODE-ASSOCIATION
- NODE-ASSOCIATION-DOCUMENT
- NODE-ASSOCIATION-SYSTEM-ASSOCIATION-DOCUMENT
- OPERATIONAL-MISSION-THREAD-CAPABILITY
- OPERATIONAL-SCENARIO
- ORGANIZATION-DOCUMENT
- ORGANIZATION-TYPE-DOCUMENT
- PLAN-DOCUMENT
- PROCESS-ACTIVITY-DOCUMENT
- SYSTEM-ASSOCIATION-DOCUMENT
- SYSTEM-DETAIL-INTEROPERABILITY-DOCUMENT
- SYSTEM-DOCUMENT

4 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 materiel independent. However, operations and their relationships may be influenced by new technologies such as collaboration technology, where process improvements are in practice before policy can reflect the new procedures. There may be some cases, as well, in which it is necessary to document the way processes are performed given the restrictions of current systems, in order to examine ways in which new systems could facilitate streamlining the processes. In such cases, an OV may have materiel constraints and requirements that must be addressed. For this reason, it may be necessary to include some high-level Systems View (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)

4.1 HIGH-LEVEL OPERATIONAL CONCEPT GRAPHIC (OV-1)

4.1.1 High-Level Operational Concept Graphic (OV-1) – Product Description

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.

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, and it is intended for presentation to high-level decision makers.

Product Detailed Description. OV-1 consists of a graphical executive summary for a given architecture with accompanying text. The product identifies the mission/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 an idea of the players and operations involved.

The content of OV-1 depends on the scope and intent of the architecture, but 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) and 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 be produced to focus the effort and illustrate its scope. After other products within the architecture's scope have been developed and verified, another version of this product may be produced to reflect adjustments to the scope and other architecture details 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 to present them to high-level decision makers.

OV-1 is the most general of the architecture products and the most flexible in format. Because the format is freeform and variable, no template is shown for this product. However, the product usually consists of one or more graphics (or possibly a movie), as needed, as well as explanatory text. Two graphical examples are shown in **Figure 4-1** and **Figure 4-2**.

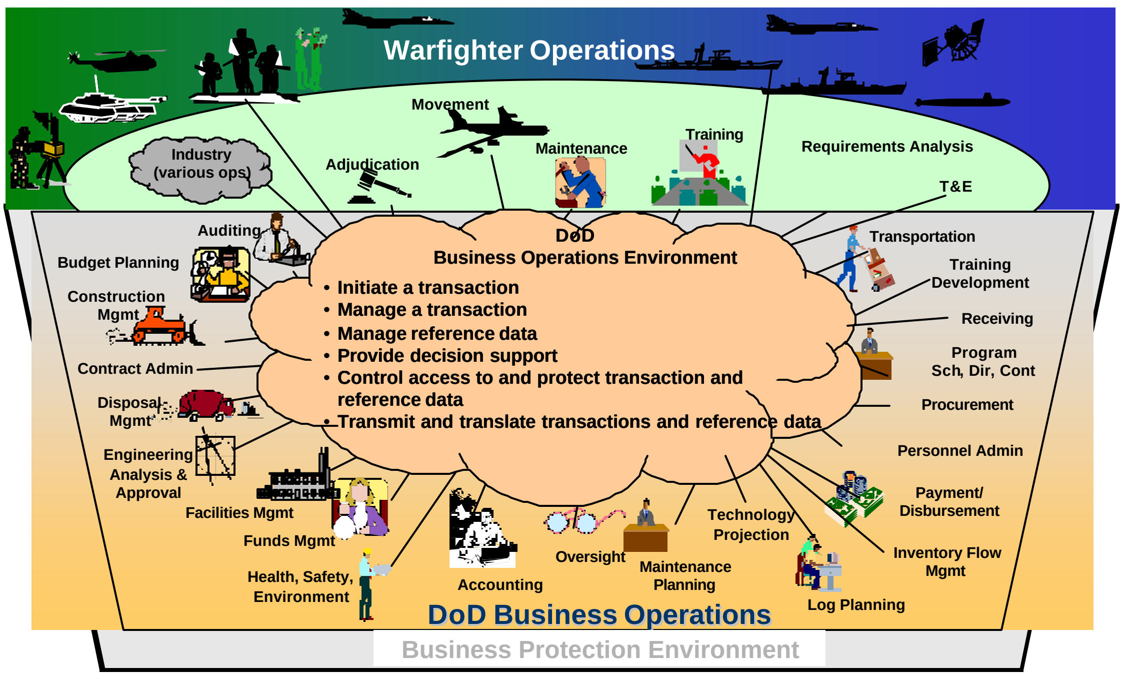


Figure 4-1. DoD Electronic Commerce Concept of Operations (OV-1)

Figure 4-1 illustrates the concept of electronic commerce in the DoD, including the relevant business areas and support to the warfighter [DEB, 2000]. Figure 4-2 illustrates an operational concept for strike [ASN(RDA)CHENG, 2002].

Figure 4-2. Joint Task Force Concept of Operations (OV-1)

4.1.2 UML Representation

It is suggested that OV-1s be constructed in free-format form. If desired, additional, more detailed operational concept diagrams can be constructed using the Unified Modeling Language (UML) use case diagrams. In this situation, several use case diagrams can be constructed, each focusing on a different architecture mission or operational objective. The use case diagrams are further refined to form OV-5.

4.1.3 High-Level Operational Concept Graphic (OV-1) – Data Element Definitions

OV-1 consists of an informal, graphical representation of operations as well as explanatory text. This is a freeform product. However, architecture data elements that appear in an OV-1 are related to others appearing in several OV and SV products. **Table 4-1** lists some of these architecture data elements.

Table 4-1. Data Element Definitions for High-Level Operational Concept Graphic (OV-1)

Data Elements	Attributes	Example Values/Explanation
Graphical Box Types		Graphical Box Types
Asset Icon*		
	Name*	The generic asset name that appears on the graphic (e.g., AWACS, fighter squadron, and carrier battle group)
	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., open ocean, littoral, air, space, land, or specific land location or installation)

Data Elements	Attributes	Example Values/Explanation
Target Area		A geospatial feature of operational interest that is represented graphically in an architecture product
	Identifier	Label on 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, etc.)
	Description*	A description of the kind of line represented
	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 asset 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 con
	Target Area Name	Name/identifier of target area associated with asset icon
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

4.1.4 CADM Support for High-Level Operational Concept Graphic (OV-1)

OV-1 is stored as an instance of DOCUMENT with a category code designating it as a CONCEPT-GRAPHIC (a subtype of DOCUMENT). Each graphical element in that diagram can be represented by an instance of GRAPHIC (also a subtype of DOCUMENT). The set of graphics is represented by a single instance of DOCUMENT with the associated elements identified through the use of DOCUMENT-ASSOCIATION.

Tables in a CADM-structured database containing reference operational information that are candidates for representation in a CONCEPT-GRAPHIC include the following:

- Information exchange requirements (IERs) and their underlying needlines and information content, populated by use of INFORMATION-EXCHANGE-REQUIREMENT, EXCHANGE-NEED-LINE-REQUIREMENT, PROCESS-ACTIVITY-EXCHANGE-REQUIREMENT, and INFORMATION-REQUIREMENT (these are all subtypes of INFORMATION-TECHNOLOGY-REQUIREMENT, which is, in turn, a subtype of GUIDANCE) and related to the CONCEPT-GRAPHIC through populating GUIDANCE-DOCUMENT
- Data dictionaries, activity models, and other information management objects (stored as instances of appropriate subtypes of INFORMATION-ASSET), which are related to the CONCEPT-GRAPHIC through populating INFORMATION-ASSET-DOCUMENT
- Use of NODE and NODE-ASSOCIATION and their associations to entities such as the following: ORGANIZATION, ORGANIZATION-TYPE, MISSION, SYSTEM, and TASK

DOCUMENT-ASSOCIATION is used to identify various instances of DOCUMENT related to a specific CONCEPT-GRAPHIC, to include capstone requirements documents, concepts of operations, guidance documents, and joint publications.

ORGANIZATION-TYPE-ESTABLISHMENT identifies what is commonly known as a table of organization and equipment and may be populated where required to support details for an OV-1 (e.g., major equipment items).

Figure 4-3 provides a high-level diagram from the CADM showing key entities that are used to store architecture data for this architecture product in a CADM-conformant database.

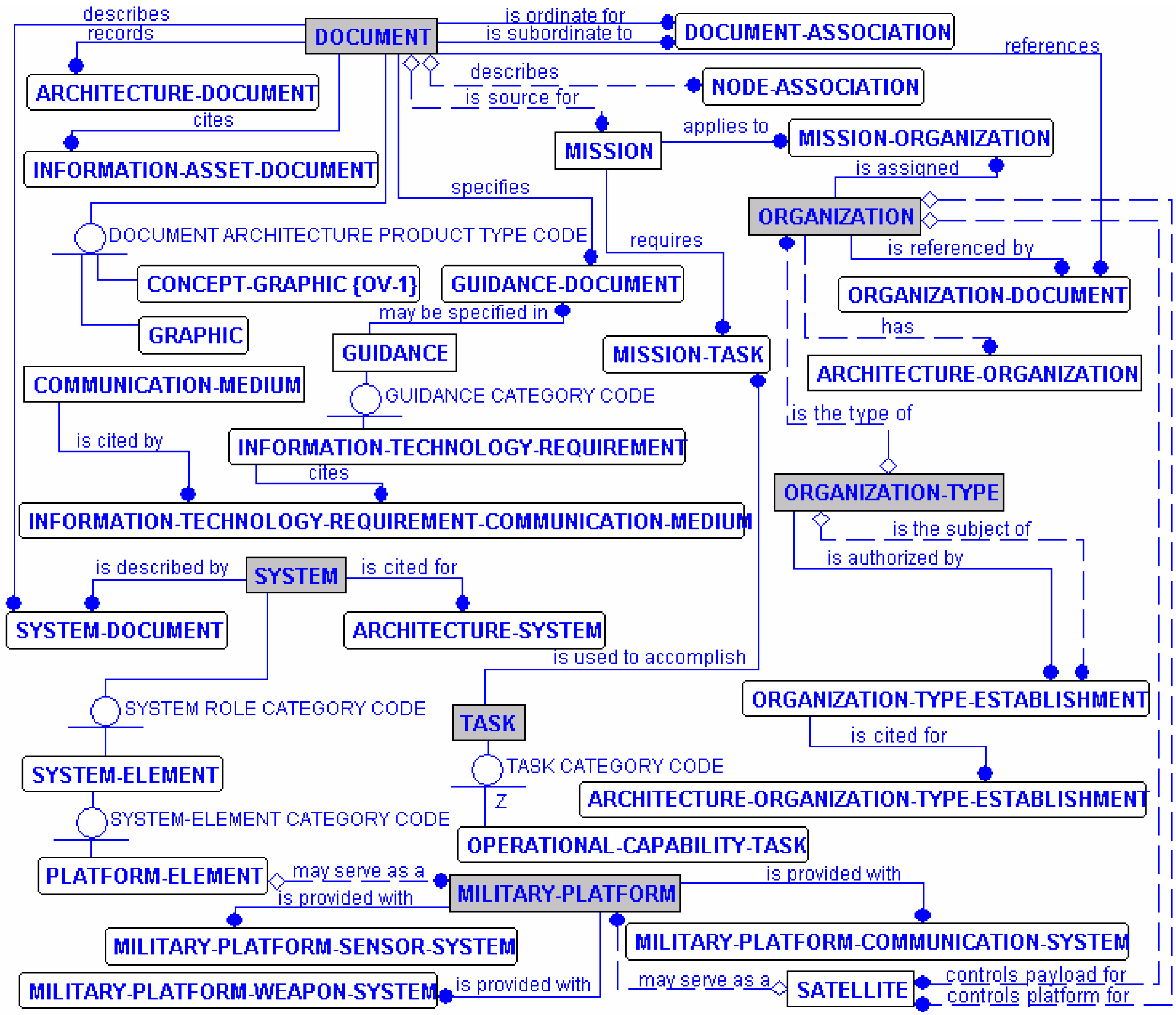


Figure 4-3. CADM ER Diagram for High-Level Operational Concept Graphic (OV-1)

4.2 OPERATIONAL NODE CONNECTIVITY DESCRIPTION (OV-2)

4.2.1 Operational Node Connectivity Description (OV-2) – Product Description

Product Definition. The Operational Node Connectivity Description 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.

Product Purpose. OV-2 is intended to track 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.

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.

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, including, but not limited to, representing an operational/human role (e.g., Air Operations Commander), an organization (e.g., Office of the Secretary of Defense) or organization type, i.e., a logical or functional grouping (e.g., Logistics Node, Intelligence Node), and so on. The notion of operational node will also vary depending on the level of detail addressed by the architecture effort.

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, 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). Furthermore, the needline systems equivalent is the interface line depicted in SV-1. The actual implementation of an interface may take more than one form and 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 (indicating the direction of information flow) and 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.

There is a one-to-many relationship from needlines to information exchanges (e.g., 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 example, the needline represents a number of information exchanges, consisting of various types of reports (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 (i.e., operational nodes that are not strictly within the scope of the subject architecture but that act as important sources of information required by nodes within the architecture or important destinations for information provided by nodes within the architecture).

Operational Activities. The operational activities (from the OV-5 Operational Activity Model) performed by a given node may be listed 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 first-order attention on operational *activities* and only second-order attention on nodes, which can be shown as annotations on the activities.

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 together.

OVs usually avoid representing real physical facilities as operational nodes and focus on virtual or logical nodes that can be based on operational (human) roles or missions. Operational nodes are independent of materiel considerations; indeed, they exist to fulfill the missions of the enterprise and to perform its tasks and activities (business processes, procedures, and operational functions). Use of operational nodes supports analysis and design by separating business process modeling and information requirements from the materiel solutions that support them. Similarly, tasks and activities are organized, and communities of interest are defined to suit the mission and process requirements; the materiel is flexibly and automatically configurable to support the operational processes. However, an OV often has materiel constraints and requirements that must be addressed. Where appropriate, system or physical nodes that constitute the location of an operational node may augment the description of an operational node. These are often taken as recommendations or boundaries for further SV details. **Figure 4-4** provides a template for OV-2.

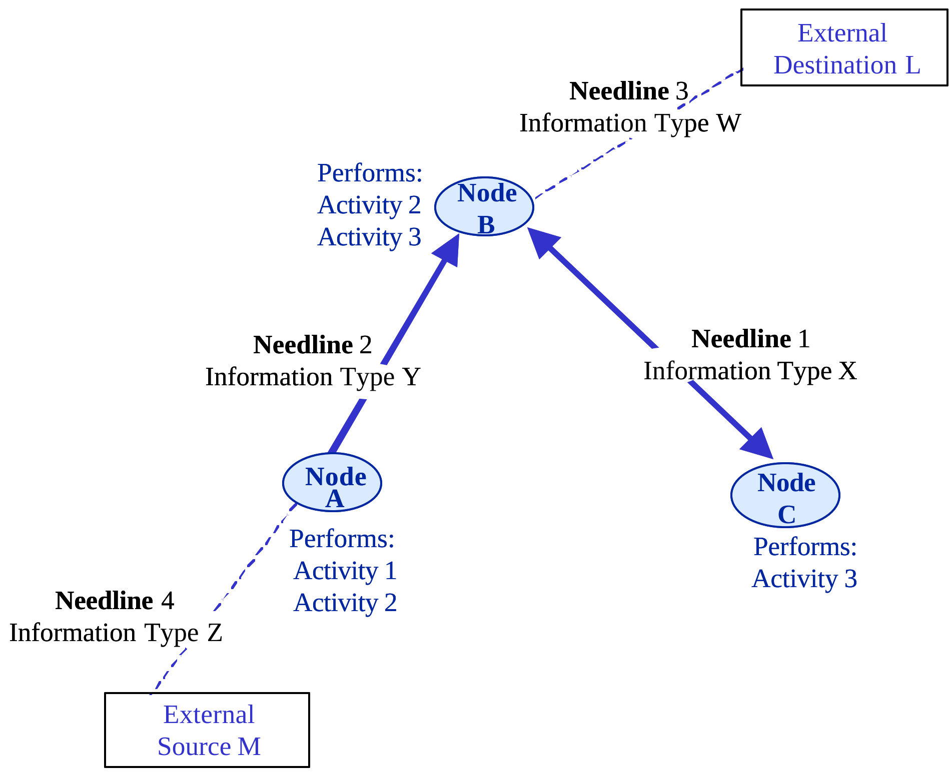


Figure 4-4. Operational Node Connectivity Description (OV-2) Template

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 being provided. For example, a communications service provider will describe needlines in terms of the type of information to be transferred, with what 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, but any given subscriber may only deal with a subset of this information. **Figure 4-5** is a notional example of 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.

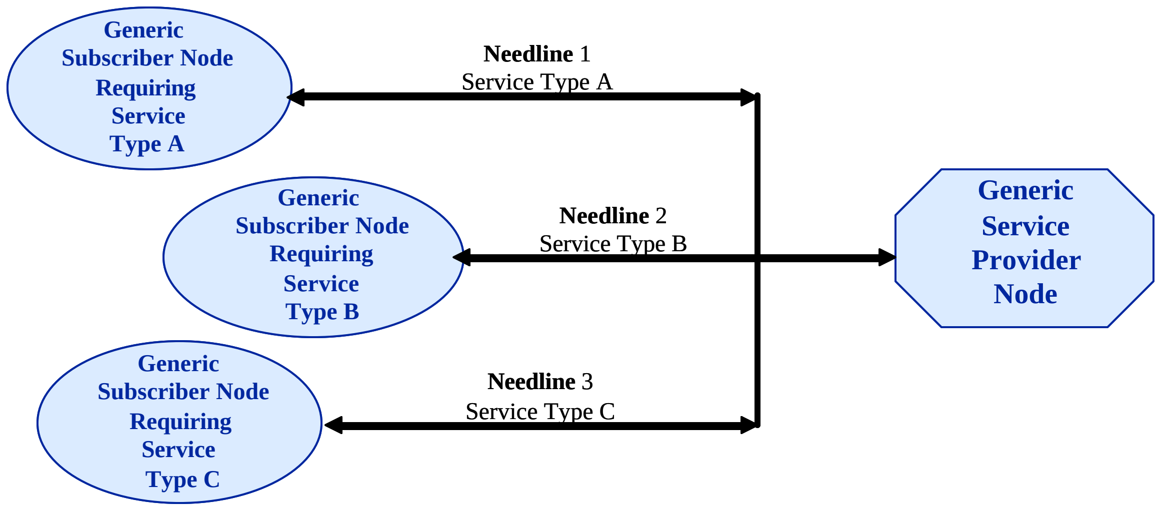


Figure 4-5. Notional Example of an OV-2 Depicting Service Providers

4.2.2 UML Representation

OV-2, describing operational (logical, organization, role) nodes, may be developed in UML using collaboration diagrams. This application of a collaboration diagram does not represent a realization of a use case. The actors represent operational nodes, and they are the same actors appearing on an OV-5 use case diagram. These operational nodes may be displayed 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 (i.e., the use cases). OV-2's corresponding collaboration diagrams emphasize who (i.e., the actors) and what (i.e., the UML links [needlines] and UML messages [information exchanges] exchanged between them). **Figure 4-6** is an example of such a diagram.

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. Such a product may be modeled 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).

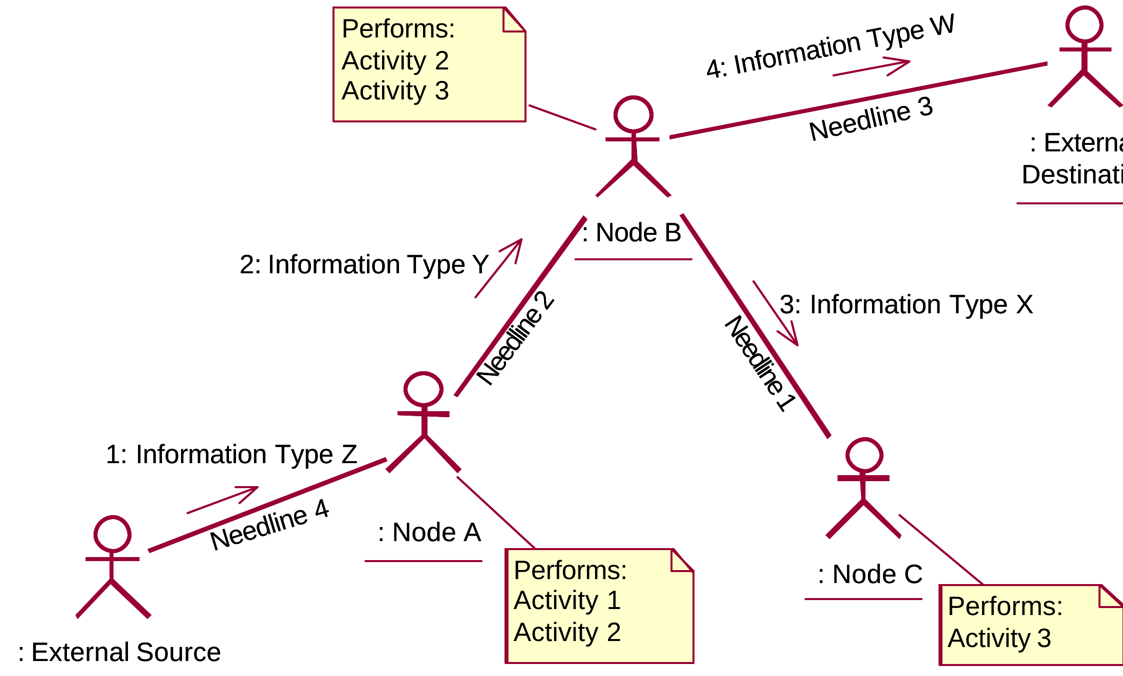


Figure 4-6. UML Operational Node Connectivity Description (OV-2) Template

4.2.3 Operational Node Connectivity Description (OV-2) – Data Element Definitions

OV-2 focuses on the operational nodes, the needlines between them, and the type of information associated with the needline. Associated operational activities may also be noted. **Table 4-2** describes the architecture data elements for OV-2.

Table 4-2. Data Element Definitions for Operational Node Connectivity Description (OV-2)[8]

Data Elements	Attributes	Example Values/Explanation
Graphical Box Types		
Operational Node*		
	Name*	Name or label of node box on diagram
	Description	Text description of mission or role being performed by the node
	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 (if any)
	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 type(s) of information associated with the needline
	Description	Text description of needline

8 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.

	From Operational Node*	**Name of 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)**
	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, the source node must be 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 child (i.e., subordinate) node**
Referenced Types		
Operational Activity		**See OV-5 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 operational node**
Operational Node Represents Organization Type		
	Operational Node Name	**Name/identifier of operational node representing the organization type**
	Organization Type Name	**Name/identifier of organization type represented by operational 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**
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**

4.2.4 CADM Support for Operational Node Connectivity Description (OV-2)

OV-2 is stored as an instance of DOCUMENT with a category code designating it as a NODE-CONNECTIVITY-DESCRIPTION (a subtype of DOCUMENT). Each graphical element in that diagram can be represented by an instance of GRAPHIC (also a subtype of DOCUMENT). Organizational elements are stored as instances of ORGANIZATION, ORGANIZATION-TYPE, and OPERATIONAL-ROLE. Categories (subtypes) of ORGANIZATION-TYPE include OPERATIONAL-FACILITY and OPERATIONAL-ELEMENT.

The entities NODE, NODE-ASSOCIATION, NODE-LINK (a subtype of NODE-ASSOCIATION), and NETWORK (which may include subnetworks) store the OV-2 nodal elements. NETWORK-NODE identifies each NODE of each NETWORK, and it may be used to permit one NODE to represent an entire NETWORK (when OV-2s have drill-down capabilities with multiple views). NODE-ASSOCIATION-NETWORK identifies each of the NODE-ASSOCIATION instances, to include

NODE-LINK instances, for each NETWORK. The NODE instances in a specific ARCHITECTURE may be listed as instances of ARCHITECTURE-NODE. Quantifiable capability requirements of a NODE-LINK may be specified in NODE-LINK-CAPABILITY.[9]

Nodes may represent any or all of the following operational elements:[10]

- ORGANIZATION by use of NODE-ORGANIZATION
- ORGANIZATION-TYPE by use of NODE-ORGANIZATION-TYPE
- OPERATIONAL-ROLE by use of NODE-OPERATIONAL-ROLE and by characterizing the OPERATIONAL-NODE, where appropriate, by reference to instances OCCUPATIONAL-SPECIALTY, POSITION, PERSON-TYPE, or SKILL
- PROCESS-ACTIVITY by use of NODE-PROCESS-ACTIVITY
- TASK by use of NODE-TASK (instances of TASK and PROCESS-ACTIVITY are related by populating PROCESS-ACTIVITY-TASK)
- NETWORK (e.g., a communications network cited for OV-2) by use of NETWORK-NODE
- SYSTEM (e.g., a COMMUNICATION-SYSTEM cited for OV-2) by use of NODE-SYSTEM
- DATA-ITEM-TYPE (e.g., a class of information being exchanged) by use of NODE-DATA-ITEM-TYPE

Operational requirements may be represented in any or all of the following ways:

- Information exchange requirements (IERs) use of INFORMATION-EXCHANGE-REQUIREMENT, INFORMATION-TECHNOLOGY-REQUIREMENT, NODE-ASSOCIATION (together with NODE-LINK where applicable), and NODE-ASSOCIATION-INFORMATION-TECHNOLOGY-REQUIREMENT
- Needlines (or groups of needlines) underlying IERs (or potential IERs) by use of EXCHANGE-NEED-LINE-REQUIREMENT, INFORMATION-TECHNOLOGY-REQUIREMENT, NODE-ASSOCIATION (together with NODE-LINK where applicable), and NODE-ASSOCIATION-INFORMATION-TECHNOLOGY-REQUIREMENT

---

[9] Because the measures being quantified are stored in CAPABILITY and because a parallel structure exists in the CADM for SYSTEM, MATERIEL-ITEM, and other entities, the CADM provides a means to use the same measures to specify what each SYSTEM, MATERIEL-ITEM, or other object can provide to meet the stated capability requirements. Entities used for this purpose include: SYSTEM-CAPABILITY, MATERIEL-ITEM-CAPABILITY-NORM, NETWORK-CAPABILITY, and REQUIRED-INFORMATION-TECHNOLOGY-CAPABILITY.

[10] Not listed here or in the accompanying table and figure are the following entities from the CADM that can additionally be used to characterize what a node represents: NODE-COMMUNICATION-MEDIUM, NODE-ACTIVITY-MODEL-INFO-ELEMENT-ROLE, NODE-ACTIVITY-MODEL-THREAD, NODE-FACILITY, NODE-INFORMATION-ASSET, NODE-INTERNET-ADDRESS, NODE-MATERIEL, NODE-MISSION-AREA, and NODE-PROCESS-ACTIVITY.

- Information content of IERs (or potential IERs) by use of INFORMATION-REQUIREMENT, INFORMATION-TECHNOLOGY-REQUIREMENT, NODE-ASSOCIATION (together with NODE-LINK where applicable), and NODE-ASSOCIATION-INFORMATION-TECHNOLOGY-REQUIREMENT
- Requirements for Exchanging Information between Operational Activities by use of EXCHANGE-NEED-LINE-REQUIREMENT-REQUIREMENT, which references a Source and Destination PROCESS-ACTIVITY
- Communication media by use of COMMUNICATION-MEDIUM, NODE-LINK-COMMUNICATION-MEDIUM, and INFORMATION-TECHNOLOGY-REQUIREMENT-COMMUNICATION-MEDIUM
- Capability by use of OPERATIONAL-CAPABILITY-TASK, a subtype of TASK

Potentially, an OV-2 can apply to more than one ARCHITECTURE. All of the architectures to which OV-2 applies are identified in ARCHITECTURE-DOCUMENT.

The relationship of OV-2 to other architecture products (such as the CONCEPT-DIAGRAM [OV-1], INFORMATION-EXCHANGE-MATRIX [OV-3] or ACTIVITY-MODEL-SPECIFICATION [OV-5]) is specified by populating DOCUMENT-ASSOCIATION, since these are all instances of DOCUMENT. **Figure 4-7** provides a high-level diagram from the CADM showing key entities that are used to store architecture data for OV-2 in a CADM-conformant database.

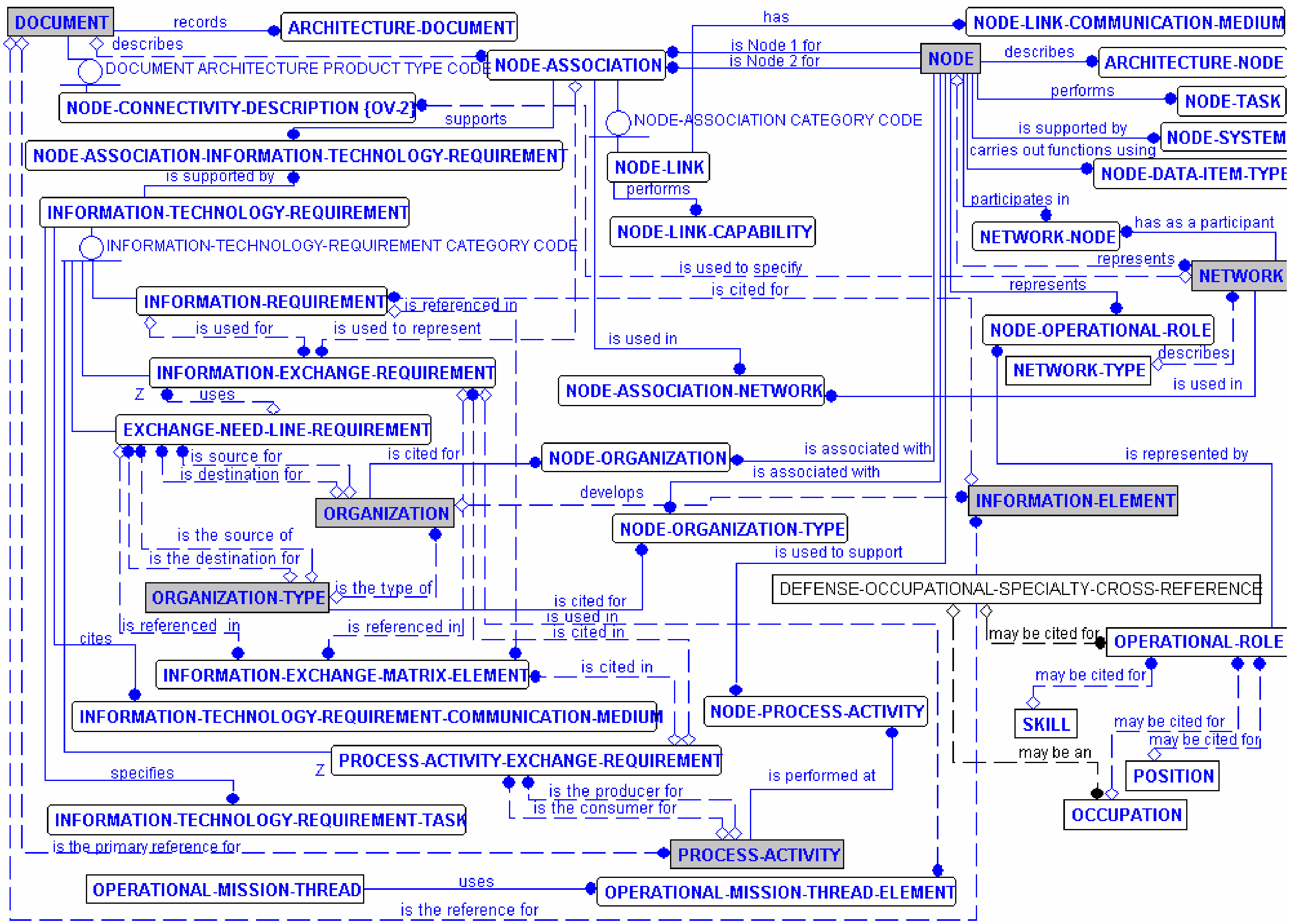


Figure 4-7. CADM ER Diagram for Operational Node Connectivity Description (OV-2)

4.3 OPERATIONAL INFORMATION EXCHANGE MATRIX (OV-3)

4.3.1 Operational Information Exchange Matrix (OV-3) – Product Description

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." [CJCSI 6212.01B, 2000] There is not a one-to-one mapping of OV-3 information exchanges to OV-2 needlines; rather, many individual information exchanges may be associated with one needline.

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. For example, if the subject architecture concerns tactical battlefield targeting, then the timeliness of the enemy target information is a significant attribute of the information exchange.

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 and 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 act, including the information element that needs to be exchanged and the attributes associated with the information element (e.g., Scope), as well as 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 such as Information Assurance attributes. The specific attributes of the information elements included are dependent on the objectives of the specific architecture effort and include the information scope, accuracy, and language. An information element may be used in one or more information exchanges.

The specific mission/scenario and the related task from the Universal Joint Task List (UJTL) or Mission Essential Task List (METL) can be noted as an attribute of the information exchange. A mission/scenario may be selected from the Joint Mission Areas. UJTL and METL tasks are related to a specific scenario that requires the information.

The emphasis in this product is on the logical and operational characteristics of the information. 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. Rather, this product is intended to capture the most important aspects of selected information exchanges.

A partial format for OV-3 matrix can be found in CJCSI 6212.01B, and that format is required for C4ISP development. However, additions to the CJCSI 6212.01B matrix to meet program-unique needs should also be allowed. **Figure 4-8** shows a representative format for the OV-3 Operational Information Exchange Matrix.

Needline Identifier	Information Exchange Identifier	Information Element Description					Producer		Consumer	
		Information Element Name and Identifier	Content	Scope	Accuracy	Language	Sending Op Node Name and Identifier	Sending Op Activity Name and Identifier	Receiving Op Node Name and Identifier	Receiving Op Activity Name and Identifier

Needline Identifier	Information Exchange Identifier	Nature of Transaction					Performance Attributes		Information Assurance					Security			
		Mission/Scenario UJTL or METL	Transaction Type	Triggering Event	Interoperability Level Required	Criticality	Periodicity	Timeliness	Access Control	Availability	Confidentiality	Dissemination Control	Integrity	Accountability	Protection (Type Name, Duration, Date)	Classification	Classification Caveat

Figure 4-8. Operational Information Exchange Matrix (OV-3) – Template

In Figure 4-8, each information exchange is associated with the needline it helps satisfy. There may be many individual exchanges that collectively satisfy a single needline. Note also 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 show 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 this 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 a Logical Data Model (OV-7).

4.3.2 UML Representation

There is no equivalent diagram to this 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 is 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 is used to create 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 possible in the UML tool. As a result, a script may be developed that will automatically generate the matrix.

4.3.3 Operational Information Exchange Matrix (OV-3) – Data Element Definitions

Figure 4-8 illustrates how the end product should look to the user. **Table 4-3** describes the architecture data elements for OV-3.

Table 4-3. Data Element Definitions for Operational Information Exchange Matrix (OV-3)

Data Elements	Attributes	Example Values/Explanations
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 produces the information
	Sending Operational Activity Name*	The identity of the operational activity producing the information
	Receiving Operational Node Name*	Name/identifier of the operational node that consumes the information
	Receiving Operational Activity*	The identity of the operational activity consuming the information
	Mission/Scenario UJTL or METL	Joint Mission Area, cross-mission area domain, Universal Joint Task List activity, related specific scenario, or other task-related basis of the architecture
	Transaction Type*	Descriptive field that identifies the type of exchange
	Triggering Event*	Brief textual description of the event(s) that triggers the information exchange
	Interoperability Level Required	Level of Information Systems Interoperability (LISI), or other interoperability measure, required
	Criticality*	The criticality assessment of the information being exchanged in relationship to the mission being performed
	Periodicity*	How often the information exchange occurs; may be an average or a worst case estimate and may include conditions (e.g., wartime or peacetime)
	Timeliness*	Required maximum allowable time of exchange from node to node (in seconds)
	IA- Access Control	The class of mechanisms used to ensure only those authorized can access information
	IA - Availability	The relative level of effort required to be expended to ensure that the information can be accessed
	IA - Confidentiality	The kind of protection required for information to prevent unintended disclosure
	IA - Dissemination Control	The kind of restrictions on receivers of the information based on sensitivity of information
	IA - 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

Data Elements	Attributes	Example Values/Explanations
	Protection Type Name	The Name for the type of protection
	Protection Duration	How long the information must be safeguarded
	Protection Suspense Calendar Date	The calendar date on which the designated level of safeguarding discontinues
	Classification*	Classification code for the information
	Classification Caveat*	A set of restrictions on information of a specific classification; supplements a security classification with information on access, dissemination, and other types of restrictions
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 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 codes of the natural language(s) involved in the information exchange; relevant for multinational 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 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

Data Elements	Attributes	Example Values/Explanations
Information Exchange has Consuming Operational Activity*		
	Operational Activity Name*	Name of the operational activity (at the receiving 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

4.3.4 CADM Support for Operational Information Exchange Matrix (OV-3)

Figure 4-9 provides an overview of the CADM data structures related to each INFORMATION-EXCHANGE-REQUIREMENT and, more generally, to any INFORMATION-TECHNOLOGY-REQUIREMENT. Using these entities, detailed characteristics can be stored for each IER cited in OV-3.

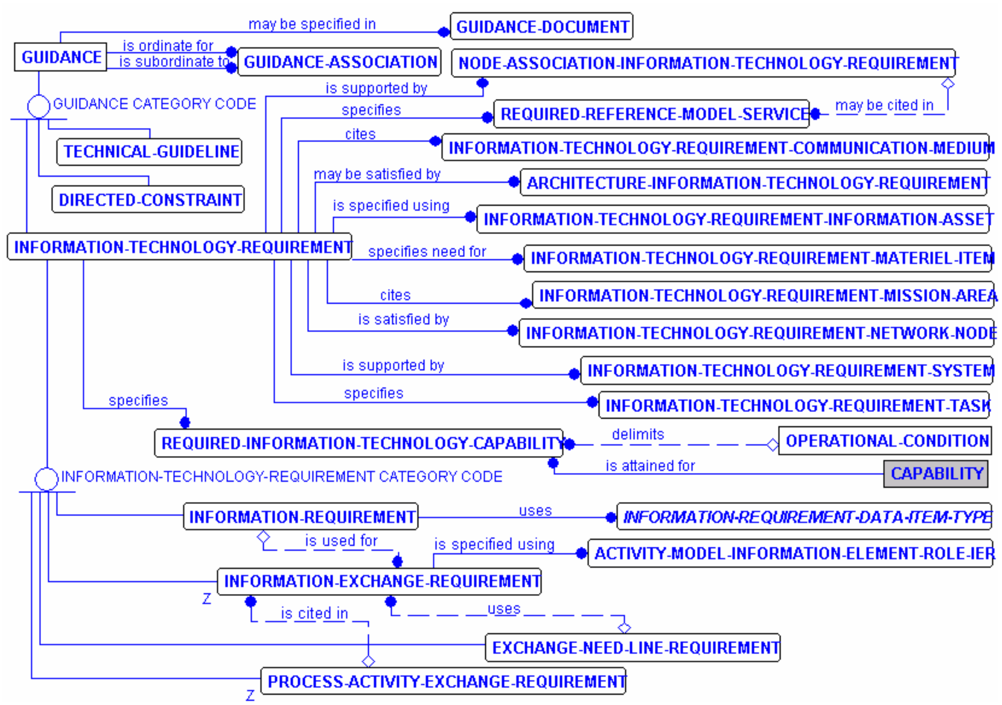


Figure 4-9. CADM ER Diagram for Operational Information Exchange Matrix (OV-3)

OV-3 is stored as an instance of DOCUMENT with a category code designating it as an INFORMATION-EXCHANGE-MATRIX (a subtype of DOCUMENT). Each row of OV-3 is specified as an instance of INFORMATION-EXCHANGE-MATRIX-ELEMENT. The details of row are specified by the following references (foreign keys) contained in INFORMATION-EXCHANGE-MATRIX-ELEMENT:[11]

- COMMUNICATION-MEDIUM Identifier
- Data Item MATERIEL-ITEM Identifier
- Exchange Needline Requirement GUIDANCE Identifier
- Information Exchange Requirement GUIDANCE Identifier
- Information Requirement GUIDANCE Identifier
- Message Standard AGREEMENT Identifier (usually applies only to SV-6)
- Process Activity Exchange Requirement GUIDANCE Identifier
- Time Frame PERIOD Identifier (usually applies only to SV-6)

Using these references, all the attributes of GUIDANCE, INFORMATION-TECHNOLOGY-REQUIREMENT, INFORMATION-EXCHANGE-REQUIREMENT, EXCHANGE-NEEDLINE-REQUIREMENT, PROCESS-ACTIVITY-EXCHANGE-REQUIREMENT, INFORMATION-REQUIREMENT, AGREEMENT, INFORMATION-TECHNOLOGY-STANDARD, and MESSAGE-STANDARD can be inserted into OV-3 from the Architecture Database. In addition, these entities contain references (foreign keys) that allow all of the attributes of the following additional entities to be inserted into OV-3:[12]

- CAVEATED-SECURITY-CLASSIFICATION
- Consumer Battlefield Function TASK
- Consumer DEPLOYMENT-LOCATION-TYPE
- Consumer ORGANIZATION-TYPE (e.g., OPERATIONAL-FACILITY)
- DATA-ITEM-TYPE
- Destination ORGANIZATION
- Destination ORGANIZATION-TYPE
- Destination Resourcing ORGANIZATION
- Destination TASK
- EXCHANGE-RELATIONSHIP-TYPE
- INFORMATION-EXCHANGE-REQUIREMENT-ELEMENT
- INFORMATION-EXCHANGE-REQUIREMENT-ELEMENT-DEPLOYMENT-MISSION-TYPE

---

[11] This entity has additional attributes used for SV-6 Systems Data Exchange Matrix.

[12] References to additional entities are implementation specific and are listed for the Systems Data Exchange Matrix (SV-6).

- INFORMATION-EXCHANGE-REQUIREMENT-ELEMENT-DEPLOYMENT-PHASE
- INFORMATION-EXCHANGE-REQUIREMENT-ELEMENT-PRODUCT
- INFORMATION-ELEMENT
- NODE-ASSOCIATION
- Producer Battlefield Function TASK
- Producer DEPLOYMENT-LOCATION-TYPE
- Producer ORGANIZATION-TYPE (e.g., OPERATIONAL-FACILITY)
- Receiver ORGANIZATION-TYPE (e.g., OPERATIONAL-FACILITY)
- SECURITY-CLASSIFICATION
- Sender ORGANIZATION-TYPE (e.g., OPERATIONAL-FACILITY)
- Source ORGANIZATION
- Source ORGANIZATION-TYPE
- Source Resourcing ORGANIZATION
- Source TASK

The following indicate the ways in which the CADM supports draft IER requirements provided by the Office of the Joint Staff J6 to the Architecture Framework Working Group in January 2003:

- Link to OV-2—Number given to each needline in OV-2; provided by the Exchange NeedLine Requirement GUIDANCE IDENTIFIER in INFORMATION-EXCHANGE-MATRIX-ELEMENT
- Rationale/UJTL Number—Set of joint mission tasks from the UJTL (CJCM 3500.04B) for each mission area; provided by the TASK COMMAND LEVEL CODE and TASK HIERARCHY NUMBER NAME in the entity TASK, which is associated to a specific INFORMATION-EXCHANGE-REQUIREMENT through one or both of the Source TASK IDENTIFIER and Destination TASK IDENTIFIER in INFORMATION-EXCHANGE-REQUIREMENT
- Event—Event that triggers the need for the information exchange; provided textually in INFORMATION-EXCHANGE-REQUIREMENT CAUSAL EVENT TRIGGER TEXT; may be specified more formally using the Operational Rule GUIDANCE IDENTIFIER in INFORMATION-EXCHANGE-REQUIREMENT-TRIGGER-OPERATIONAL-RULE
- Information Characterization—Critical information characteristics that describe what information is being exchanged and how it is to be used; provided textually in GUIDANCE SYNOPSIS TEXT for the Information Requirement GUIDANCE IDENTIFIER in INFORMATION-EXCHANGE-MATRIX-ELEMENT; provided more formally through the use of the attributes of INFORMATION-ELEMENT and the associated INFORMATION-REQUIREMENT
- Send Node—Sending Node; provided by the ORGANIZATION-TYPE NAME and ORGANIZATION-TYPE IDENTIFIER for the Sending ORGANIZATION-TYPE IDENTIFIER in EXCHANGE-NEEDLINE-REQUIREMENT, whose identifier appears

in INFORMATION-EXCHANGE-MATRIX-ELEMENT; may be specified more formally through the Source (Node 1) NODE IDENTIFIER in INFORMATION-EXCHANGE-REQUIREMENT, whose identifier appears in INFORMATION-EXCHANGE-MATRIX-ELEMENT (Note: In some architectures, the Sending Node represents an OPERATIONAL-FACILITY, which is an ORGANIZATION-TYPE in the CADM. In network-centric operations, the Sending Node may represent an information network.)

- Receive Node—Receiving Node; provided by the ORGANIZATION-TYPE NAME and ORGANIZATION-TYPE IDENTIFIER for the Receiving ORGANIZATION-TYPE IDENTIFIER in EXCHANGE-NEEDLINE-REQUIREMENT, whose identifier appears in INFORMATION-EXCHANGE-MATRIX-ELEMENT; may be specified more formally through the Destination (Node 2) NODE IDENTIFIER in INFORMATION-EXCHANGE-REQUIREMENT, whose identifier appears in INFORMATION-EXCHANGE-MATRIX-ELEMENT (Note: In some architectures, the Receiving Node represents an OPERATIONAL-FACILITY, which is an ORGANIZATION-TYPE in the CADM. In network-centric operations, the Receiving Node may represent an information network.)
- Critical—Criticality assessment of the information being exchanged in relationship to the mission; attribute (Information Criticality Code) is formally defined[13] as "The code that represents the seriousness of the benefit that the information exchange element provides to the objective of the action being taken for a specific INFORMATION-EXCHANGE-REQUIREMENT-ASSURANCE." The (DoD-approved) values of this code are the following:
  - 1 = Category 1 Mission Critical (Force C2)—Critical and high-level information (e.g., emergency action message and commander's guidance)
  - 2 = Category 2 Mission Critical (Mission Operations)—Required in support to operations (e.g., joint task force contingency plans and operations plan)
  - 3 = Category 3 Mission Critical (Core Functions)—Ongoing information exchanges (e.g., configuration and guidance information and restricted frequency list)
  - 4 = Mission critical [not otherwise specified]
  - 5 = Mission support—Logistics, transportation, medical (e.g., gallons of petroleum-oil-lubrication scheduled for delivery)
  - 6 = Administrative—Personnel, pay, training, etc. (e.g., change in allotment)
- Format—Description of Data Type; may be specified from a user-defined pick list with DATA-ITEM-TYPE IDENTIFIER in INFORMATION-EXCHANGE-MATRIX-ELEMENT; may be more formally defined using the Message-Standard

[13] Approved under DoD 8320.1 procedures as Standard Data Element 63248 (INFORMATION-EXCHANGE-REQUIREMENT-ASSURANCE INFORMATION CRITICALITY CODE), Version 1, DoD Data Dictionary System, January 2003.

AGREEMENT IDENTIFIER in INFORMATION-EXCHANGE-MATRIX-ELEMENT or using the INFORMATION-REQUIREMENT PHYSICAL FORMAT TYPE CODE[14]

- Interoperability Level Required—Class of technical means intended to be used for information exchange; approved domain values include the following high-level technical means:
  - 01 = Level 0—Isolated Level (Manual), without sublevel distinction
  - 07 = Level 1—Connected Level (Peer-to-Peer), without sublevel distinction
  - 12 = Level 2—Functional Level (Distributed), without sublevel distinction
  - 16 = Level 3—Domain Level (Integrated), without sublevel distinction
  - 20 = Level 4—Enterprise Level (Universal), without sublevel distinction
- Timeliness—Required maximum time from node to node expressed in seconds; may be specified by the INFORMATION-REQUIREMENT Timeliness Code[15] and numerically through the two attributes INFORMATION-EXCHANGE-REQUIREMENT-Timeliness High Quantity and INFORMATION-EXCHANGE-REQUIREMENT-Timeliness Low Quantity
- Access Control—Mechanism used to ensure that only authorized users can access information; specified using INFORMATION-EXCHANGE-REQUIREMENT-ASSURANCE ACCESS CONTROL TYPE CODE, whose approved domain values include the following:
  - 1 = Not Required—No checks of any kind; anybody can access the information or the information system (e.g., access to most World Wide Web sites)
  - 2 = Profile—Access is controlled by assessing whether the individual seeking access displays the characteristics typically required (e.g., a carload of individuals are granted access to a post because they are in uniform and the car has a sticker)
  - 3 = Password and Identification Document—Individual seeking access must be known and provide a predetermined password (e.g., bank ATMs require both the user's card [ID] and the user to enter a personal identification number [PIN] [password])

---

[14] Approved domain values include: 01 = Audio; 02 = Text; 03 = Graphic (To Include Facsimile); 04 = Imagery Not Otherwise Specified; 05 = Imagery, Still; 06 = Imagery, Stop Motion; 07 = Imagery Full Motion; 08 = Data Not Otherwise Specified; 09 = Data, ASCII; 10 = Data, Bit-Oriented; 11 = Face-To-Face Meeting; 12 = Courier; 13 = Film; 14 = Paper; 15 = Magnetic Tape; 16 = Optical Disk; 17 = Magnetic Disk; 18 = Video (Full-Motion Video With Audio); 19 = Visual Not Otherwise Specified.

[15] Approved as Standard Data Element 63267, Version 1, DoD Data Dictionary System, January 2003. Domain values include: 1D = Up to One Day (8 Hour -24 Hours); 1H = Up to One Hour (10 Minutes -1 Hour); 1M = Up to One Month (1 Day -30 Days); 8H = Up to 8 Hours (1 Hour -8 Hours); GM = Greater than One Month; LG = Large (Greater Than 30 Days); M = Moderate (1-10 Seconds); NK = Not Known; NRT = Near-Real-Time (Less Than 1 Second); NS = Not Specified; RT = Real-Time; S = Slow (10 Seconds -10 Minutes).

- 4 = SSL (Secure Socks Layer [Server-Based])
- 5 = ID Cert/ACL—An identification certificate and presence of the identified entity on a valid access control list (ACL)
- 6 = Crypto Ignition Key (CIK)—Key required for secure access (e.g., STU III)
- 7 = Pairwise Key—Source encrypts the information and the destination decrypts the information using symmetric keys

- Protect—Mechanism used to ensure that only authorized users can access information; specified using one or more of the following three information assurance attributes:
  - INFORMATION-EXCHANGE-REQUIREMENT-ASSURANCE PROTECTION DURATION CODE, whose approved domain values include the following:
    - 1 = None
    - 2 = Encrypted for Transmission Only (EFTO)—After transmission is completed, information protection is not required
    - 3 = Originating Agency's Determination Required (OADR)
    - 4 = Specified until explicit expiration date
    - 5 = Specified as end of mission
    - 6 = Specified as a period of time beginning as of the date and time of transmission and ending after an explicitly provided length of time
  - INFORMATION-EXCHANGE-REQUIREMENT-ASSURANCE PROTECTION ELAPSED-TIME QUANTITY (in days)
  - INFORMATION-EXCHANGE-REQUIREMENT-ASSURANCE PROTECTION SUSPENSE CALENDAR DATE
- Classification—Classification of the information; specified by using relationships of INFORMATION-EXCHANGE-REQUIREMENT with two DoD-approved classification entities: SECURITY-CLASSIFICATION and CAVEATED-SECURITY-CLASSIFICATION
- Block Increment—Designated block or increment the IER is required (usually by separate time frames); specified using INFORMATION-TECHNOLOGY-REQUIREMENT BLOCK NAME

Figure 4-10 provides a high-level diagram from the CADM showing key entities that are used to store architecture data for OV-3 in a CADM-conformant database.

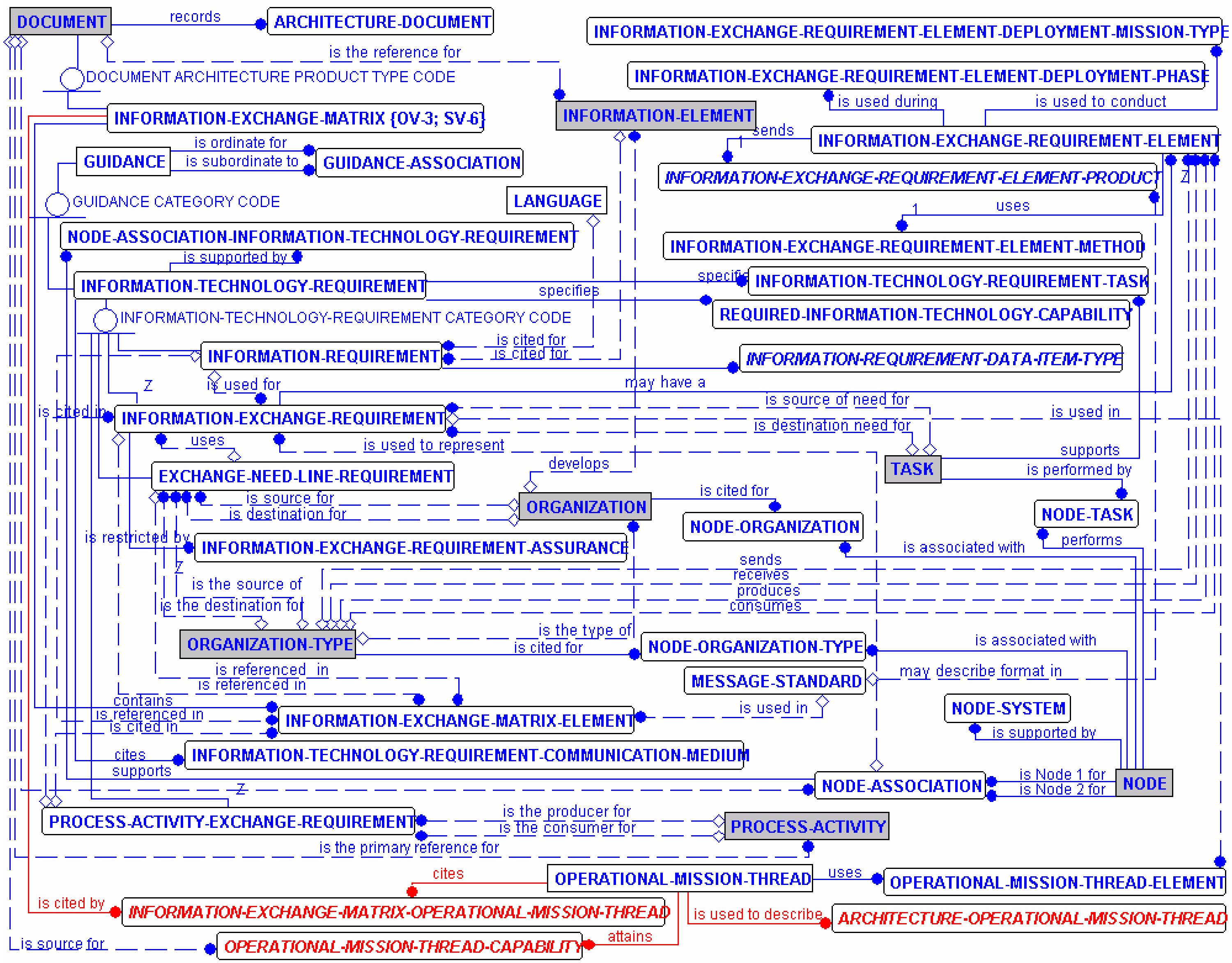


Figure 4-10. CADM ER Diagram for Operational Information Exchange Matrix (OV-3)

4.4 ORGANIZATIONAL RELATIONSHIPS CHART (OV-4)

4.4.1 Organizational Relationships Chart (OV-4) – Product Description

Product Definition. The Organizational Relationships Chart illustrates the command structure or relationships (as opposed to relationships with respect to a business process flow) among human roles, organizations, or organization types that are the key players in an architecture.

Product Purpose. This product clarifies the various relationships that can exist between organizations and sub-organizations within the architecture and between internal and external organizations.

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. Others may be defined 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 the goals of the architecture. For example, dynamic teams or task forces (i.e., new operational nodes) may be created in real time with only limited lifespans and assigned missions, and could have needlines assigned to them. The creating node and the created node have a unique relationship that should be documented. 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 in order to conduct their corresponding operational activities of OV-5.

Human roles whose skills are needed to perform the operational activities or business processes described in the architecture may also be defined in OV-4. The corresponding operational activities should be decomposed to a degree that allows them to be correlated to specific human roles within organizations. In addition, and specifically in the case of target architectures, human roles that do not reflect a specific supervisory reporting, command and control, or coordination organizational structure may be used in OV-4. In this case, OV-4 may be developed using strictly human roles that are the key players in an architecture.

Organizational relationships are important to depict in an OV (for a current architecture), because they can illustrate fundamental human roles (e.g., who or what type of skill is needed to conduct operational activities) as well as management relationships (e.g., command structure or relationship to other key players). Also, organizational relationships may influence how the operational nodes in an OV-2 are connected. A template is shown in **Figure 4-11**.

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.

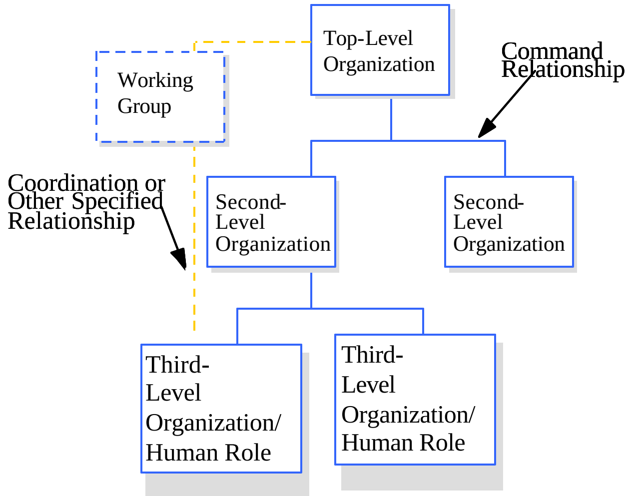


Figure 4-11. Organizational Relationships Chart (OV-4) – Template

4.4.2 UML Representation

A class diagram (using the actor icon) may be used to represent Organizational Relationships Charts. Class relationship notation can be used to show relationships among organizations. **Figure 4-12** illustrates such a diagram.

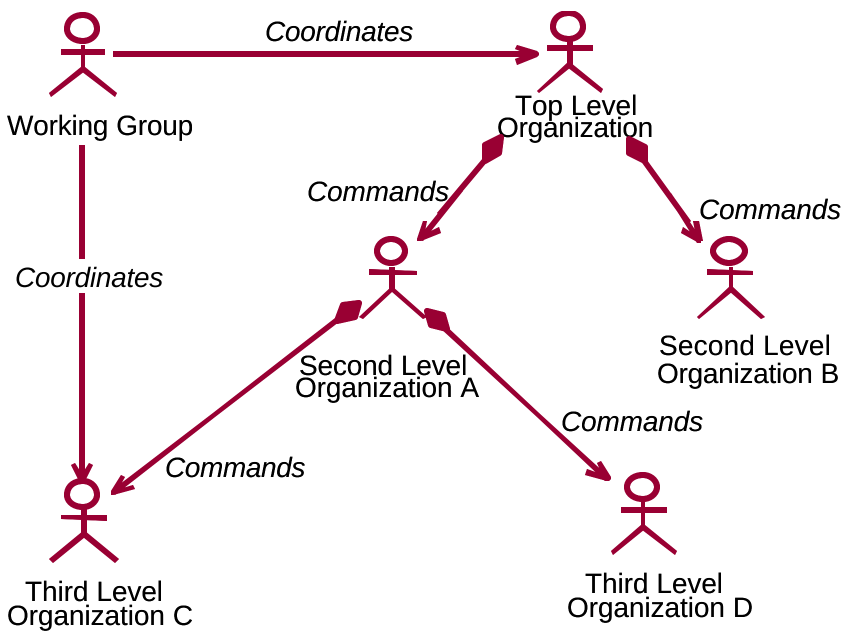


Figure 4-12. UML Organizational Relationships Chart (OV-4) – Template

Classes and their relationships represented in these diagrams trace to the operational nodes (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.

4.4.3 Organizational Relationships Chart (OV-4) – Data Element Definitions

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

Table 4-4. Data Element Definitions for Organizational Relationships Chart (OV-4)

Data Elements	Attributes	Example Values/Explanation
Graphical Box Types		
Organization*		
	Name*	Name of organization that appears on the graphic
	Description*	Text description of the organization's purpose, including the spelling out of all acronyms
	(Military) Service	Army, Navy, Air Force, Marine Corps, Joint
	Code/Symbol	Service office code or symbol
	Role/Responsibility	Text description of the role played by the described organization
Organization Type		
	Name	Name of organization type
	Description	Text description of the organization type
Human Role		
	Name	Name of a human role (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
	Relationship Type*	The type of the 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

4.4.4 CADM Support for Organizational Relationships Chart (OV-4)

Each OV-4 is represented as an instance of ORGANIZATIONAL-RELATIONSHIP-CHART, a subtype of DOCUMENT, which has attributes for Name, Time Frame, Category Code (with domain value Organizational Relationship Chart), etc. The key entities used to specify the architecture data in OV-4 are presented in **Figure 4-13**, a high-level diagram from the CADM showing key entities that are used to store architecture data for OV-4 in a CADM-conformant database.

Each ORGANIZATION or ORGANIZATION-TYPE in OV-4 is related to a NODE. Hierarchically related pairs of nodes are expressed as instances of NODE-HIERARCHY, a subtype of NODE-ASSOCIATION. The specific node pairs for OV-4 are designated as instances of the associative entity ORGANIZATIONAL-NODE-HIERARCHY. NODE-ORGANIZATION and NODE-ORGANIZATION-TYPE are used to identify the organization elements depicted by NODE. Each ORGANIZATION-TYPE is associated with an instance of NODE-ICON, which can be used to enforce consistent graphical representations of an ORGANIZATION or ORGANIZATION-TYPE.

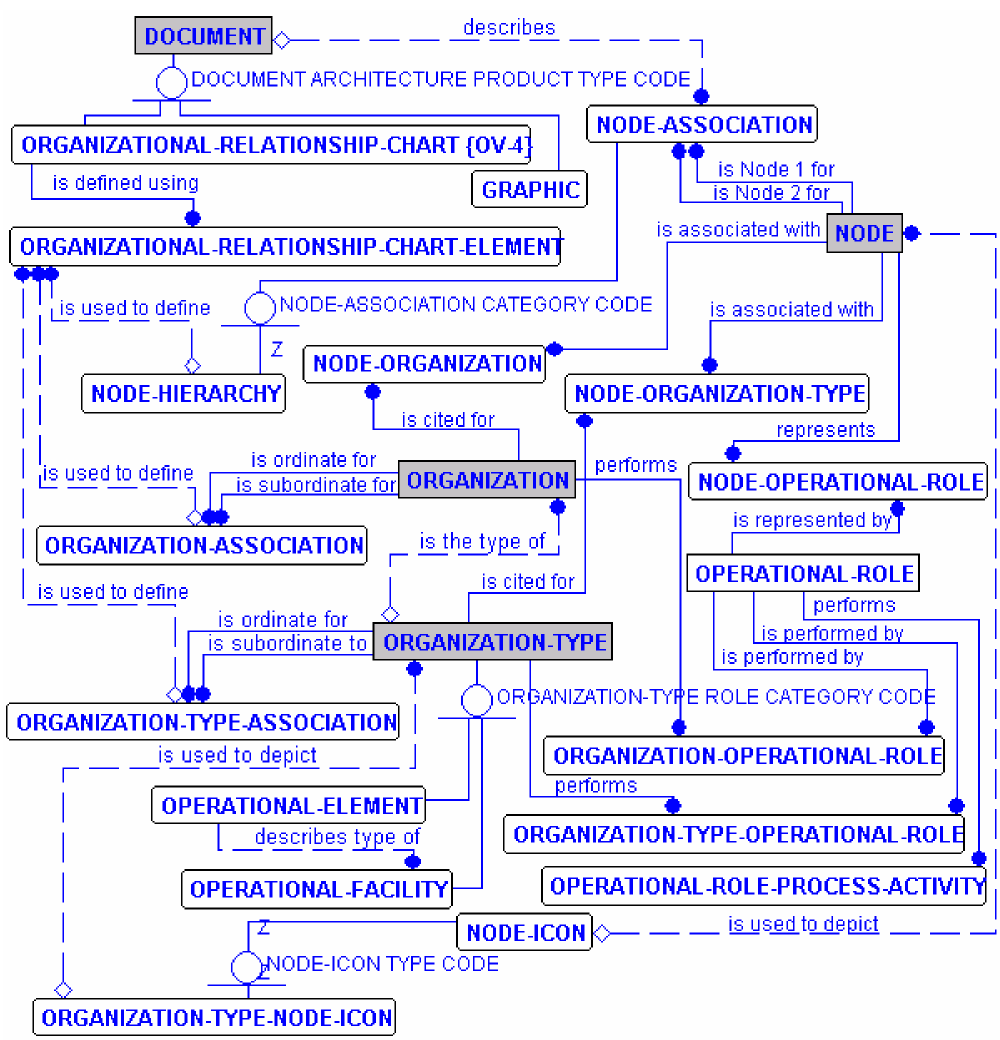


Figure 4-13. CADM ER Diagram for Organizational Relationships Chart (OV-4)

4.5 OPERATIONAL ACTIVITY MODEL (OV-5)

4.5.1 Operational Activity Model (OV-5) – Product Description

Product Definition. The Operational Activity Model 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/from activities that are outside the scope of the architecture. High-level operational activities should trace to (are decompositions of) a Business Area, an Internal Line of Business, and/or a Business Sub-Function as published in OMB's Business Reference Model [OMB, 2003].

Product Purpose. OV-5 can be 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 (information flows among the activities) that need to be scrutinized further
- Provide a necessary foundation for depicting activity sequencing and timing in OV-6a, OV-6b, and OV-6c

Product Detailed Description. OV-5 describes capabilities, operational activities (or tasks), I/O flows between activities, and I/O flows to/from activities that are outside the scope of the architecture.

The Framework does not endorse a specific modeling methodology. However, if the *Integration Definition for Function Modeling* (IDEF0) method [FIPS 183, 1993] is used, the activities also show controls (factors that affect the way that the activity is performed) and may show mechanisms (the resources, including operational nodes, that perform the activity). While some may illustrate corresponding systems as mechanisms in this model, the reader is cautioned 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. I/Os that are produced or consumed by leaf operational activities that cross operational node boundaries are carried by needlines of OV-2. In addition, operational activities can be annotated (e.g., via the mechanism arrow in an IDEF0 diagram) with the corresponding operational node from OV-2.

Annotations to the activities may also identify the costs (actual or estimated) 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 can later be used in OV-3.

OV-5 is a key product for describing capabilities and relating capabilities to mission accomplishment. The DoD Dictionary of Military Terms [DoD JP 1-02, 2001] defines a capability as "the ability to execute a specified course of action. (A capability may or may not be accompanied by an intention.)" A capability can be defined by one or more sequences of activities, referred to as operational threads or scenarios. A capability may be further described in terms of the attributes required to accomplish the set of activities (such as the sequence and timing of operational activities or materiel that enable the capability), in order to achieve a given mission objective. Capability-related attributes may be associated with specific activities or with the information flow between activities, or both. When represented by a set of operational activities, a capability can also be linked to an operational node in an OV-2.

Integrated architectures with Doctrine, Organization, Training, Materiel, Leadership & education, Personnel, and Facilities (DOTMLPF) information provide a structured and organized approach for defining capabilities and understanding the underlying requirements for achieving those capabilities. The full spectrum of DOTMLPF is modeled and related so that analyses and decisions can be supported by describing the sequence and timing of activities; tying them to the operational nodes (representing organizations or human roles); relating them to their supporting systems or system functions; and specifying the actions, events, and related guard conditions or business rules that constrain those activities. Below is a detailed description of how DOTMLPF is tightly weaved into this and related products.

- Doctrine is represented as controls (or guard conditions) in OV-5.
- Organization is represented via the operational nodes of OV-2, which can be shown as mechanisms (or annotations to the activities) in OV-5.
- Training is represented via the operational node, which may represent a human role which in turn, embodies a certain skill set or knowledge domain required to perform the activities, which are, in turn, related to operational activities of OV-5.
- Materiel is tied to the elements in OV-5, where mechanisms may be used to represent systems that support operational activities. In addition, further materiel detail may be related to the activities via SV-5, by relating those activities to the system functions that are executed by systems that automate them (wholly or partially). Each operational thread or scenario (represented by an OV-6c) is associated with a certain capability, since a capability is defined in terms of the activities and their attributes depicted in OV-6c. Consequently, an SV-5 may also be used to relate a capability to the systems that support it, by labeling a set of activities with its associated capability (defined in OV-6c), and by labeling the system functions with the systems that execute them (defined in SV-1, SV-2, and SV-4).
- Leadership may be represented indirectly in OV-5 by first mapping activities to operational nodes via mechanisms and through the relationships of an operational node in OV-2 to organizations, organization types, or leadership human roles in OV-4.
- Personnel may be represented indirectly through the relationships of an operational node in OV-2 to organizations, organization types, or human roles in OV-4.

- Facility is tied to OV-5, because an operational node is directly tied to the systems nodes (facilities) that house the systems, which may be shown as mechanisms that support operational activities.

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

OV-5 is frequently used in conjunction with a process flow model (such as an IDEF3 model, or a UML sequence diagram) that 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, a process flow model may be annotated with the names of the operational nodes responsible for conducting those activities. A process flow model may be described in OV-6c.

The decomposition levels and the amount of detail shown on OV-5 should be aligned with the operational nodes that are responsible for conducting the operational activities (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 4-14** depicts templates for the Operational Activity Hierarchy Chart and one level of a process-flow OV-5.

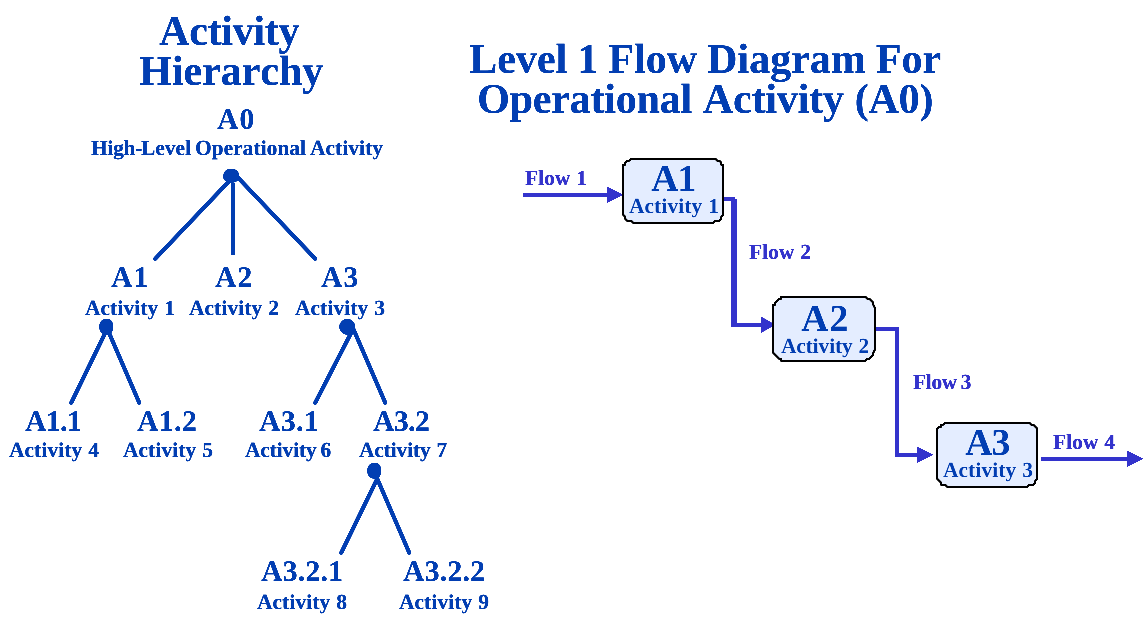


Figure 4-14. Operational Activity Hierarchy Chart and Operational Activity Diagram (OV-5) – Templates

Figure 4-15 is a process-oriented OV-5 template showing how some additional architecture data could be added as annotations to the original template. For example, activities may be annotated with information concerning the operational nodes that conduct them, the materiel that supports them, the cost of conducting the activity, and so forth. (The types of additional architecture data are notional.)

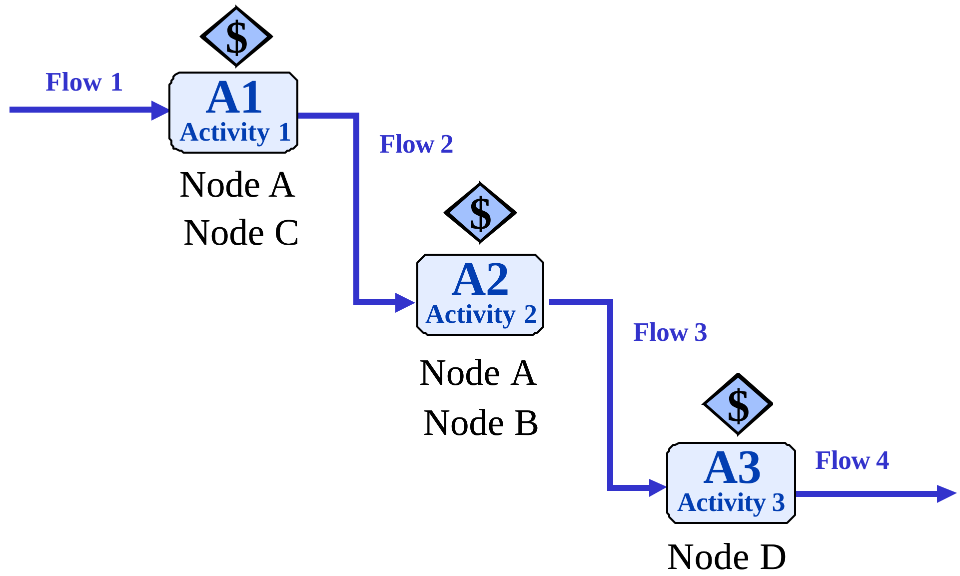


Figure 4-15. Operational Activity Model (OV-5) – Template with Notional Annotations

4.5.2 UML Representation

OV-5 may be documented 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." [OMG, 2001] The Dictionary of Military and Associated Terms defines an activity as a "function, mission, action, or collection of actions." [DoD JP-1-02] That is, the UML definition for use cases matches the use of the term operational activity as defined for this document.

Operational use cases are used to depict operational activities, where actors represent operational nodes (collaborating to conduct these use cases). In such a diagram, it is suggested 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. These operational use cases can then be decomposed 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 of architecture data (constraints). IDEF0 I/Os are reflected as event-action pairs appearing in sequence diagrams (OV-6c). **Figure 4-16** is an OV-5 use case diagram template.

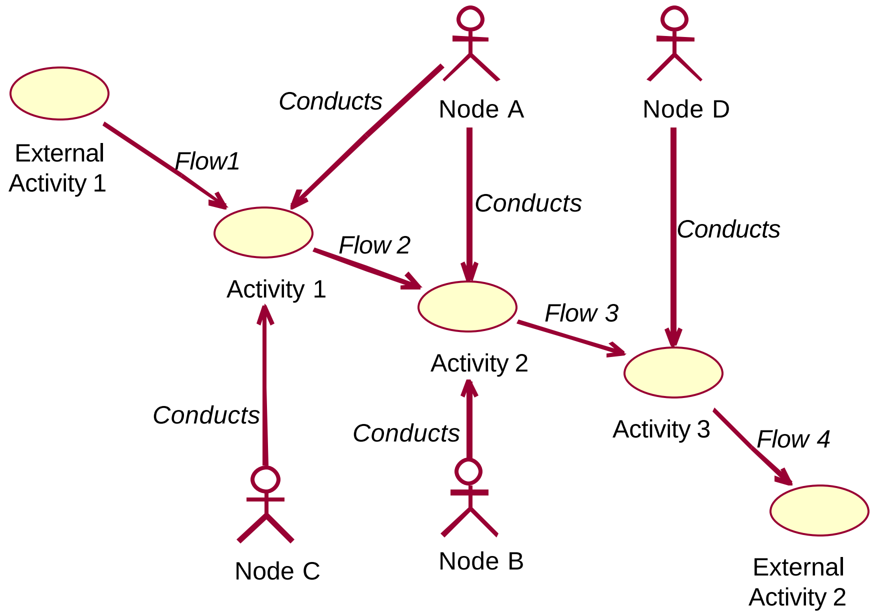


Figure 4-16. UML Use Case Diagram for Operational Activity Model (OV-5) – Template

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 that this diagram be used 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 (state diagrams) and can be useful for presenting another view of the dynamic behavior of an architecture or system. By using the swimlanes variant, 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 swimlane relate to the use cases and specify the sequence for the use cases that the operational node participates in (or collaborates on). The activity dia gram is used here to add the sequential flow aspect (among operational activities) to a use case diagram. The UML activity flow diagram template for OV-5 is shown in **Figure 4-17**.

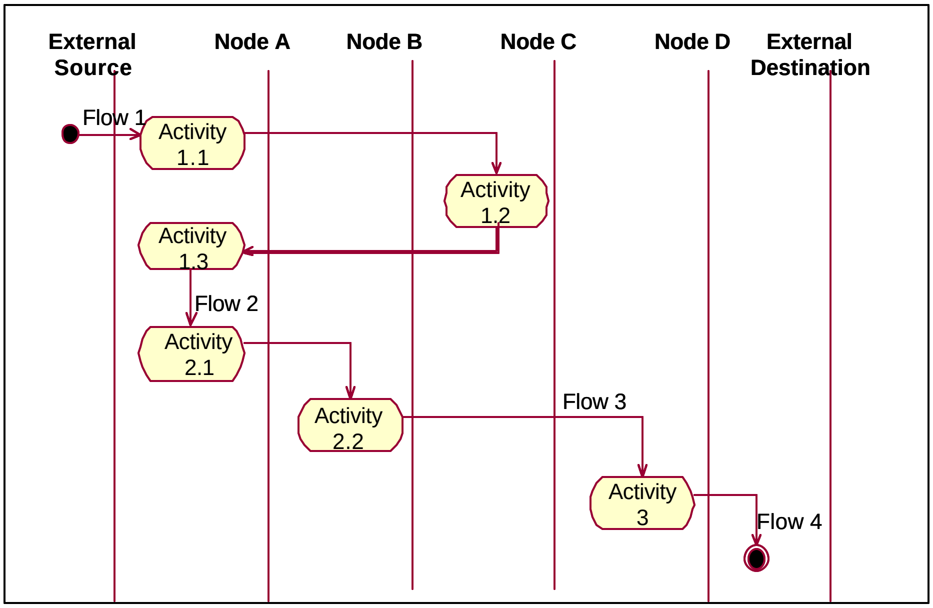


Figure 4-17. UML Activity Diagram for Operational Activity Model (OV-5) – Template

The activities in the activity diagram must correspond to and provide detail for the use cases (operational activities of OV-5) appearing in the related use case diagram(s) to ensure model consistency. Objects used to define activity diagram swimlanes must also correspond to the actors in related OV-2 collaboration diagrams (the same actors as in the related OV-5 use case diagrams).

4.5.3 Operational Activity Model (OV-5) – Data Element Definitions

Table 4-5 describes the architecture data elements for OV-5.

Table 4-5. Data Element Definitions for Operational Activity Model (OV-5)

Data Elements	Attributes	Example Values/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 policy or doctrine 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 (i.e., subordinate) activity
Flow Connector*		(For process-oriented models)
	Name*	Label of connector on graphic
	Type*	If using IDEF0: One of Input, Output, Control, Mechanism

Data Elements	Attributes	Example Values/Explanation
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)
	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 N/A
	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 in order 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
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)
Referenced Types		
Operational Node*		See OV-2 Definition Table
Information Exchange*		See OV-3 Definition Table
Human Role		See OV-4 Definition Table
System		See SV-1/2 Definition Table
System Function		See SV-4 Definition Table
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 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

Data Elements	Attributes	Example Values/Explanation
	Operational Node Name*	Name/identifier of the operational node that performs the operational activity
Operational Node, Systems Node, or System 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
	Model Name	Name of the model to which the diagram belongs
Facing Page Text References Diagram		
	Facing Page Text Identifier	Identifier/title for a page of text
	Diagram Title	Title of the diagram 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		
	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

4.5.4 CADM Support for Operational Activity Model (OV-5)

Each OV-5 is stored as an instance of DOCUMENT with a category code designating it as an ACTIVITY-MODEL-SPECIFICATION (a subtype of DOCUMENT). The structure and content specification of the activity model is stored as an instance of INFORMATION-ASSET with a category code designating it as an ACTIVITY-MODEL (a subtype of INFORMATION-ASSET).

An ACTIVITY-MODEL-SPECIFICATION can apply to many architectures. All the ARCHITECTURE instances to which that DOCUMENT applies are specified in ARCHITECTURE-DOCUMENT.

As depicted in **Figure 4-18**, each activity and subactivity cited in the ACTIVITY-MODEL is stored as an instance of PROCESS-ACTIVITY, and its occurrence in the activity model is stored as an instance of ACTIVITY-MODEL-PROCESS-ACTIVITY. The specific hierarchical breakdown of the activities in a specific ACTIVITY-MODEL is specified by instances of ACTIVITY-MODEL-PROCESS-ACTIVITY-ASSOCIATION.[16] Note that different activity models can depict *different* hierarchical breakdowns of the same activities; thus, PROCESS-ACTIVITY-ASSOCIATION is *not* used for this purpose. Where applicable, PROCESS-ACTIVITY-CAPABILITY may be used to document specific capabilities underlying an operational activity.

The information flows and their component information flows cited in the ACTIVITY-MODEL are stored as instances of INFORMATION-ELEMENT, and their occurrences in the activity model for a specific ACTIVITY-MODEL-PROCESS-ACTIVITY are stored as an instance of ACTIVITY-MODEL-INFORMATION-ELEMENT-ROLE. The decomposition of an INFORMATION-ELEMENT into its components is specified by using INFORMATION-ELEMENT-ASSOCIATION. (Since INFORMATION-ELEMENT-ASSOCIATION is not dependent on any activity model or process activity, such decomposition is *not* dependent on the activity model or on the activities specified in the activity model but is expected to be consistent across activity models.)

As each INFORMATION-ELEMENT occurs in an IDEF0 activity model for a specific activity, using MODEL-INFORMATION-ELEMENT-ROLE, there are four roles that can occur: input, control, output, and mechanism. Note that these roles are for the use of the INFORMATION-ELEMENT in relation to a specific ACTIVITY-MODEL-PROCESS-ACTIVITY and are not directly associated with the INFORMATION-ELEMENT itself.[17] The four roles are specified by use of an attribute (ACTIVITY-MODEL-INFORMATION-ELEMENT-ROLE Type Code) to designate exactly one of the following four subtypes:

- ACTIVITY-MODEL-INFORMATION-ELEMENT-ROLE-INPUT
- ACTIVITY-MODEL-INFORMATION-ELEMENT-ROLE-CONTROL
- ACTIVITY-MODEL-INFORMATION-ELEMENT-ROLE-OUTPUT
- ACTIVITY-MODEL-INFORMATION-ELEMENT-ROLE-MECHANISM

---

[16] Not (yet) included in the CADM are specifications for the Node Tree Diagram from IDEF0 that graphically depicts the activity hierarchy in a specific activity model. In the CADM this diagram is specified as an instance of DOCUMENT with a category code designating it as a NODE-TREE (a subtype of DOCUMENT), and its contents are specified with NODE-TREE-NODE-HIERARCHY and NODE-HIERARCHY, for which each NODE in the NODE-HIERARCHY represents a specific instance of PROCESS-ACTIVITY identified in NODE-PROCESS-ACTIVITY.

[17] It is for this reason that the DoD data standard for the entity listing information flows, formerly ICOM, has been changed to be INFORMATION-ELEMENT. The new name (ACTIVITY-MODEL-INFORMATION-ELEMENT-ROLE) for the entity ACTIVITY-ICOM has also been approved as a change to the DoD data standards.

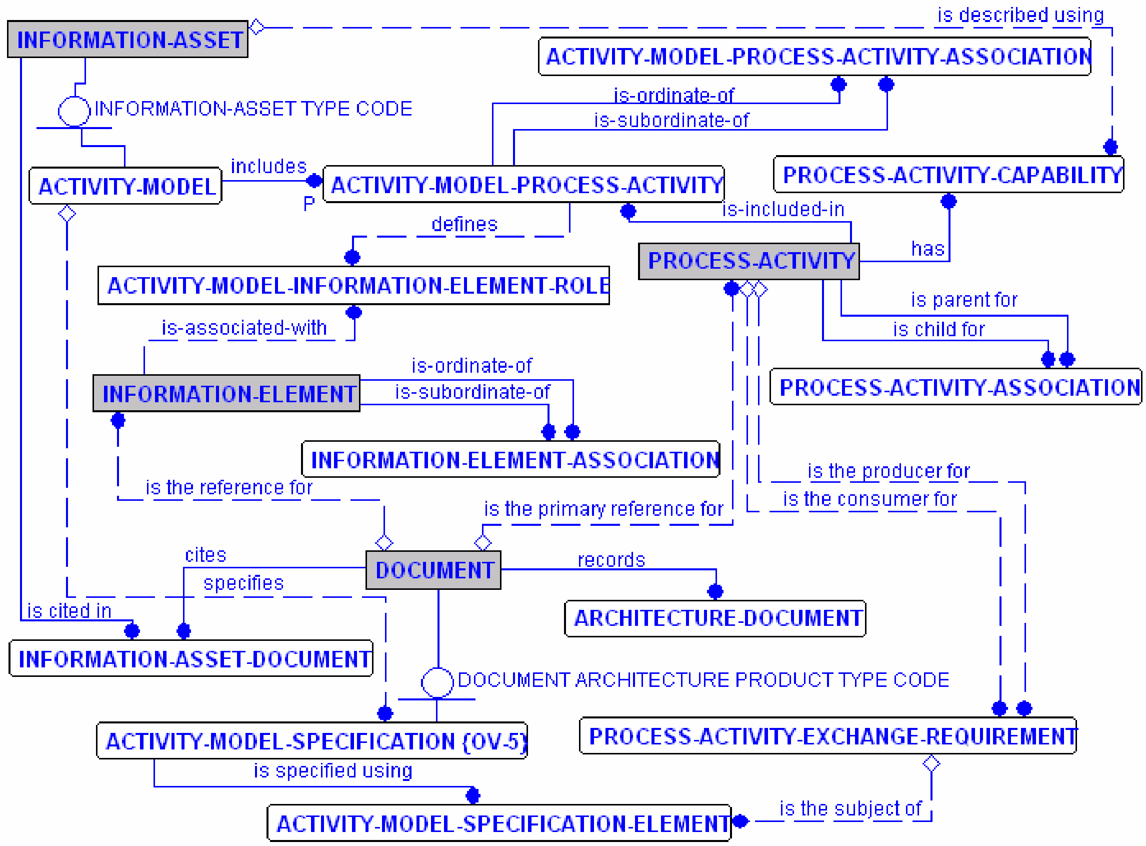


Figure 4-18. CADM ER Diagram for Operational Activity Model (OV-5)

Mechanisms can be very important for characterizing how elements of an activity model are or are to be supported in carrying out the underlying requirement. In general, mechanisms can include organizations, classes of organizations, systems, classes of materiel, persons with specific skills, and facilities. However, OV-5 focuses on requirements and therefore may intentionally omit references to materiel items, systems, and facilities. The specification of mechanisms in the CADM is by (optional) use of the following references (foreign key attributes):[18]

- ORGANIZATION Identifier in ACTIVITY-MODEL-INFORMATION-ELEMENT-ROLE-MECHANISM-REFERENCE (a subtype of ACTIVITY-MODEL-INFORMATION-ELEMENT-ROLE-MECHANISM)
- ORGANIZATION-TYPE Identifier in ACTIVITY-MODEL-INFORMATION-ELEMENT-ROLE-MECHANISM-REFERENCE
- OPERATIONAL-ROLE Identifier in ACTIVITY-MODEL-INFORMATION-ELEMENT-ROLE-MECHANISM-REFERENCE
- MATERIEL-ITEM Identifier in ACTIVITY-MODEL-INFORMATION-ELEMENT-ROLE-MECHANISM-REFERENCE

---

[18] Included in the All_CADM is the OCCUPATIONAL-SPECIALTY Identifier in ACTIVITY-MODEL-INFORMATION-ELEMENT-ROLE-MECHANISM-REFERENCE.

- SYSTEM Identifier in ACTIVITY-MODEL-INFORMATION-ELEMENT-ROLE-MECHANISM-SYSTEM (a subtype of ACTIVITY-MODEL-INFORMATION-ELEMENT-ROLE-MECHANISM)
- FACILITY Identifier in ACTIVITY-MODEL-INFORMATION-ELEMENT-ROLE-MECHANISM-FACILITY (a subtype of ACTIVITY-MODEL-INFORMATION-ELEMENT-ROLE-MECHANISM)
- FACILITY-TYPE Identifier in ACTIVITY-MODEL-INFORMATION-ELEMENT-ROLE-MECHANISM-FACILITY (a subtype of ACTIVITY-MODEL-INFORMATION-ELEMENT-ROLE-MECHANISM)

It is optional to incorporate some or all of the details of an activity model into a CADM-structured database. At a minimum, both INFORMATION-ELEMENT and PROCESS-ACTIVITY should be populated, so that the information flows and operational activities required for OV-3 and SV-5, respectively, are available for cross reference. Because system functions are specified by use of SYSTEM-FUNCTION, a subtype of PROCESS-ACTIVITY, the entity PROCESS-ACTIVITY-ASSOCIATION is used to record the cross reference of SYSTEM-FUNCTION instances to other instances of PROCESS-ACTIVITY.

A key element of operational requirement traceability is the relation of instances of PROCESS-ACTIVITY to TASK instances such as those included in the UJTL [CJCSM 3500.04C, 2002], the UJTL extensions for tactical tasks, and mission essential tasks. This traceability is provided by populating PROCESS-ACTIVITY-TASK.

Documents cited for details of an ACTIVITY-MODEL (or other subtypes of INFORMATION-ASSET) are stored as instances of INFORMATION-ASSET-DOCUMENT.

Expanding on the core entities shown in Figure 4-18, **Figure 4-19** provides a high-level diagram from the CADM showing key entities that are used to store architecture data for OV-5 in a CADM-conformant database.

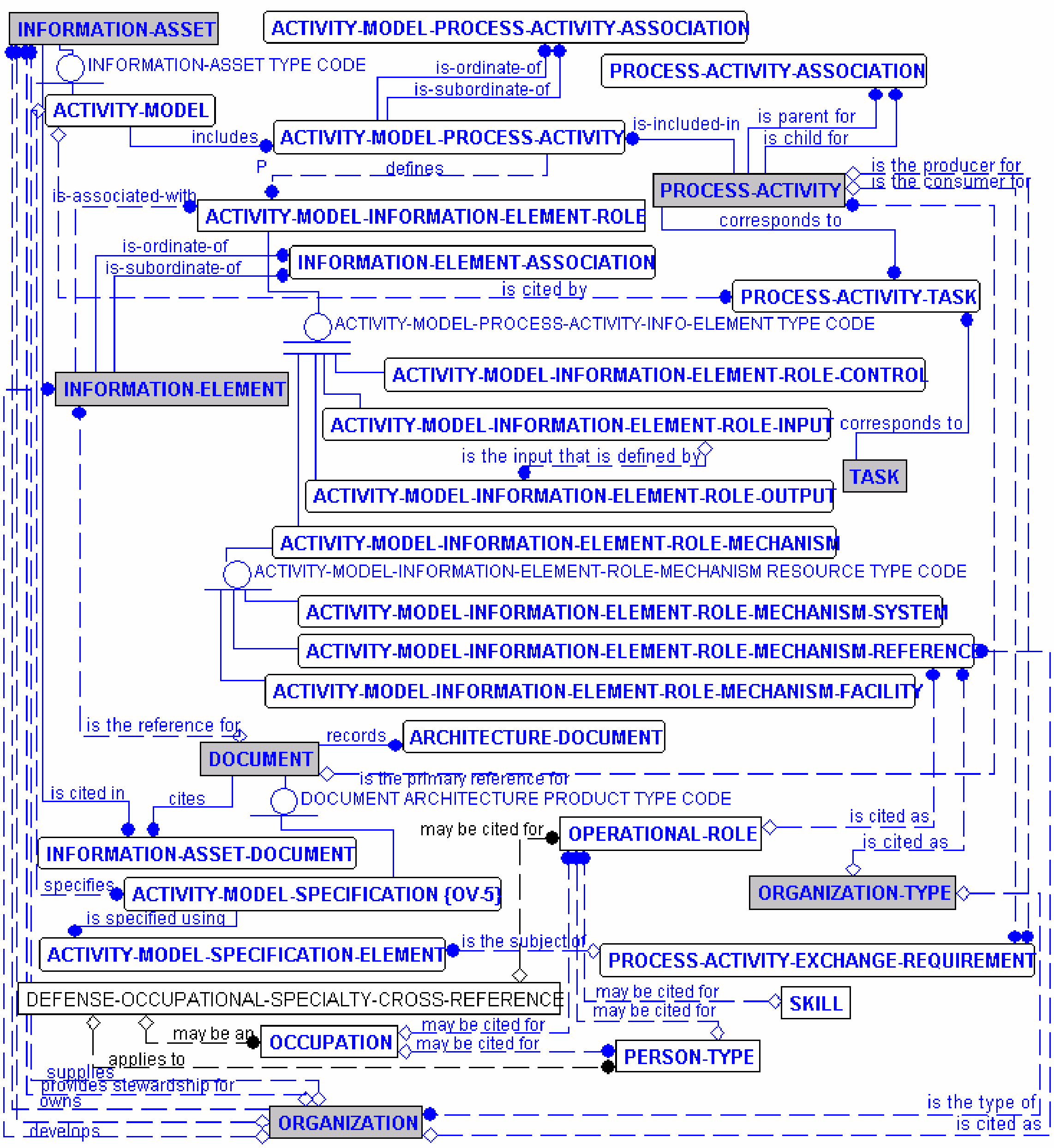


Figure 4-19. CADM ER Diagram for Operational Activity Model (OV-5)

4.6 OPERATIONAL ACTIVITY SEQUENCE AND TIMING DESCRIPTIONS (OV-6A, 6B, AND 6C)

4.6.1 Overview of Operational Activity Sequence and Timing Descriptions

OV products discussed in previous sections model the static structure of the architecture elements and their relationships. Many of the critical characteristics of an architecture are only discovered when the dynamic behavior of these elements is modeled to incorporate sequencing and timing aspects of the architecture.

The dynamic behavior referred to here concerns the timing and sequencing of events that capture operational behavior of a business process or mission thread for example. Thus, this behavior is related to the activities of OV-5. Behavior modeling and documentation is essential to a successful architecture description, because it is how the architecture behaves that is crucial in many situations. Knowledge of the operational nodes, activities, and information exchanges is crucial; but knowing when, for example, a response should be expected after sending message X to node Y can also be crucial to achieving successful operations.

Several modeling techniques may be used to refine and extend the architecture's OV to adequately describe the dynamic behavior and timing performance characteristics of an architecture, such as 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]. OV-6 includes three such models. They are:

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

OV-6 products portray some of the same architecture data elements, but each also portrays some unique architecture data elements. OV-6b and OV-6c may be used 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 (conditions) as described in OV-6a.

4.6.2 CADM Support for Operational Activity Sequences and Threads (OV-6)

Figure 4-20 provides a high-level diagram from the CADM showing key entities that are used to store architecture data for sequences and threads, which underlie OV-6a, OV-6b, and OV-6c. The CADM uses entities such as ACTION and ACTION-ASSOCIATION to support all the temporal and functional relationships among pairs of (operational and other) ACTION instances, as shown in Figure 4-20. The temporal attributes of ACTION-ASSOCIATION permit arbitrary sequencing of ACTION instances, as well as specifying timing offsets needed for planning concepts (e.g., a fire plan) such as relative timing from an H-Hour or a D-Day.

Threads of activities are specified using OPERATIONAL-MISSION-THREAD (for sequences of IERs), PROCESS-ACTIVITY-THREAD (for sequences of process activities), and ACTIVITY-MODEL-THREAD (for sequences of process activities embedded in a single activity model). These and related entities are also shown in Figure 4-20. Three entities—OPERATIONAL-MISSION-THREAD-CAPABILITY, PROCESS-ACTIVITY-CAPABILITY, and REQUIRED-INFORMATION-TECHNOLOGY-CAPABILITY (directly related in to INFORMATION-TECHNOLOGY-REQUIREMENT, the parent of INFORMATION-EXCHANGE-REQUIREMENT and PROCESS-ACTIVITY-EXCHANGE-REQUIREMENT)—provide the ability to express specific capabilities for threads of IERs and sequences of operational activities.

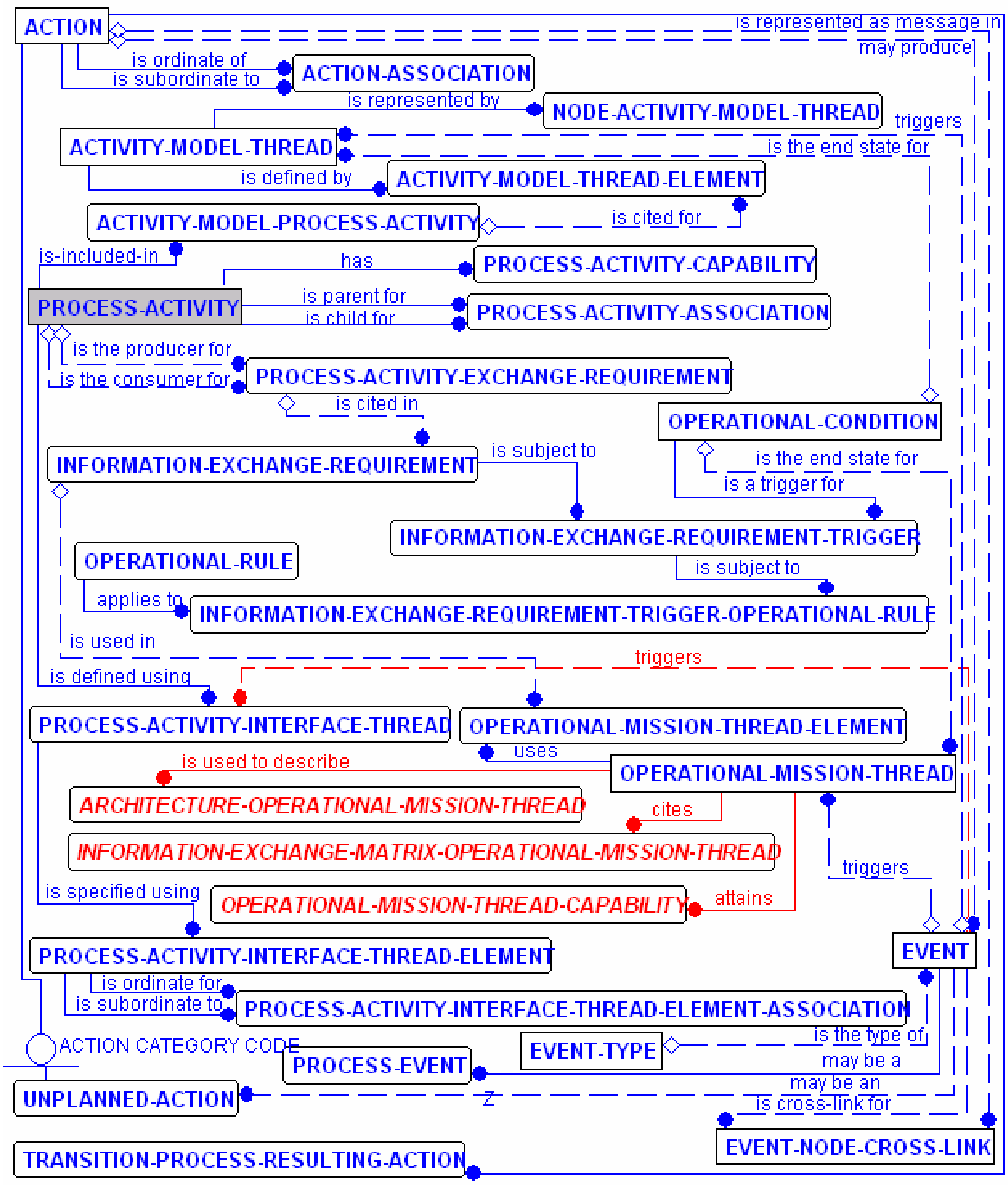


Figure 4-20. CADM ER Diagram for Operational Activity Sequences and Threads (OV-6)

4.6.3 Operational Rules Model (OV-6a)

Product Definition. The Operational Rules Model specifies operational or business rules are 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 doctrine, guidance, rules of engagement, and so forth. At the operation level, rules may include such things as a military Operational Plan (OPLAN). At lower levels, OV-6a describes the rules under which the architecture or its nodes behave under specified conditions. Such rules can be expressed in a textual form, for example, "If (these conditions) exist, and (this event) occurs, then (perform these actions)."

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

Product Detailed Description. *Rules are statements that define or constrain some aspect of the mission, or the architecture.* It is 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 (i.e., mission-oriented) not systems-oriented. These rules can include such guidance as the conditions under which operational control passes from one entity to another, or 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. OV-6a can also be used 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 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 and/or action assertions.

Operational rules can be grouped into the following categories:

- <u>Structural Assertions</u>: These rules concern mission or business domain terms and facts that are usually captured by the entities and relationships of Entity-Relationship models. They reflect static aspects of business rules that may also be captured in the Logical Data Model (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

Each architecture may select the formal language in which to record its OV-6a. The formal language selected should be referenced and well documented.

Figure 4-21 illustrates an example Action Assertion that might be part of OV-6a. The example is given in a form of structured English.

A base fact could be:

"Battle Damage Assessment consists of three activities: Conduct Battle Damage, Conduct Munitions Effects Assessment, and Recommend Restrike"

Derived facts could be:

"Recommend Restrike activity cannot be completed before a Battle Damage Assessment Report has been completed."

"Recommend Restrike activity cannot be completed before a Munitions Effects Assessment Report has been completed."

A derivation used to derive this derived fact above would be:

A Restrike Recommendation is based on facts contained in Battle Damage Assessment Reports, and Munitions Effects Assessment Reports

Figure 4-21. Operational Rules Model (OV-6a) – Action Assertion Example

4.6.4 UML Representation

There is no equivalent diagram in UML. 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 capture of business requirements and concept of operations for the use cases and activities of OV-5. If one considers Operational Rules to be equivalent to complex, nested If-Then-Else and CASE statements, then these statements can be unambiguously derived from UML statechart diagrams. Guard conditions can be specified for state transitions. Consequently, OV-6a may be generated 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.

4.6.5 Operational Rules Model (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 (i.e., Structural Assertion, Action Assertion, or Derivation), 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 4-6** describes the architecture data elements for OV-6a.

Table 4-6. Data Element Definitions for Operational Rules Model (OV-6a)

Data Elements	Attributes	Example Values/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 or derivation in selected formal language
	Formal Language*	Name of the formal language with which the rule is expressed
Referenced 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 rule model
Rule Applies to Operational Activity*		(If any rules are related to activities)
	Rule Name*	Name of action assertion or derivation rule
	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

4.6.6 CADM Support for Operational Rules Model (OV-6a)

Figure 4-22 provides a high-level diagram from the CADM showing key entities that are used to store architecture data for OV-6a in a CADM-conformant database. Each OV-6a is represented as an instance of RULE-MODEL, a subtype of DOCUMENT, which has attributes for Name, Time Frame, Category Code (with domain value Rule Model), and so on.

The CADM specification for OV-6a also allows the user to describe structural assertions, action assertions with their respective conditions, as well as the applicable integrity constraints, authorizations, and derivations via the instantiation of GUIDANCE Text for OPERATIONAL-RULE Category Code = *Structural Assertion*, *Action Assertion*, *Integrity Rule*, *Authorization*, or *Derivation*. Individual rules can be expressed explicitly, where appropriate, as relationships in a CONCEPTUAL-DATA-MODEL, SYSTEM-FUNCTIONALITY-DESCRIPTION, or ACTIVITY-MODEL. They can also be stated in formal or informal terms, as appropriate, as instances of OPERATIONAL-RULE, TECHNICAL-GUIDELINE, INFORMATION-TECHNOLOGY-REQUIREMENT, and other instances of GUIDANCE. Three subtypes of OPERATION-RULE are defined for the CADM to support specific OV-6a architecture data requirements from the Framework: ACTION-ASSERTION-RULE, STRUCTURAL-ASSERTION-RULE, and DATABASE-RULE. These rules are related to each other through GUIDANCE-ASSOCIATION and to various architecture products through GUIDANCE-DOCUMENT. The specific set of OPERATIONAL-RULE instances for a RULE-MODEL is specified as instances of the associative entity RULE-MODEL-OPERATIONAL-RULE.

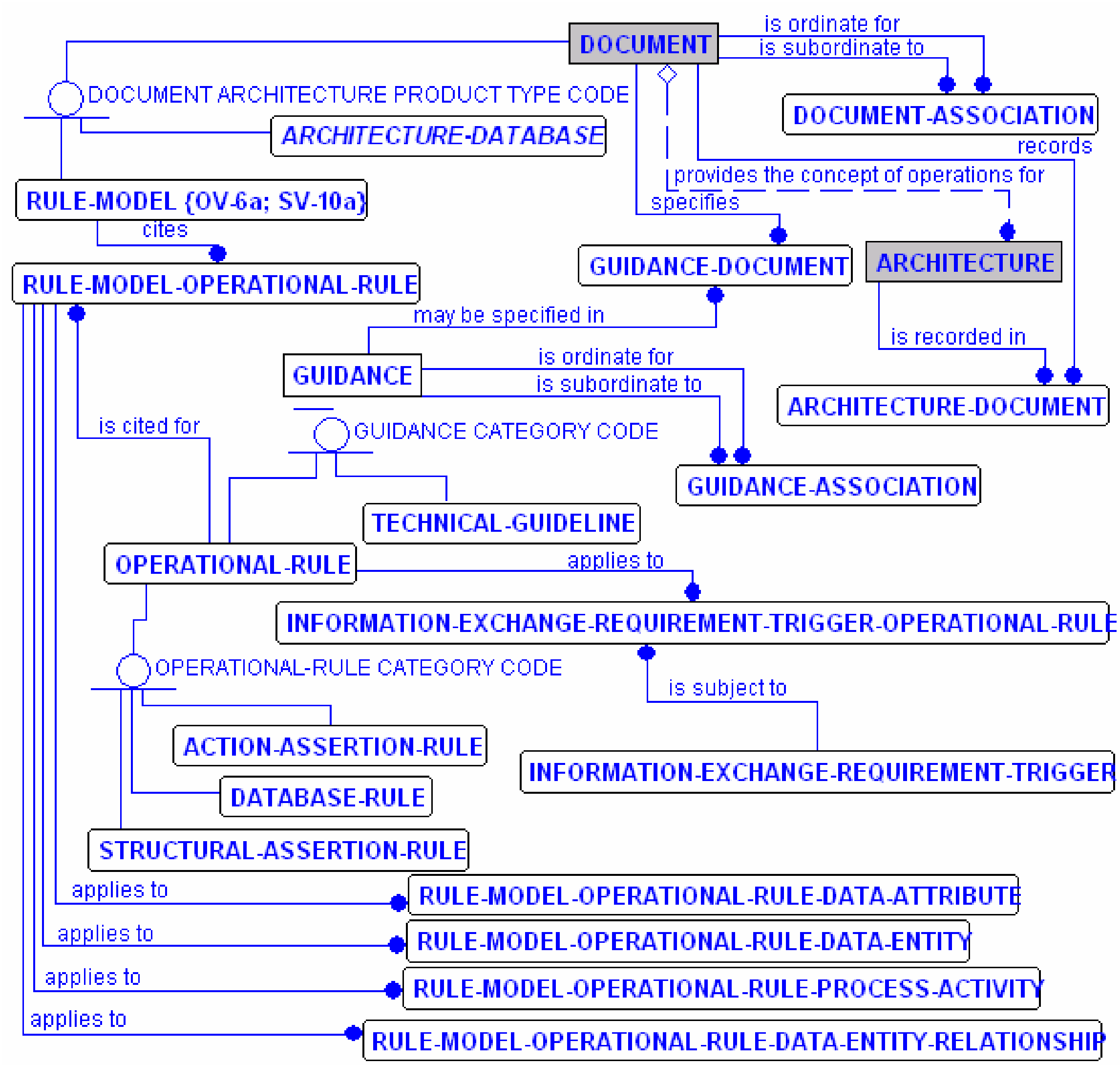


Figure 4-22. CADM ER Diagram for Operational Rules Model (OV-6a)

4.6.7 Operational State Transition Description (OV-6b)

Product Definition. The Operational State Transition Description 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.

Product Purpose. The explicit sequencing of activities in response to external and internal events is not fully expressed in OV-5. An Operational State Transition Description can be used to describe the explicit sequencing of the operational activities. Alternatively, OV-6b can be used 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.

Product Detailed Description. OV-6b is based on the statechart diagram. A state machine is defined as "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 action(s) 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.

Statechart diagrams can be unambiguously converted 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 4-23 provides a template for a simple OV-6b. The black dot and incoming arrow point to initial states (usually one per diagram), while terminal states are identified by an outgoing arrow pointing to a black dot with a circle around it. States are indicated by rounded corner box icons and labeled by name or number and, optionally, any actions associated with that state. Transitions between states are indicated by one-way arrows labeled with an event/action notation that indicates an event-action pair, and which 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 4-23. Operational State Transition Description – High-Level Template

Another dynamic behavior modeling technique is Colored PetriNet (CPN) [Kristensen, 1998]. A CPN can be used 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) and can be used to dynamically simulate an OV-5. With a CPN simulation engine or a discrete event workflow simulation engine, parallel event processing and decision support can be achieved.

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 and can make it possible to animate, analyze, and validate these complex relationships. This information can then be used to create new alternatives or it can be used to compare multiple alternatives until an optimal solution is found.

Most importantly, simulation takes information that traditionally was collected in static reports, and makes this information dynamic.

Dynamic analysis is able to 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, and discover and eliminate duplication of effort. Measures of Effectiveness (MOE) relative to mission objectives being evaluated and Measures of Operational Outcomes (MOO) relative to how operational requirements contribute to end results can be determined.

In addition to examining behavior over time, one 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 we eventually need to align architectures to funding decisions to ensure that investment decisions are directly linked to mission objectives and their outcomes. **Figure 4-24** illustrates the anatomy of one such dynamic model.

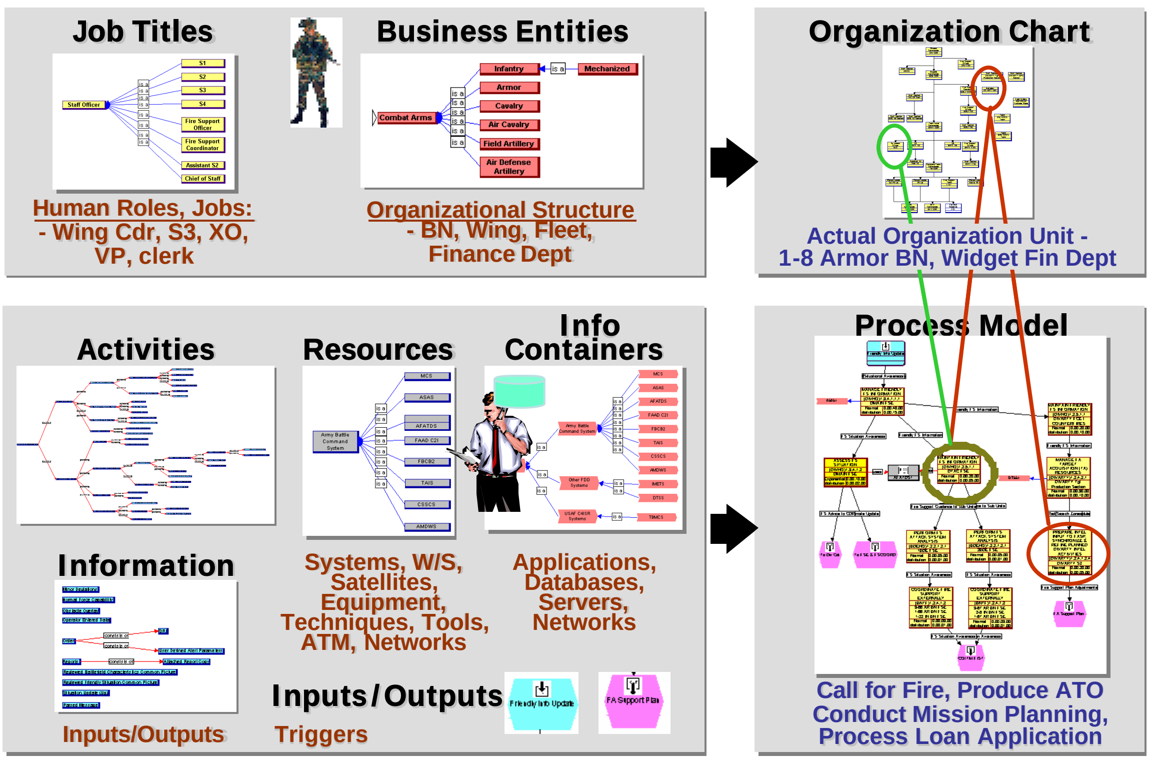


Figure 4-24. Anatomy of an Executable Operational Architecture

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 of processes, human and system resources, and their used capacity over time during a simulation run. These histograms are used to perform dynamic impact analysis of the behavior of the executable architecture. **Figure 4-25** is an example showing the results of a simulation run of human resource capacity.

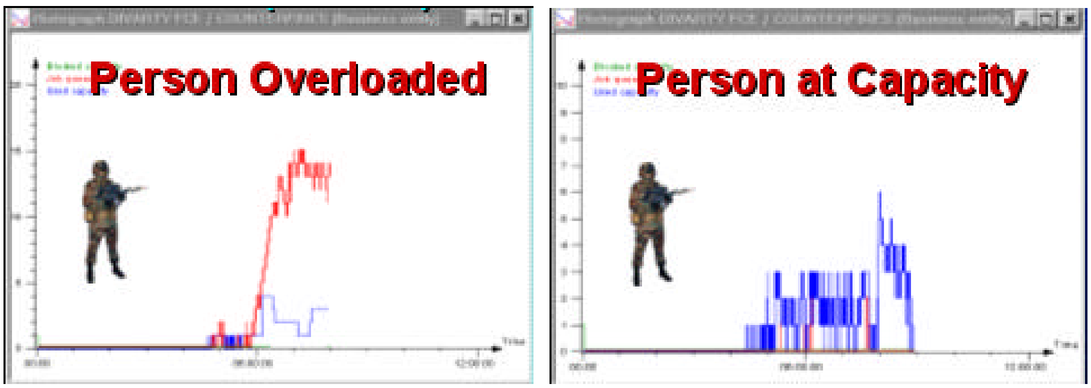


Figure 4-25. Sample Histograms Showing Results of a Simulation Run

4.6.8 UML Representation

OV-6b may be produced 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 4-23) 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, where a state is defined in UML 1.4 as "A condition or situation during the life of an object during which it 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 swimlanes) should be mappable to a set of statechart diagrams for those operational nodes, and 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 and are the same events shown on the sequence diagrams of OV-6c.

4.6.9 Operational State Transition Description (OV-6b) – Data Element Definitions

OV-6b describes the detailed time sequencing of activities or work flow in the business process, depicting how the current state of a process or activity changes in response to external and internal events. **Table 4-7** describes the architecture data elements for OV-6b.

Table 4-7. Data Element Definitions for Operational State Transition Description (OV-6b)

Data Elements	Attributes	Example Values/Explanation
Graphical Box Types		
State*		
	Name*	State name
	Description*	Textual description as necessary
	Type*	e.g., simple, composite, synchstate, or sub state
	Number of Partitions	Number of contained state charts
Graphical Arrow Types		
Transition*		
	Label*	Identifier for transition
	Description*	Textual description of transition
	Type*	e.g., 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*		
	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 being 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 provides 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 the 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
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 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

Data Elements	Attributes	Example Values/Explanation
Transition is Associated With Source		
	Source State Name*	Name of originating s tate vertex (state or pseudostate) 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 Control of Operational Activity*		When an event represents information (I/O) or a control, whose receipt triggers a transition, 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)

4.6.10 CADM Support for Operational State Transition Description (OV-6b)

Figure 4-26 provides a high-level diagram from the CADM showing key entities that are used to store architecture data for OV-6b in a CADM-conformant database. Each OV-6b is represented as an instance of STATE-TRANSITION-DESCRIPTION, a subtype of DOCUMENT, which has attributes for Name, Time Frame, Category Code (with domain value State Transition Description), and so forth.

The CADM specification for OV-6b also allows the user to describe all the relevant components of a state transition diagram by the proper instantiation of PROCESS-STATE, PROCESS-STATE-ACTION, TRANSITION-PROCESS, TRANSITION-PROCESS-RESULTING-ACTION, PROCESS-PSEUDO-STATE, STATE-TRANSITION-DESCRIPTION, PROCESS-STATE-VERTEX, COMPOSITE-PROCESS STATE, and NESTING-PROCESS-STATE.

As noted, OV-6b is an instance of DOCUMENT in the subtype STATE-TRANSITION-DESCRIPTION. To avoid confusion with U.S. State, the term "PROCESS-STATE" has been used (parallel to PROCESS-ACTIVITY) instead of STATE. The key entities in the CADM are independent entities TRANSITION-PROCESS, PROCESS-STATE-VERTEX, ACTION, and EVENT. Instances of TRANSITION-PROCESS are defined in terms of pairs of instances of PROCESS-STATE-VERTEX, each of which may be a PROCESS-PSEUDO-STATE (representing forks, joins, branches, etc.) or PROCESS-STATE. Key associations include TRANSITION-PROCESS-RESULTING-ACTION and PROCESS-STATE-ACTION. The CADM provides explicit specification of four types of events called out in UML: SIGNAL-EVENT, CALL-EVENT, TIME-EVENT, and CHANGE-EVENT. The CADM also explicitly permits a STATE-TRANSITION-DESCRIPTION (OV-6b) to represent a specific SYSTEM (including a specific component), SOFTWARE-ITEM, or ACTION (which may represent a TASK or a PROCESS-ACTIVITY).

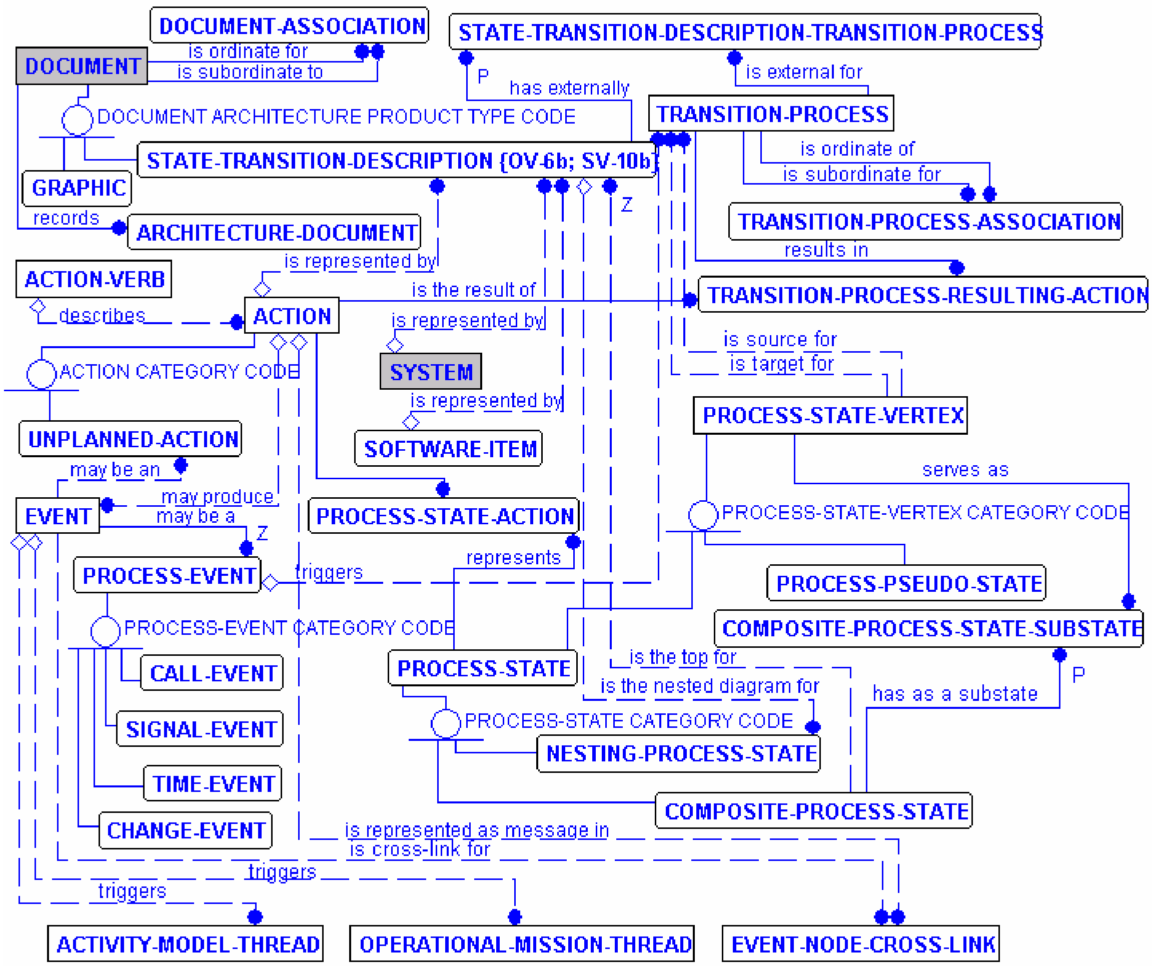


Figure 4-26. CADM ER Diagram for Operational State Transition Description (OV-6b)

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

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.

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 in order to perform its assigned operational activity.

Product Detailed Description. OV-6c allows the tracing of actions in a scenario or critical sequence of events. OV-6c can be used by itself or in conjunction with OV-6b to describe the dynamic behavior of business processes or a mission/operational thread. An operational thread is defined as 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 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. The sequence of activities forms the basis for defining and understanding the many factors that impact on the capability.

Integrated architectures with DOTMLPF information provide a structured and organized approach for defining capabilities and understanding the underlying requirements for achieving those capabilities. By describing the sequence and timing of activities, tying them to the operational nodes (representing organizations or human roles), relating them to their supporting systems or system functions, and specifying the actions, events, and related guard conditions or business rules that constrain those activities, the full spectrum of DOTMLPF is modeled and related, so that analyses and decisions can be supported. Below is a detailed description of how DOTMLPF is tightly woven into this and related products.

- Doctrine is represented as guard conditions, which are associated with events, which, in turn, map to controls in OV-5.
- Organization is represented via the lifelines or swimlanes, which map to operational nodes of OV-2; which, in turn, map to organizations, organization types, or human roles of OV-4; and mechanisms in OV-5.
- Training is represented via the lifeline or swimlane, since the operational node (shown on the diagram as a lifeline or swimlane) may represent a human role, which, in turn, embodies a certain skill set or knowledge domain required to perform the actions (which are, in turn, related to operational activities of OV-5).
- Materiel is tied to the elements in OV-6c because this product is tightly coupled with OV-5, where mechanisms may be used to represent systems that support operational activities. In addition, further materiel detail may be related to the activities via SV-5, by relating those activities to the system functions that are executed by systems that automate them (wholly or partially). Each operational thread or scenario (represented by an OV-6c) is associated with a certain capability, since a capability is defined in terms of the activities and their attributes depicted in OV-6c. Consequently, an SV-5 may also be used to relate a capability to the systems that support it, by labeling a set of activities with their associated capability (defined in OV-6c), and by labeling the system functions with the systems that execute them (defined in SV-1, SV-2, and SV-4).
- Leadership may be represented either directly via the lifeline or swimlane or indirectly through the relationships of an operational node (shown as a lifeline or swimlane on the diagram) in OV-2 to organizations, organization types, or leadership human roles in OV-4.
- Personnel may be represented directly via the lifeline or swimlane, or indirectly through the relationships of an operational node (shown as a lifeline or swimlane on the diagram) in OV-2 to organizations, organization types, or human roles in OV-4.
- Facility is tied to the elements in OV-6c because the lifeline or swimlane representing an operational node is directly tied to the systems node (facilities)

that house the systems, which may be shown as mechanisms that support operational activities.

The Framework does not endorse a specific event-trace modeling methodology. Two such types of models include UML sequence diagrams [OMG, 2003] and IDEF3 [IDEF3, 1995]. The OV-6c product may be developed 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. Different scenarios should be depicted by separate diagrams. **Figure 4-27** 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 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. 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. Labels indicating timing constraints or providing descriptions can be shown 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.

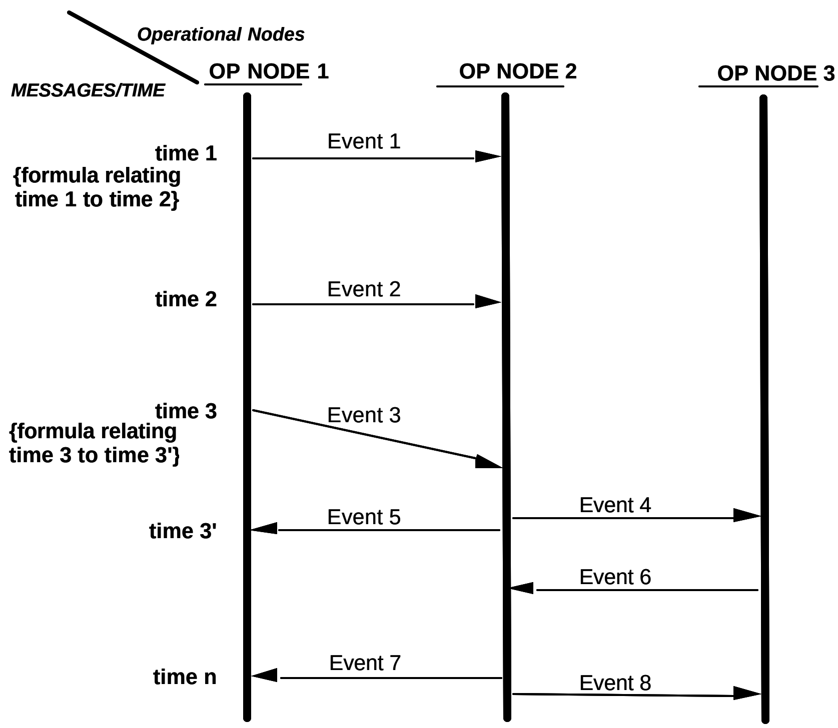


Figure 4-27. Operational Event-Trace Description (OV-6c) – UML-type Template

Figure 4-28 shows an example of an Operational Event-Trace Description using IDEF3 notation. Boxes convey a Unit of Behavior (UOB), which is a step in the process and ties to an Operational Activity in OV-5 via an IDEF0 Reference Property. A swimlane represents a horizontal or vertical division of the diagram for showing what operational nodes perform what processes (UOBs), and links show control flow between UOBs (not information flow). IDEF3 link types include temporal (sequence, precedence) and logical (guard conditions or business rules).

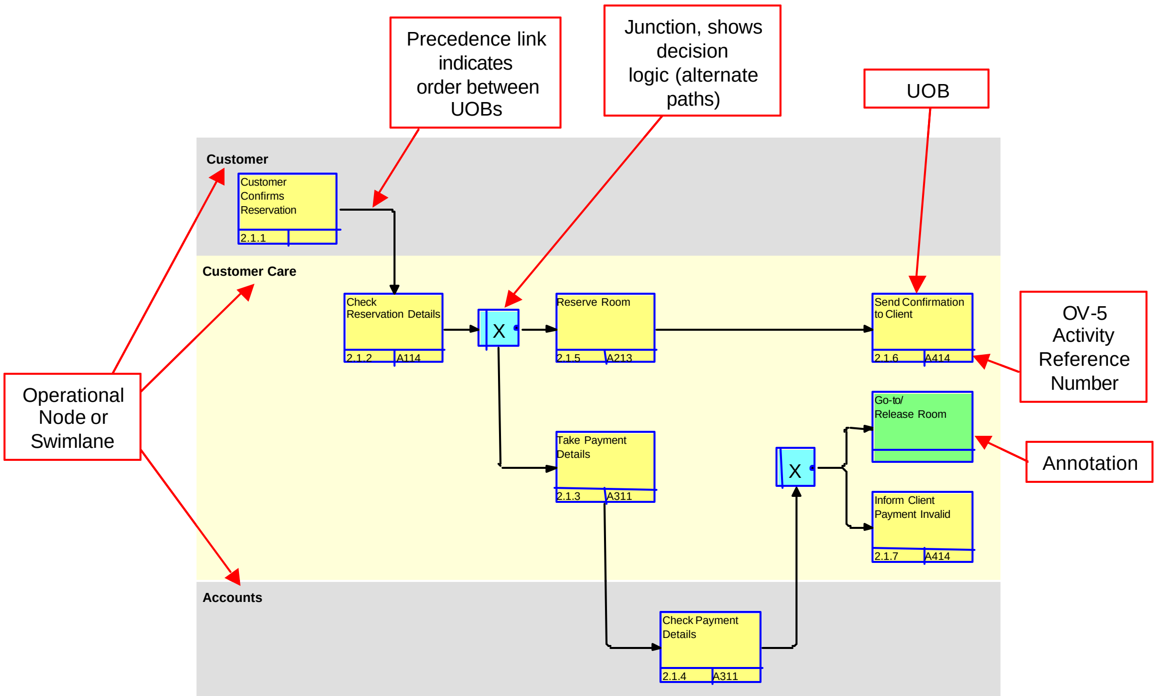


Figure 4-28. Operational Event-Trace Description (OV-6c) – IDEF3 Example

4.6.12 UML Representation

UML sequence diagrams may be used to model OV-6c (see section 4.6.11 for a description and Figure 4-28 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 diagram. Objects (actor instances) referenced in the sequence diagrams should also be traceable to class diagrams of OV-4.

4.6.13 Operational Event-Trace Description (OV-6c) – Data Element Definitions

Table 4-8 describes the architecture data elements for OV-6c.

Table 4-8. Data Element Definitions for Operational Event-Trace Description (OV-6c)

Data Elements	Attributes	Example Values/Explanation
Graphical Box Types		
Lifeline or Swimlane*		
	Name*	Name of the lifeline or swimlane
	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

Data Elements	Attributes	Example Values/Explanation
Link		If using IDEF 3
	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
	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 that the operational thread supports
Event Time*		
	Identifier*	Identifier for the time when 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
Node Lifeline (or Swimlane) 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 the ending time
Event Ends at Time*		
	Event Name*	Label of the event
	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

Data Elements	Attributes	Example Values/Explanation
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*		A event represents (and can be denoted by) action(s) performed by the originating node, event(s) produced by the action(s), or both
	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
	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	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

4.6.14 CADM Support for Operational Event-Trace Description (OV-6c)

Figure 4-29 provides a high-level diagram from the CADM showing key entities that are used to store architecture data for OV-6c in a CADM-conformant database. Each OV-6c is represented as an instance of EVENT-TRACE-DESCRIPTION and a subtype of DOCUMENT, which has attributes for Name, Time Frame, Category Code (with domain value Event-Trace Description), and so forth.

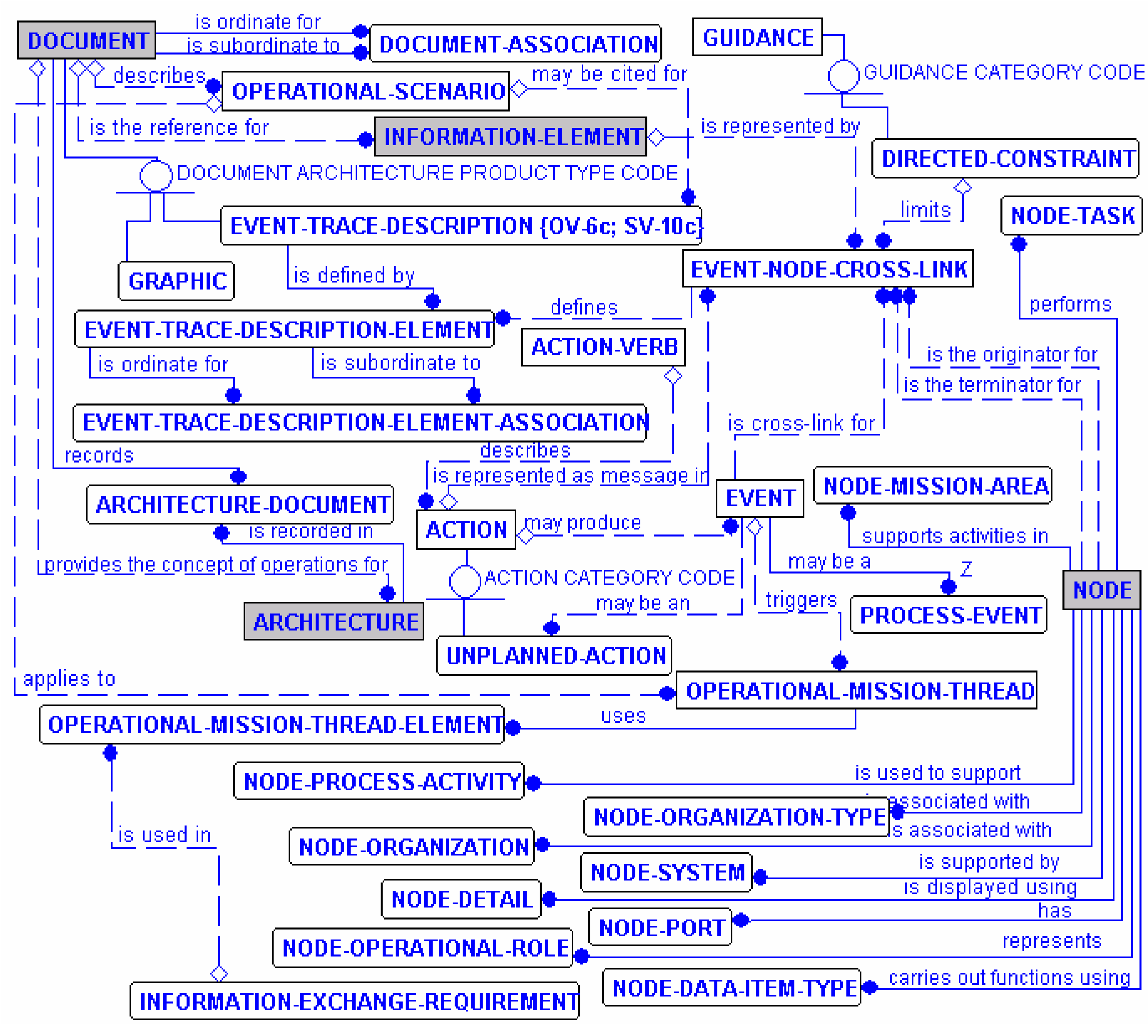


Figure 4-29. CADM ER Diagram for Operational Event-Trace Description (OV-6c)

Without defining a specific entity, each NODE is considered to have a (relative) timeline (or lifeline) in the CADM. Each such NODE can be explicitly associated with a MISSION-AREA, ORGANIZATION, ORGANIZATION-TYPE, PROCESS-ACTIVITY,[19] SYSTEM, and TASK. Each cross-link connecting two NODE instances at different times is represented (independently) as an instance of EVENT-NODE-CROSS-LINK (independence enables cross-links to be reused from one scenario to another). Cross-links can start at explicit or implicit (computed) absolute or relative times, the latter in terms of the start or end of another cross-link (using a method analogous to one defined for temporally related ACTION instances). The set of cross-links for a specific OV-6c is specified as a set of instances of EVENT-TRACE-SCENARIO-ELEMENT. Each EVENT-TRACE-DESCRIPTION can be associated directly to an OPERATIONAL-SCENARIO. As noted in the general description of OV-6 products (above), the relationship of OV-6c to capabilities is provided by OPERATIONAL-MISSION-THREAD-CAPABILITY, PROCESS-ACTIVITY-CAPABILITY, and REQUIRED-INFORMATION-TECHNOLOGY-CAPABILITY.

---

[19] The entity NODE-PROCESS-ACTIVITY is used in the OV-6c to represent the lifeline associated with one or more operational activities. Because SYSTEM-FUNCTION is a subtype (subset) of PROCESS-ACTIVITY, this same entity is used in SV-10c to represent the lifeline associated with one or more system functions.

4.7 LOGICAL DATA MODEL (OV-7)

4.7.1 Logical Data Model (OV-7) – Product Description

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

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 structure. This situation will pose significant interoperability risks, as the system data models may appear to be compatible, each having a *Target Track* data entity but having different and incompatible interpretations of what *Target 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.

Product Detailed Description. OV-7 defines the architecture domain's system data types (or entities) and the relationships among the system data types. For example, if the domain is missile defense, some possible system data types may be *trajectory* and *target* with a relationship that associates a target with a certain trajectory. On the other hand, architecture data types for the DoDAF (i.e., DoDAF-defined architecture data elements, AV-2 data types, and CADM entities) are things like an *operational node* or *operational activity.* 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 CADM. OV-7 is an architecture product and describes information about a specific architecture domain. The CADM is not an architecture product. The CADM is a database design for a repository of DoDAF products and architecture data. CADM-based repositories can store architecture products, including Logical Data Models, from any DoDAF-based architecture project. Thus, the CADM addresses a structure for storing architecture data (e.g., instances of operational nodes and operational activities), while a Logical Data Model for missile defense, for example, might define architecture domain entities and relationships such as missile tracks and points of impact.

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, such as 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. Attributes can be associated with entities and with relationships, depending on the purposes of the architecture.

Figure 4-30 provides a template for OV-7 (with attributes). The format is intentionally generic to avoid implying a specific methodology.

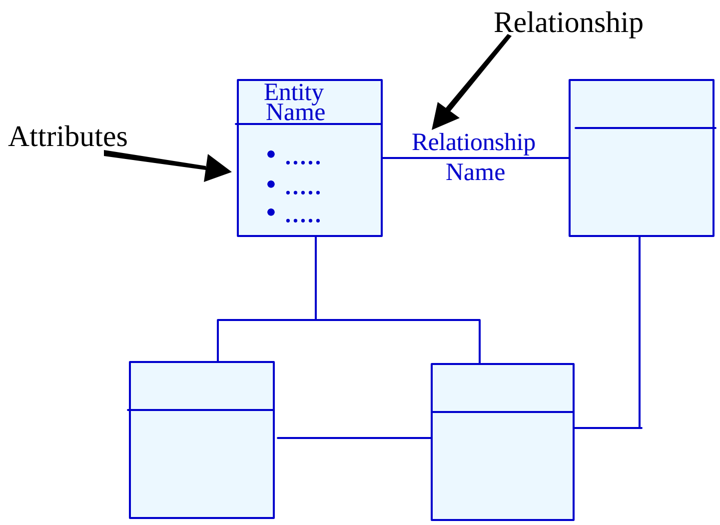


Figure 4-30. Logical Data Model (OV-7) – Template

4.7.2 UML Representation

OV-7 may be modeled in UML using a class diagram. See **Figure 4-31**. Class diagrams offer all the UML elements needed to produce entity-relationship diagrams. Class diagrams consist of classes, interfaces, collaborations, 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).

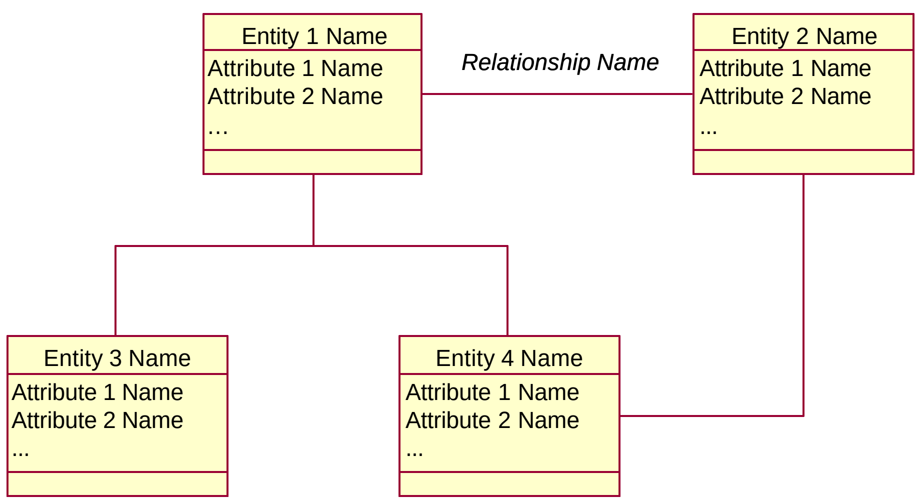


Figure 4-31. UML Class Diagram for Logical Data Model (OV-7) – Template

4.7.3 Logical Data Model (OV-7) – Data Element Definitions

Table 4-9 describes the architecture data elements for OV-7.

Table 4-9. Data Element Definitions for Logical Data Model (OV-7)

Data Elements	Attributes	Example Values/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 (e.g., DDDS reference)
Graphical Arrow Types		
Relationship Type*		
	Name*	Name/identifier of the relationship type
	Description*	Textual description of the relationship represented
	Source Entity Type Name*	Name of the entity type at the source of the relationship
	Target Entity Type Name*	Name of the entity type at the target of the relationship
	Cardinality Designation*	Examples: one to one, one to many, etc.
Category Relationship Type		
	Name	Name of the subtyping relationship
	Description	Textual description of the subtype relationship represented
	Source Discriminated Entity Type Name	Name of the supertype that is the source of the relationship
	Discriminant Attribute Type Name	Name of the attribute type that provides the discriminant for the entity type (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 (e.g., DDDS reference)
Data Model Rule		
	Name	Name/identifier of rule
	Type	Examples: Null rule; child delete rule, child update rule
	Text	Text of rule
Data Domain		
	Name	Name of data domain
	Description	Textual description of data domain
	Range Constraint	Value range allowable for attributes in data domain
	Size Constraint	Maximum number of characters in display representation
Referenced Types		
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 Elements	Attributes	Example Values/Explanation
Data Domain Constrains Values of Attribute Type		
	Data Domain Name	Name of data domain
	Attribute Type Name	Name of attribute type whose values are selected from the data domain
Relationship Type has Rule		
	Relationship Type Name	Name of a relationship type
	Rule 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 entity type
Attribute Type has Rule		
	Attribute Type Name	Name of an attribute type
	Rule Name	Name/identifier of a rule associated with an attribute
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 I/O 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 I/O (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

4.7.4 CADM Support for Logical Data Model (OV-7)

Figure 4-32 provides a high-level diagram from the CADM showing key entities that are used to store architecture data for OV-7 in a CADM-conformant database. Each OV-7 is represented as an instance of LOGICAL-DATA-MODEL, a subtype of DOCUMENT. LOGICAL-DATA-MODEL cites a specific instance of CONCEPTUAL-DATA-MODEL (a subtype of INFORMATION-ASSET). The OV-7 use of the CONCEPTUAL-DATA-MODEL is limited to the relevant set of DoD data standards from the Information Management View of the DoD Data Architecture (DDA) data model. The key entities in this specification include DATA-ENTITY, DATA-ATTRIBUTE, and

DATA-DOMAIN (all subtypes of INFORMATION-ASSET), together with other entities such as DATA-ENTITY-RELATIONSHIP, CATEGORY-RELATIONSHIP, DATA-DOMAIN-LIST, and DATA-DOMAIN-LIST-VALUE. Each DATA-ENTITY can be directly related to an ACTIVITY-MODEL-INFORMATION-ELEMENT-ROLE (providing a link exists between data models and activity models).

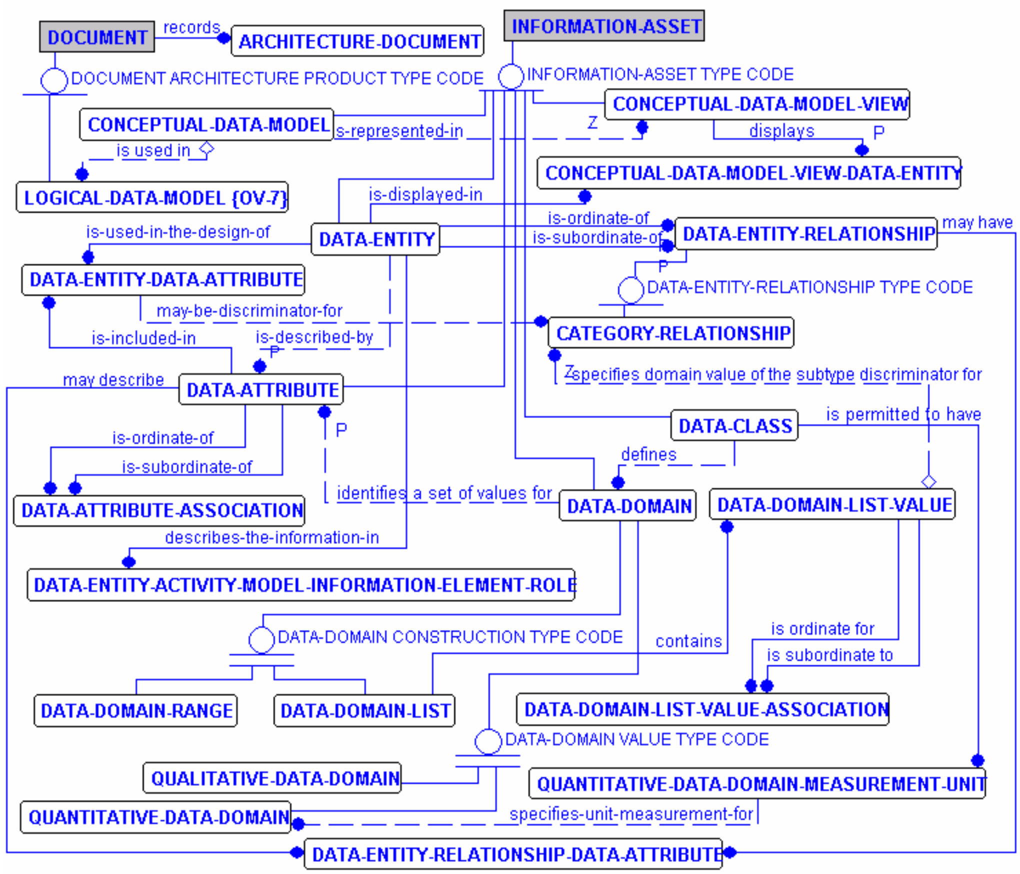


Figure 4-32. CADM ER Diagram for Logical Data Model (OV-7)

5 SYSTEMS VIEW PRODUCTS

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

There are eleven SV products:

- 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 and Timing Descriptions (SV-10a, 10b, and 10c)
- Physical Schema (SV-11)

5.1 SYSTEMS INTERFACE DESCRIPTION (SV-1)

5.1.1 Systems Interface Description (SV-1) – Product Description

Product Definition. The Systems Interface Description depicts systems nodes and the systems resident at these nodes to support organizations/human roles represented by operational nodes of the Operational Node Connectivity Description (OV-2). SV-1 also identifies the interfaces between systems and systems nodes.

Product Purpose. SV-1 identifies systems nodes and systems that support operational nodes. Interfaces that cross organizational boundaries (key interfaces) can also be identified in this product. Some systems can have numerous interfaces. Initial versions of this product may only show key interfaces. Detailed versions may also be developed, as needed, for use in system acquisition, as part of requirements specifications, and for determining system interoperabilities at a finer level of technical detail.

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, and/or human roles, while SV-1 depicts the systems nodes that house operational nodes (e.g., 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 framework is used to denote a family of systems (FoS), system of systems (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 (e.g., physical links, communications networks, routers, switches, communications systems, satellites) are documented in the Systems Communication Description (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) and 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 (e.g., multiple physical links). Details of the physical links and communications networks that implement the interfaces are documented in SV-2. Characteristics of the interface are described in Systems-Systems Matrix (SV-3). System functions and system data flows are documented in a Systems Functionality Description (SV-4), and the system data carried by an interface are documented in the Systems Data Exchange Matrix (SV-6).

An interface between systems nodes or systems may be annotated as a Key Interface (KI). A KI is defined as an interface where one or more of the following criteria are met:

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

If desired, annotations summarizing the system data exchanges carried by an interface may be added to SV-1.

Several versions of SV-1 can be developed 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. Other versions may also be developed, depending on the purposes of the particular architecture.

Figure 5-1 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 5-1, System 1 is depicted in all three nodes (i.e., Node A, Node B, and Node C), which means that the same hardware/software configuration is resident at all three systems nodes. (Here, 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 5-1 is purely for exposition purposes.

Additional information may also be shown on the graphics. For example, information, such as a system's hardware/software items or the system's functions, can be added as annotations to the system's graphical icon.

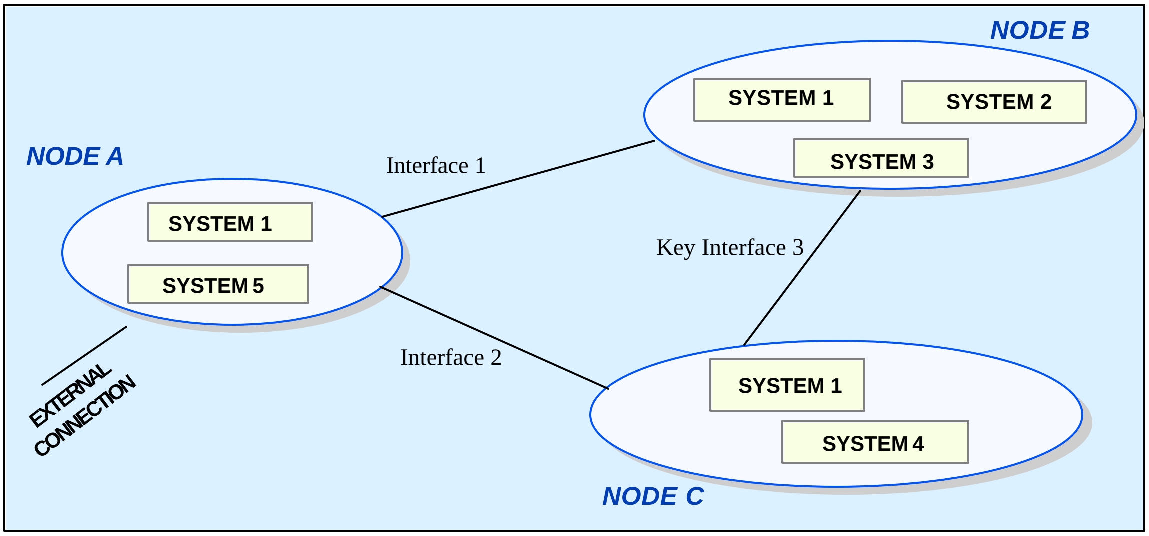


Figure 5-1. SV-1 Internodal Template Showing Systems

Figure 5-2 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 target architecture, where major system functions have been allocated to systems but other details have not yet been decided.

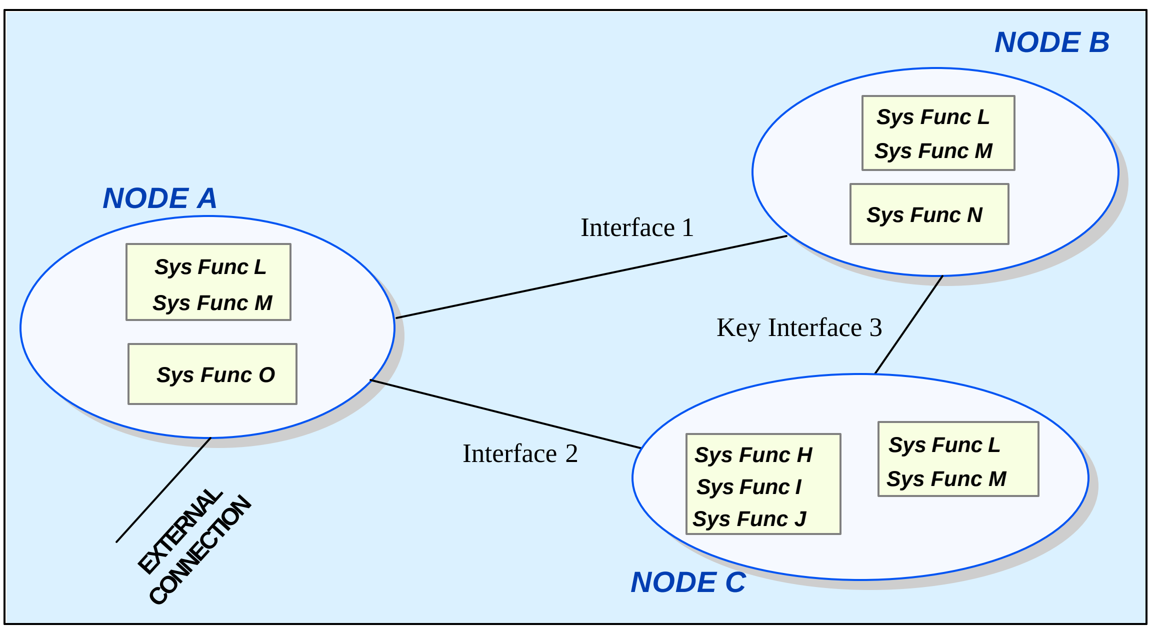


Figure 5-2. SV-1 Internodal Version – Node Edge to Node Edge Showing System Functions

Figure 5-3 provides a template of the internodal version of SV-1 that extends the node edge connections to specific systems. In Figure 5-3, the line between System 1 at Node A and System 1 at Node B indicates that 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.

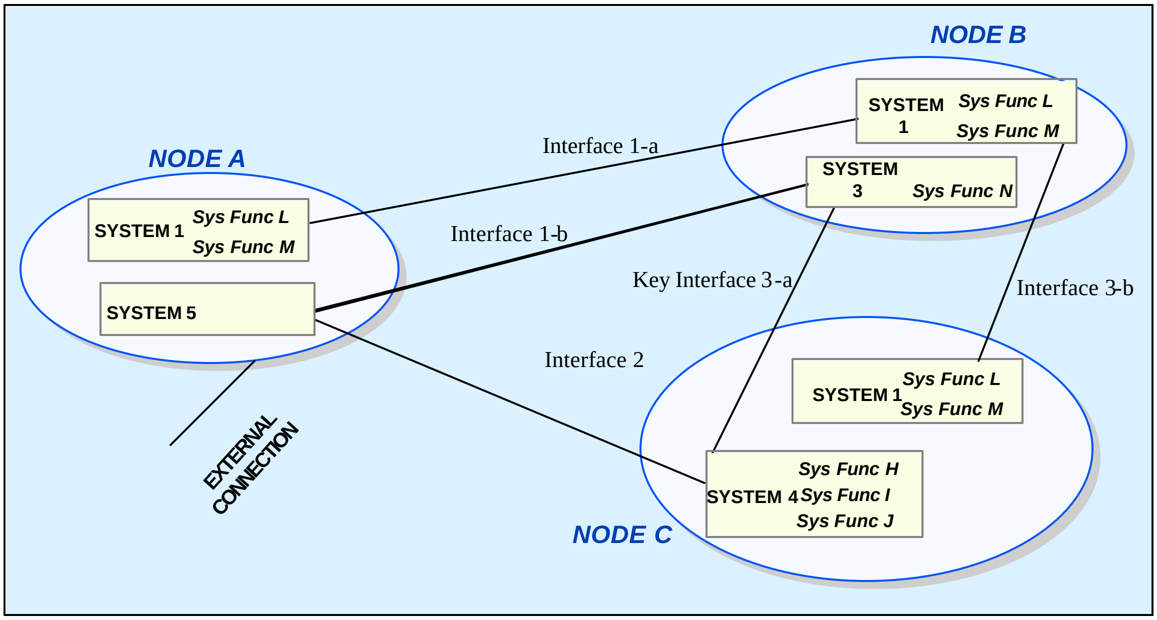


Figure 5-3. SV-1 Internodal Version Showing System-System Interfaces – Template

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 5-4** provides a template of the intranodal version of SV-1. Note that the naming convention for the interfaces in Figure 5-4 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.

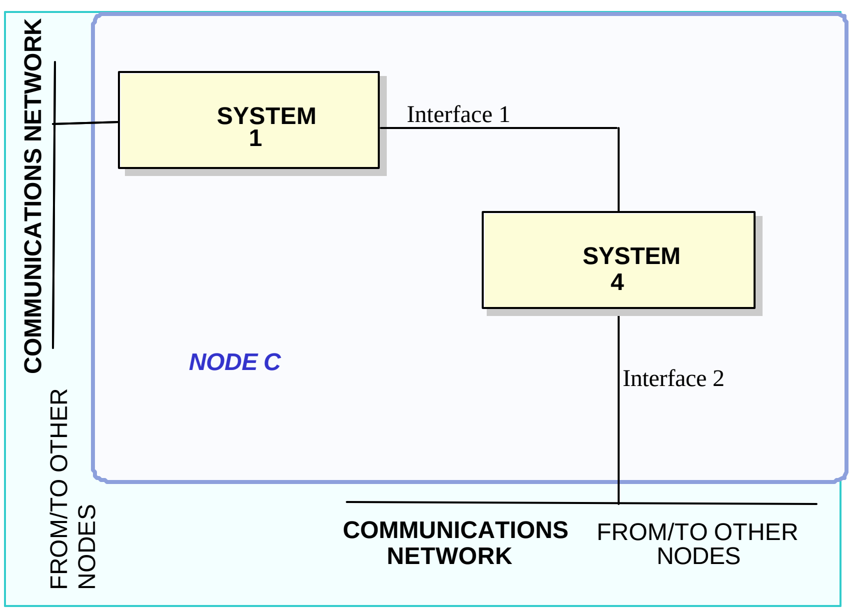


Figure 5-4. SV-1 Intranodal Version – Template

The intrasystem version can be used in 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 5-5 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.

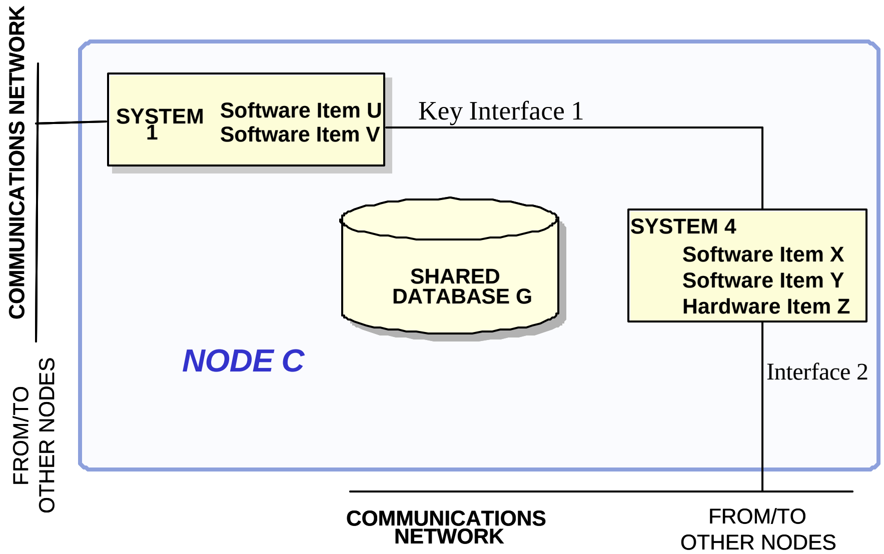


Figure 5-5. SV-1 Intrasystem Version – Example Showing a KI, a Database, and Other Software and Hardware Items

5.1.2 UML Representation

SV-1 may be developed in the Unified Modeling Language (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.

SV-1 internodal perspective showing systems may be modeled in UML using deployment diagrams that also show UML Components. This can be accomplished via the definition of new stereotypes for UML Components. Adding new stereotypes is neither necessary nor mandatory, as long as there is consistency in representing the same kind of nodes in each diagram (e.g., in any one diagram, all UML nodes are systems nodes, while all UML Components are systems that reside on the systems nodes). Components may be associated with notes (to specify allocations of resources, such as system functions, to the UML Components) and constraints.

Figure 5-6 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 (use case links).

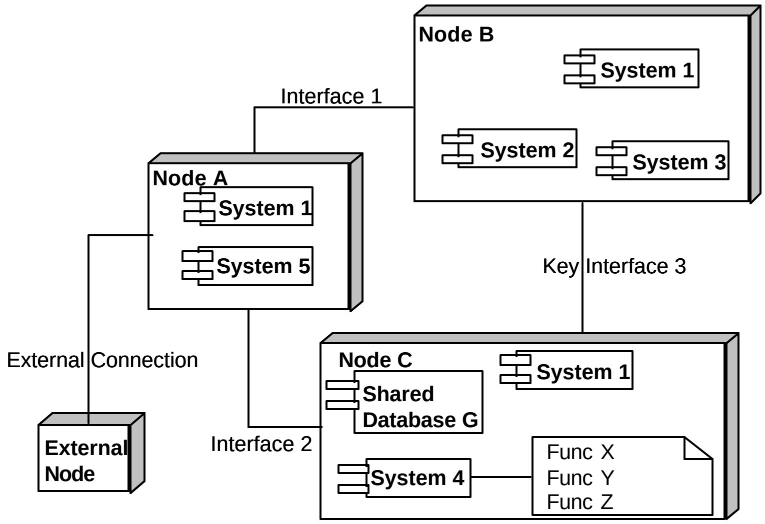


Figure 5-6. UML Node/Component Diagram for SV-1 Internodal Version Showing Systems – Template

Figure 5-7 is a template for the SV-1 internodal perspective showing system-to-system interfaces. SV-1 UML Components correlate to the systems. In UML, classes or use cases (SV-4 system functions) can be assigned to UML Components to allow system functions to be associated with systems in SV-1.

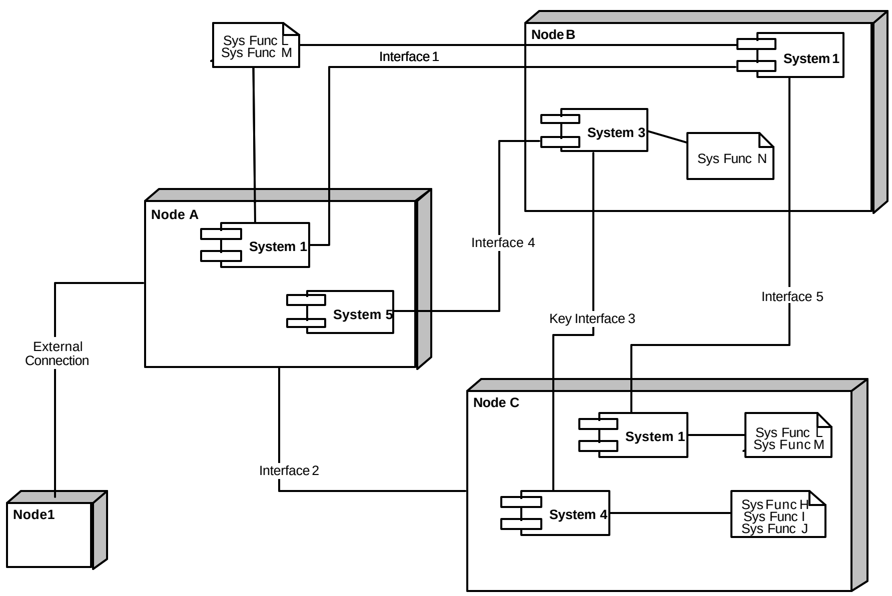


Figure 5-7. UML Node/Component Diagram for SV-1 Internodal Version Showing System-System Interfaces – Template

The SV-1 intranodal perspective may be modeled in UML using component diagrams with UML Components representing systems and notes to specify allocations of resources, such as system functions and hardware/software items, to the UML Components.

5.1.3 Systems Interface Description (SV-1) – Data Element Definitions

Table 5-1 describes the architecture data elements associated with Systems Interface Description.

Table 5-1. Data Element Definitions for Systems Interface Description (SV-1)

Data Elements	Attributes	Example Values/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
System*		(Appears in all versions) Representation of a system (FoS, SoS, subsystem, etc.)
	Name*	Name/identifier of system or subsystem
	Description*	Text summary of function or set of functions performed and constituent parts contained, e.g., 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 have 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 (e.g., hardware item); platform item (i.e., combined hardware and system software); system software; or application (i.e., mission unique) software
	Description	Text description of function(s) 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

Data Elements	Attributes	Example Values/Explanation
Relationships		
Systems Node Contains System*		
	Systems Node Name*	Name/identifier of a systems node
	System Name*	Name/identifier of a system resident at that node
System Contains (Sub)-System(s)*		
	System Name*	Name/identifier of a system
	(Sub)-System(s) 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 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

5.1.4 CADM Support for Systems Interface Description (SV-1)

SV-1 is stored as an instance of DOCUMENT with a category code designating it as a SYSTEM-INTERFACE-DESCRIPTION (a subtype of DOCUMENT). The properties of each section (for a specific interface) of the SYSTEM-INTERFACE-DESCRIPTION are specified using SYSTEM-INTERFACE-DESCRIPTION-ELEMENT.

SYSTEM-INTERFACE-DESCRIPTION-ELEMENT cites a specific system interface in terms of an instance of SYSTEM-ASSOCIATION-MEANS. SYSTEM-ASSOCIATION-MEANS, in turn, provides reference to a specific pair of SYSTEM instances (instance of SYSTEM-ASSOCIATION), a specific

TECHNICAL-INTERFACE, primary communications medium, a COMMUNICATION-SYSTEM, and an instance or AGREEMENT (e.g., an INFORMATION-TECHNOLOGY-STANDARD such as a MESSAGE-STANDARD, PROTOCOL-STANDARD or INFORMATION-TECHNOLOGY-STANDARD-PROFILE).

Each system reference in SV-1 is identified and described by populating SYSTEM. The set of SYSTEM instances addressed in a Systems View is listed by populating SYSTEM-SYSTEM-ARCHITECTURE (populating this table is optional). The set of systems related to a particular SYSTEM is specified by populating SYSTEM-ASSOCIATION. A SYSTEM is further described by populating its attributes and the attributes of SYSTEM-TYPE, as well as the attributes of the various (child) entities of SYSTEM in the CADM that include the following (not all of these appear in **Figure 5-8** provided below):

- COMMUNICATION-SYSTEM (a subtype of SYSTEM)
- INFORMATION-PROCESSING-SYSTEM (a subtype of SYSTEM that includes BATTLEFIELD-AUTOMATED-SYSTEM)
- SYSTEM-CAPABILITY
- SYSTEM-DIRECTED-CONSTRAINT
- SYSTEM-DOCUMENT
- SYSTEM-ELEMENT
- SYSTEM-EQUIPMENT-TYPE
- SYSTEM-INFORMATION-ASSET
- SYSTEM-INTERFACE-TYPE
- SYSTEM-ORGANIZATION
- SYSTEM-PROCESS-ACTIVITY
- SYSTEM-SATELLITE
- SYSTEM-CAVEATED-SECURITY-CLASSIFICATION
- SYSTEM-SOFTWARE-ITEM
- SYSTEM-STATUS
- SYSTEM-TRANSMISSION

For some architects, fundamental to SV-1 are the operational requirements for equipment and subordinate organizations for each ORGANIZATION-TYPE depicted in (or described by) SV-1. Tables of organization and establishment (TOEs) for an ORGANIZATION-TYPE are recorded in the architecture database through populating the following tables:

- ORGANIZATION-TYPE-ESTABLISHMENT
- ORGANIZATION-TYPE-ESTABLISHMENT (identifies a specific TOE)
- MATERIEL-ITEM (a class of MATERIEL)
- ORGANIZATION-TYPE-ESTABLISHMENT-MATERIEL-ITEM-DETAIL (lists quantities of a specific MATERIEL-ITEM authorized for a specific ORGANIZATION-TYPE in a specific ORGANIZATION-TYPE-ESTABLISHMENT)

- ORGANIZATION-TYPE-ESTABLISHMENT-ORGANIZATION-TYPE-DETAIL (lists quantities of a specific ORGANIZATION-TYPE authorized for a specific ORGANIZATION-TYPE in a specific ORGANIZATION-TYPE-ESTABLISHMENT)
- ORGANIZATION-TYPE-ESTABLISHMENT-ORGANIZATION-TYPE-DETAIL-ELEMENT
- ORGANIZATION-TYPE-ESTABLISHMENT-SYSTEM-DETAIL
- MATERIEL-ITEM-ESTABLISHMENT
- MATERIEL-ITEM-ESTABLISHMENT-MATERIEL-ITEM-DETAIL (lists quantities of a specific MATERIEL-ITEM, such as components or associated items of equipment, authorized for a specific MATERIEL-ITEM in a specific MATERIEL-ITEM-ESTABLISHMENT)
- MATERIEL-MATERIEL-ITEM-ESTABLISHMENT-MATERIEL-ITEM-DETAIL
- ORGANIZATION-TYPE-ESTABLISHMENT-CROSS-REFERENCE

The modified table of organization and establishment (MTOE) for an ORGANIZATION is recorded in a CADM-structured database through populating ORGANIZATION-ORGANIZATION-TYPE-ESTABLISHMENT. All the TOEs (and MTOEs) applicable to a specific ARCHITECTURE can be listed (optional) by populating ARCHITECTURE-ORGANIZATION-TYPE-ESTABLISHMENT.

Capabilities for a SYSTEM or MATERIEL-ITEM are specified in the CADM by populating the following entities: CAPABILITY, SYSTEM-CAPABILITY, and MATERIEL-ITEM-CAPABILITY-NORM.

A SYSTEM-INTERFACE-DESCRIPTION can apply to many architectures. All the ARCHITECTURE instances to which that DOCUMENT applies are specified in ARCHITECTURE-DOCUMENT. All the architecture products related to SV-1 (such as SV-2, SV-3, and SV-4) are related to SV-1 by populating DOCUMENT-ASSOCIATION (optional).

Figure 5-8 provides a high-level diagram from the CADM showing key entities that are used to store architecture data for SV-1 in a CADM-conformant database.

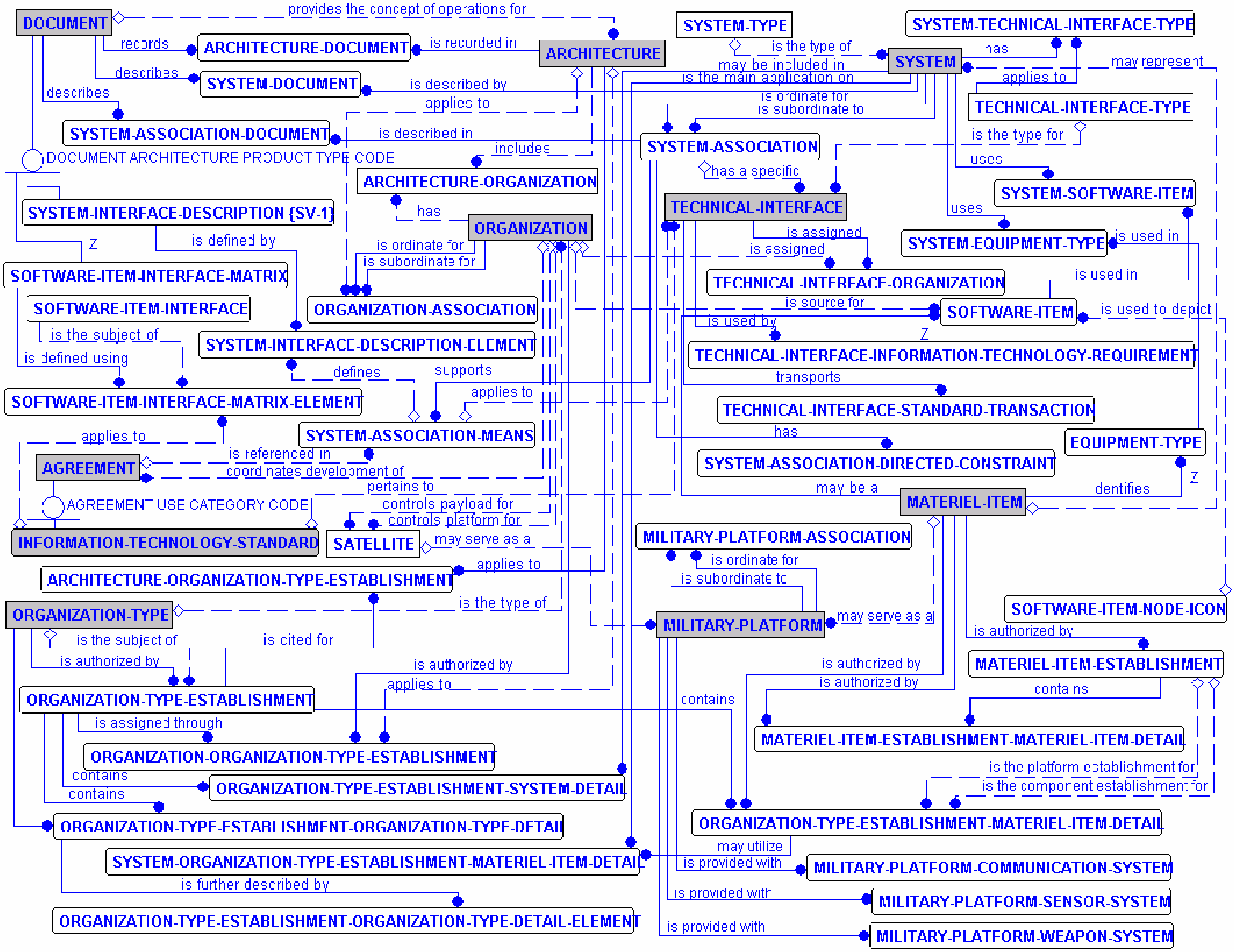


Figure 5-8. CADM ER Diagram for Systems Interface Description (SV-1)

5.2 SYSTEMS COMMUNICATIONS DESCRIPTION (SV-2)

5.2.1 Systems Communications Description (SV-2) – Product Description

Product Definition. The Systems Communications Description depicts pertinent information about communications systems, communications links, and communications networks. SV-2 documents the kinds of communications media that support the systems and implement their interfaces 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.

Product Purpose. SV-2 can be used 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.

Product Detailed Description. SV-2 documents the specific communications links or communications networks (e.g., Intelink or Joint Worldwide Intelligence Communications System [JWICS]) 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., composing parts of the implemented interface including communications systems, multiple communications links, communications networks, routers, and gateways).

Communications systems (e.g., switches, routers, and communications satellites) 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 systems (or node).

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

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.

Because SV-2 depicts the implementation details for the interfaces described in SV-1 by decomposing them into communications systems, communications links, and communications networks, it can present either internodal or intranodal versions. The internodal version details the communications links and/or communications networks that interconnect systems nodes (node edge to node edge) or specific systems (system-system from one node to other nodes). The intranodal version of SV-2 looks inside each of the represented nodes to illustrate the communications links between specific systems.

Figure 5-9 provides a template for the internodal version of SV-2. Note that Figure 5-9 translates the single-line representations of interfaces (as shown in the SV-1 internodal version) into an implementation detail of the communications infrastructure that provides the connections. The small boxes in Figure 5-9 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.

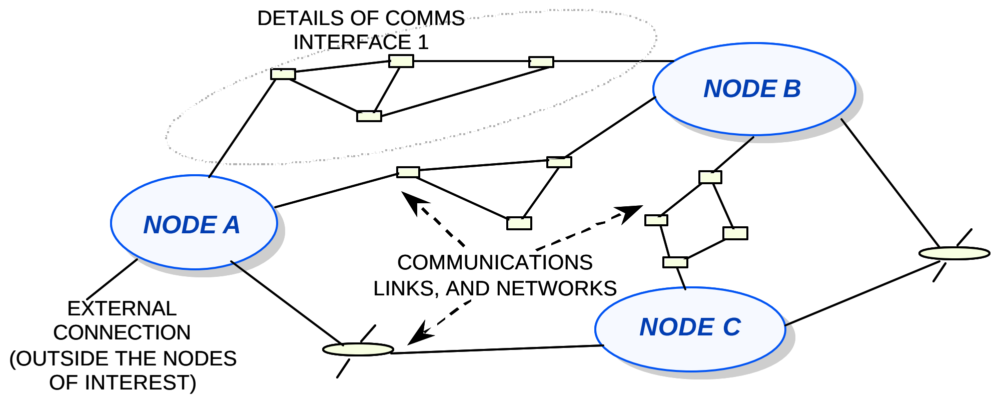


Figure 5-9. Systems Communications Description, Internodal Version (SV-2) – Template

Figure 5-10 provides a template for the intranodal version of SV-2.

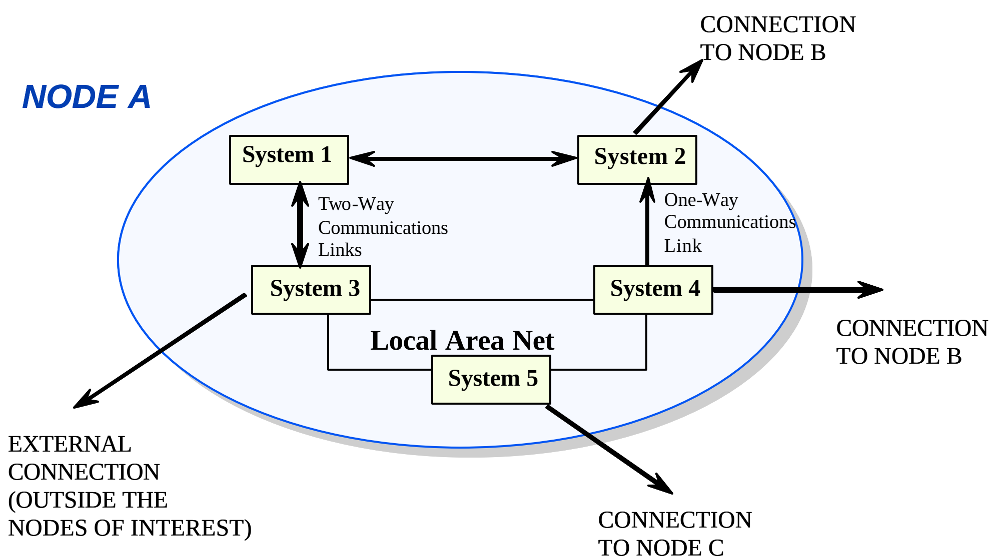


Figure 5-10. Systems Communications Description, Intranodal Version (SV-2) – Template

5.2.2 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, communications detail may be described in UML in the following manner.

SV-2 may be developed 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.

SV-2 intranodal perspective may be modeled in UML using component diagrams with UML Components representing communications systems and lines between components representing communications links.

5.2.3 Systems Communications Description (SV-2) – 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 5-2** describes the architecture data elements associated with SV-2.

Table 5-2. Data Element Definitions for Systems Communications Description (SV-2)

Data Elements	Attributes	Example Values/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
	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 (i.e., 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

Data Elements	Attributes	Example Values/Explanation
	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		
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 Link(s)*		
	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
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

Data Elements	Attributes	Example Values/Explanation
	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 (i.e., 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 Uses Standard*		For communications hardware/software item. 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

5.2.4 CADM Support for Systems Communication Description (SV-2)

SV-2 is stored as an instance of DOCUMENT with a category code designating it as a SYSTEM-COMMUNICATION-DESCRIPTION (a subtype of DOCUMENT). The properties of each section of the SYSTEM-COMMUNICATION-DESCRIPTION are specified using SYSTEM-COMMUNICATION-DESCRIPTION-ELEMENT. The subject of each SYSTEM-COMMUNICATION-DESCRIPTION-ELEMENT is a set of NODE instances and NODE-ASSOCIATION instances (which includes NODE-LINK instances) defined as a specific NETWORK. Details for SV-2 are recorded in CADM entities such as NETWORK-CAPABILITY, NETWORK-SYSTEM, NETWORK-PATH, NETWORK-

STANDARD-PROFILE, and NODE-ASSOCIATION-NETWORK. NODE-LINK-CAPABILITY record quantifiable measures of performance (MOPs) for a specific NODE-LINK.

Communication characteristics cited for a specific SV-2 are specified using the following entities, as applicable:

- COMMUNICATION-MEDIUM
- EQUIPMENT-TYPE (a subset of MATERIEL-ITEM)
- MATERIEL
- MATERIEL-ASSOCIATION
- MATERIEL-FIELDING
- MATERIEL-ORGANIZATION
- NODE-LINK-COMMUNICATION-MEDIUM
- NODE-MATERIEL
- NETWORK-DEVICE (a subset of MATERIEL)
- NETWORK-DEVICE-PROTOCOL-USAGE
- NETWORK-DEVICE-WORKSTATION
- NETWORK-DEVICE-WORKSTATION-SOFTWARE-ITEM
- NETWORK-PROTOCOL-USAGE
- SOFTWARE-ITEM (carries a MATERIEL-ITEM Identifier as its primary key attribute)
- SOFTWARE-ITEM-ASSOCIATION
- SOFTWARE-ITEM-INTERFACE
- SOFTWARE-ITEM-INTERFACE-MATRIX
- SOFTWARE-ITEM-INTERFACE-MATRIX-ELEMENT

Nodal diagrams prepared for SV-2 use the standard ICON-CATALOG in the CADM. The following subtypes are used in connection with the ICON-CATALOG to specify which fields shall be presented when users highlight (click) an icon (specified as an instance of GRAPHIC, which is a subtype of DOCUMENT): LINK-ICON and NODE-ICON.

Details for the nodal description of the communications are provided by populating the following entities (not all are shown in **Figure 5-11**):

- ARCHITECTURE-NETWORK
- ARCHITECTURE-NODE
- NETWORK
- NETWORK-CAPABILITY
- NETWORK-COMMUNICATION-MEDIUM
- NETWORK-DEMARCATION-POINT

- NETWORK-DETAIL
- NETWORK-NODE
- NETWORK-ORGANIZATION
- NETWORK-PATH
- NETWORK-PROTOCOL-USAGE
- NETWORK-STANDARD-PROFILE
- NETWORK-TYPE
- NODE
- NODE-ACTIVITY-MODEL-INFORMATION-ELEMENT-ROLE
- NODE-ACTIVITY-MODEL-THREAD {GIG}
- NODE-ASSOCIATION
- NODE-ASSOCIATION-NETWORK
- NODE-COMMUNICATION-MEDIUM
- NODE-DATA-ITEM-TYPE
- NODE-DETAIL
- NODE-FACILITY
- NODE-INFORMATION-ASSET
- NODE-LINK
- NODE-LINK-CAPABILITY
- NODE-LINK-COMMUNICATION-MEDIUM
- NODE-MATERIEL
- NODE-MISSION-AREA
- NODE-ORGANIZATION
- NODE-ORGANIZATION-TYPE
- NODE-PROCESS-ACTIVITY
- NODE-SYSTEM
- NODE-TASK

SV-2 can be populated with electronic addresses for each fielded communications device. These are specified, as available, with the following entities in the CADM:

- ADDRESS-BOOK (a subtype of DOCUMENT)
- ADDRESS-BOOK-ELEMENT (an entry in the ADDRESS-BOOK)
- ADDRESS-BOOK-ELEMENT
- INTERNET-ADDRESS
- INTERNET-ADDRESS-ASSOCIATION

- NETWORK-DEVICE-INTERNET-ADDRESS
- NETWORK-INTERNET-ADDRESSING
- NODE-INTERNET-ADDRESS

Because many other CADM entities involved in SV-2 are the same as for SV-1, their discussion is not repeated here.

Figure 5-11 provides a high-level diagram from the CADM showing key entities that are used to store architecture data for SV-2 in a CADM-conformant database.

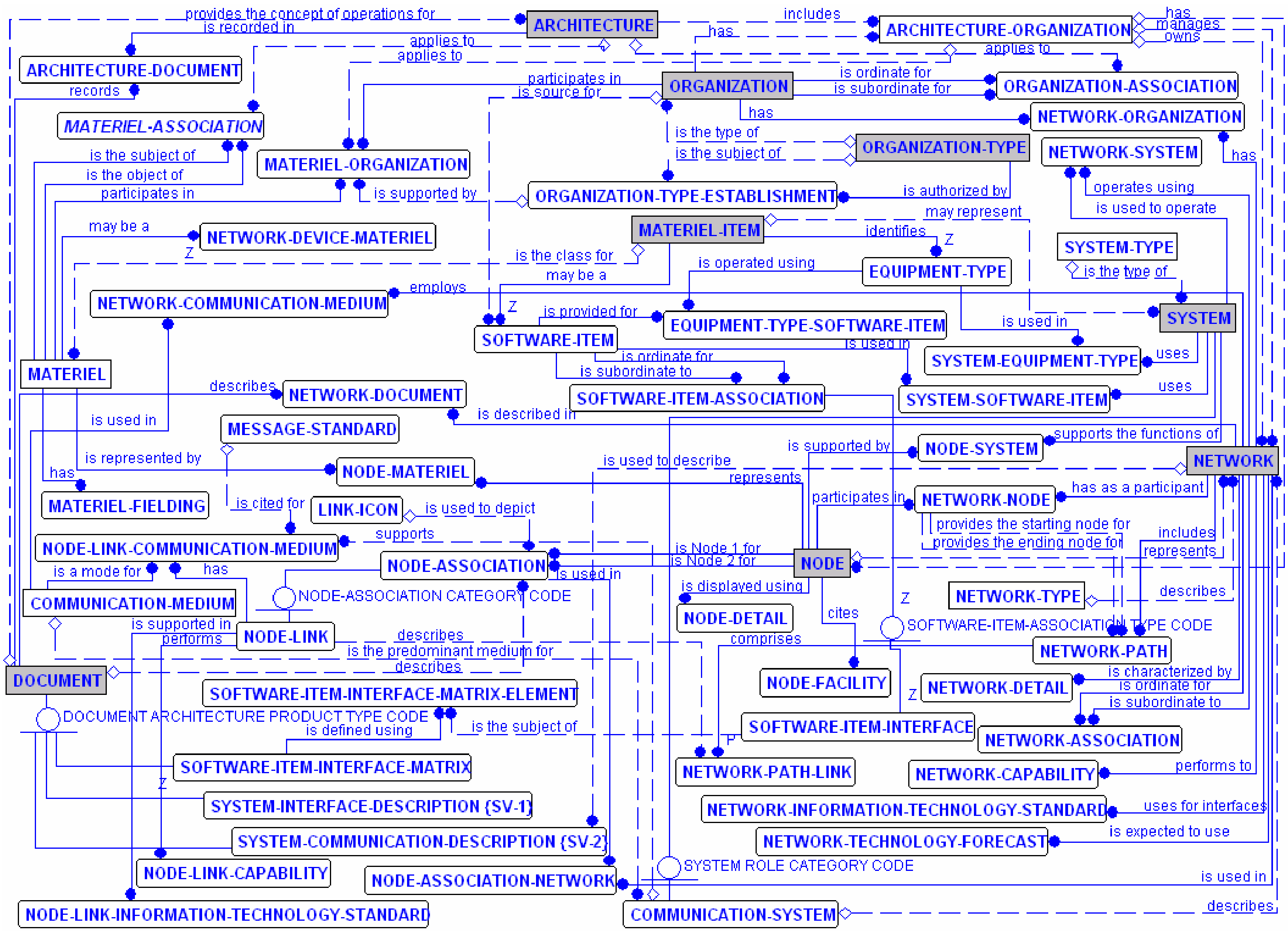


Figure 5-11. CADM ER Diagram for Systems Communications Description (SV-2)

5.3 SYSTEMS-SYSTEMS MATRIX (SV-3)

5.3.1 Systems-Systems Matrix (SV-3) – Product Description

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

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).

SV-3 can be organized in a number of ways (e.g., 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/functionality, and the redistribution of systems and processes in context with evolving operational requirements.

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. Many types of interface information can be presented in the cells of SV-3. The system-system interfaces can be represented using a number of different symbols and/or color codes that depict different interface characteristics. The following are examples:

- Status (e.g., existing, planned, potential, deactivated)
- Purpose (e.g., C2, intelligence, logistics)
- Classification level (e.g., SECRET, TS/SCI)
- Means (e.g., JWICS, SIPRNET)
- Standard
- Key Interface

Figure 5-12 provides a template for SV-3. Each cell may contain several interface characteristics, but this is not shown in the notional example provided below.

	SYSTEM 1	SYSTEM 2	SYSTEM 3	SYSTEM 4	SYSTEM 5	SYSTEM 6	SYSTEM 7	SYSTEM 8	SYSTEM 9	SYSTEM 10
SYSTEM 1		●								
SYSTEM 2	●		●	●	●	●				
SYSTEM 3		●		●	●	●				
SYSTEM 4		●	●		●	●				
SYSTEM 5		●	●	●		●				
SYSTEM 6	●	●	●	●	●		●	●	●	●
SYSTEM 7						●				
SYSTEM 8						●				
SYSTEM 9						●				
SYSTEM 10						●				

Figure 5-12. Systems-Systems Matrix (SV-3) – Template

5.3.2 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).

5.3.3 Systems-Systems Matrix (SV-3) – Data Element Definitions

Table 5-3 defines the architecture data elements for SV-3.

Table 5-3. Data Element Definitions for Systems-Systems Matrix (SV-3)

Data Elements	Attributes	Example Values/Explanation
Non-Graphical 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**
	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**
Relationship		
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 can not 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**

5.3.4 CADM Support for Systems-Systems Matrix (SV-3)

SV-3 is stored as an instance of DOCUMENT with a category code designating it as a SYSTEM-SYSTEM-MATRIX (a subtype of DOCUMENT). The properties of each cell of the SYSTEM-SYSTEM-MATRIX are specified using SYSTEM-SYSTEM-MATRIX-ELEMENT. While not mandatory in the CADM, the provider SYSTEM and the recipient SYSTEM are related to each other in a

CADM-structured database as one or more instances of SYSTEM-ASSOCIATION, with additional information about that association stored in SYSTEM-ASSOCIATION-MEANS and SYSTEM-ASSOCIATION-DIRECTED-CONSTRAINT.

All of the attributes of the following entities are available for use in specifying the properties of a cell in SV-3 from the references (foreign keys) in the entities SYSTEM-SYSTEM-MATRIX-ELEMENT, SYSTEM-ASSOCIATION, SYSTEM-ASSOCIATION-MEANS, and SYSTEM-ASSOCIATION-DIRECTED-CONSTRAINT:

- AGREEMENT (e.g., INFORMATION-TECHNOLOGY-STANDARD, to include MESSAGE-STANDARD, PROTOCOL-STANDARD, and INFORMATION-TECHNOLOGY-STANDARD-PROFILE)
- CAVEATED-SECURITY-CLASSIFICATION
- COMMUNICATION-MEDIUM
- COMMUNICATION-SYSTEM
- DIRECTED-CONSTRAINT (a subtype of GUIDANCE)
- SECURITY-CLASSIFICATION

Status, purpose, and other properties to be associated to the cells of SV-3 are specified in DATA-DOMAIN, DATA-DOMAIN-LIST, and DATA-DOMAIN-LIST-VALUE. The appropriate value for each cell is specified in the association entity SYSTEM-SYSTEM-MATRIX-ELEMENT-DATA-DOMAIN-LIST-VALUE.

Details for each Provider SYSTEM and Receiver SYSTEM, where applicable and required, are available in the following entities: SYSTEM-DIRECTED-CONSTRAINT, SYSTEM-STATUS, SYSTEM-STATUS-DEPENDENCY, SYSTEM-SYSTEM-ARCHITECTURE, and SYSTEM-TYPE.

A SYSTEM-SYSTEM-MATRIX can apply to many architectures. All the ARCHITECTURE instances to which that DOCUMENT applies are specified in ARCHITECTURE-DOCUMENT. All the architecture products related to SV-3 (such as SV-1, SV-2, and SV-4) are related to SV-3 by populating DOCUMENT-ASSOCIATION (optional).

Figure 5-13 provides a high-level diagram from the CADM showing key entities that are used to store architecture data for SV-3 in a CADM-conformant database.

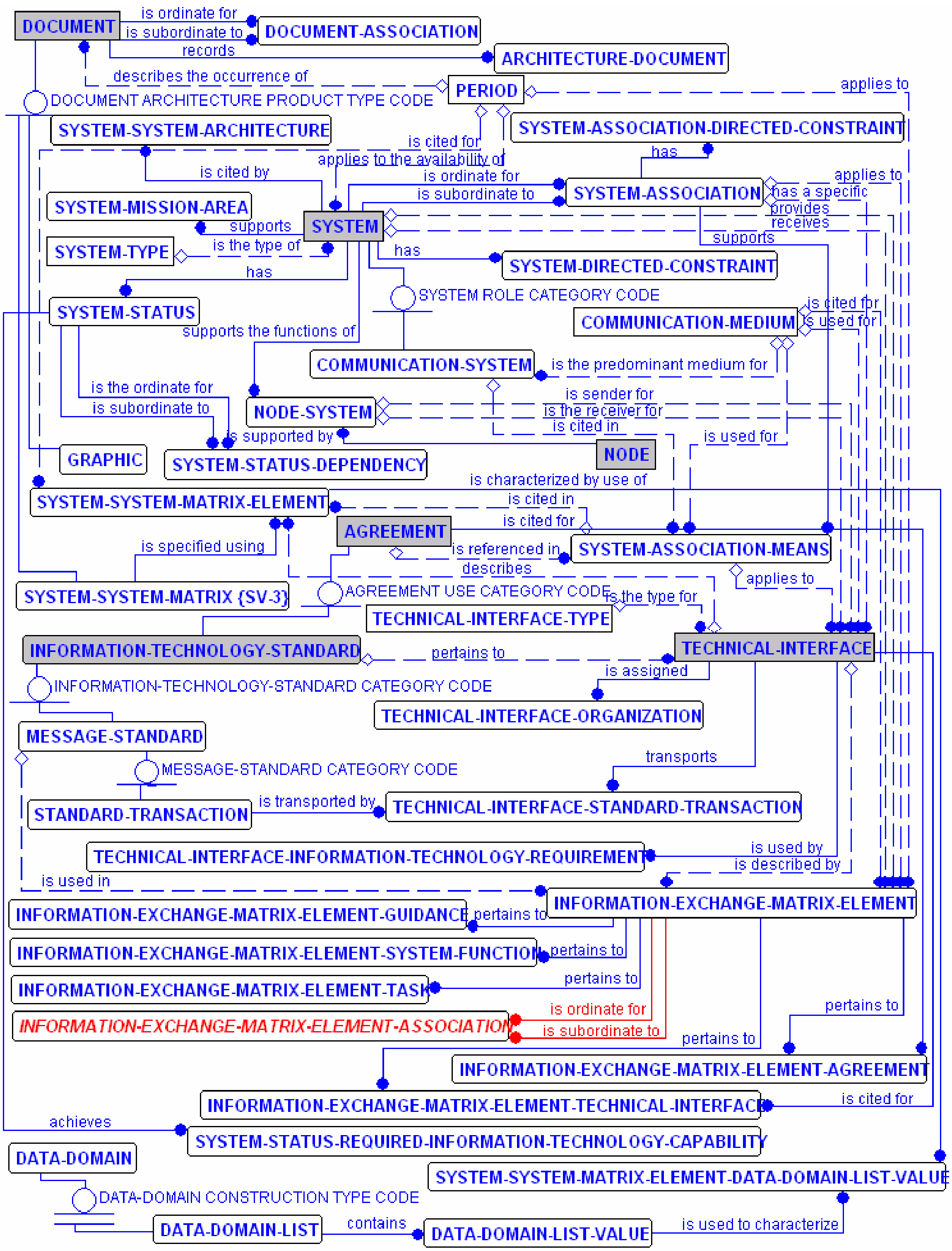


Figure 5-13. CADM ER Diagram for Systems-Systems Matrix (SV-3)

5.4 SYSTEMS FUNCTIONALITY DESCRIPTION (SV-4)

5.4.1 Systems Functionality Description (SV-4) – Product Description

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

Product Purpose. The primary purposes of SV-4 are to (a) develop a clear description of the necessary system data flows that are input (consumed) by and output (produced) by each system, (b) ensure that the functional connectivity is complete (i.e., that a system's required inputs are all satisfied), and (c) ensure that the functional decomposition reaches an appropriate level of detail.

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, system data flow without node considerations, function to system allocations, and function to node allocations.

The system functions documented in the SV-4 may be identified using the Service Component Reference Model (SRM),[20] or some other system function taxonomy, and correlated 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 or functions that consume or produce system data from/to system functions that belong to external systems. The external system data sources and/or sinks can be used to represent the human that interacts with the system or external systems. The system data flows between the external system data source/sink (representing the human or system) and the HCI, GUI, or interface function can be used to represent human-system interactions, or system-system interfaces. Standards that apply to system functions, such as HCI and GUI standards, are also specified in this product.

Like OV-5, SV-4 may be hierarchical in nature and 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 5-14** shows a template for a functional decomposition model of SV-4.

---

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

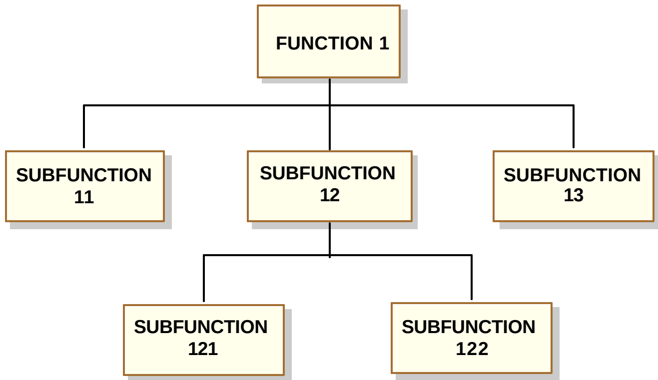


Figure 5-14. Systems Functionality Description (SV-4) – Template (Functional Decomposition)

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 5-15** shows a template for the 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. Internal system data repositories are represented by parallel lines. SV-4s can be arranged hierarchically, with each system function at the parent level being decomposed into a child SV-4 at the next level.

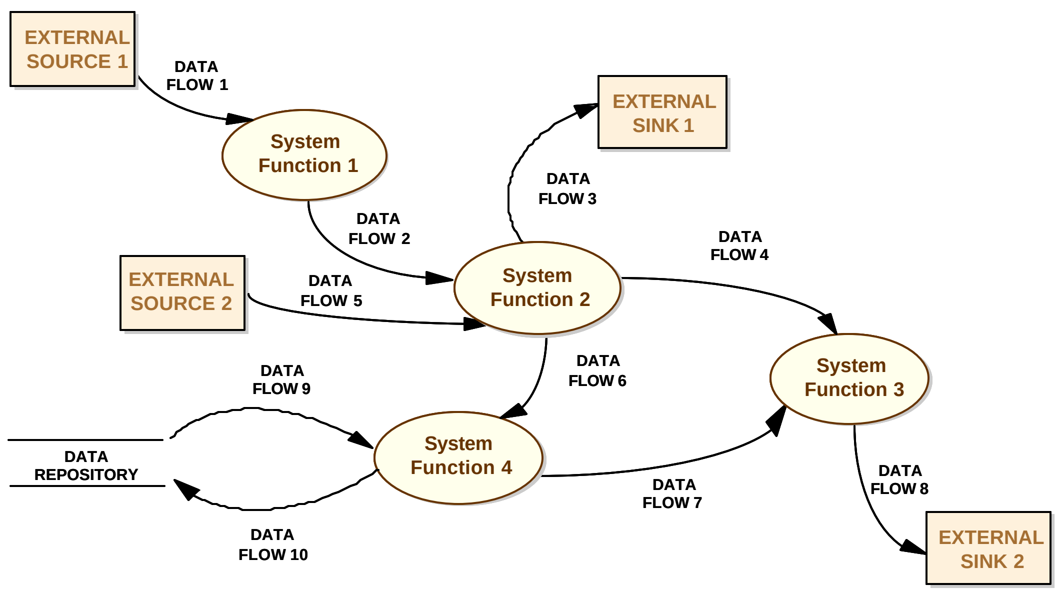


Figure 5-15. Systems Functionality Description (SV-4) – Template (Data Flow Diagram)

5.4.2 UML Representation

Use cases, classes, and class operations may be used to represent system functions in SV-4. Use case diagrams are used to 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), and 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 (correspond to those in OV-5 use case diagrams), and other systems external to the system being described but not necessarily external to the architecture. They are the equivalent of external sources and sinks in DFD notation. In addition, class diagrams are used to 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 5-16** is a sample SV-4 template of a UML use case diagram. **Figure 5-17** shows the corresponding class diagram.

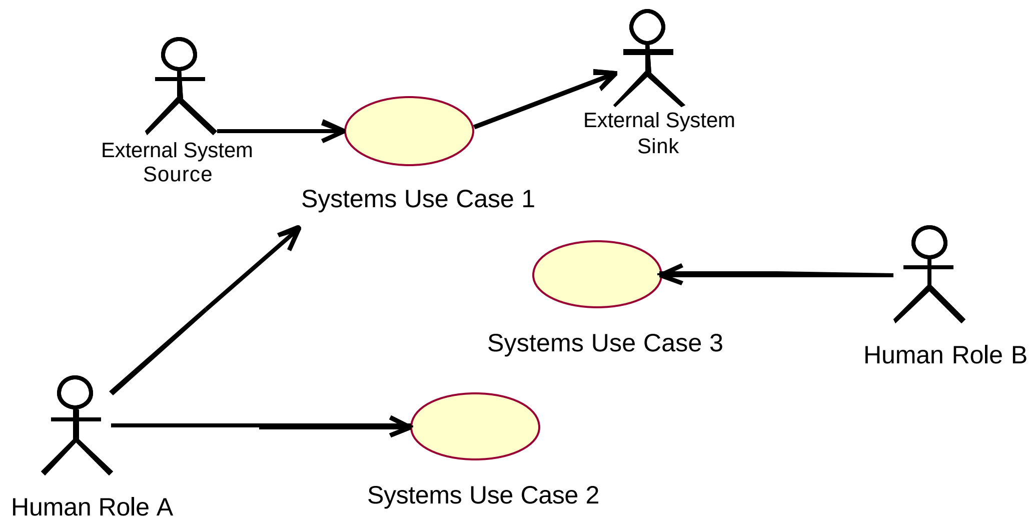


Figure 5-16. UML Use Case Diagram for Systems Functionality Description (SV-4)

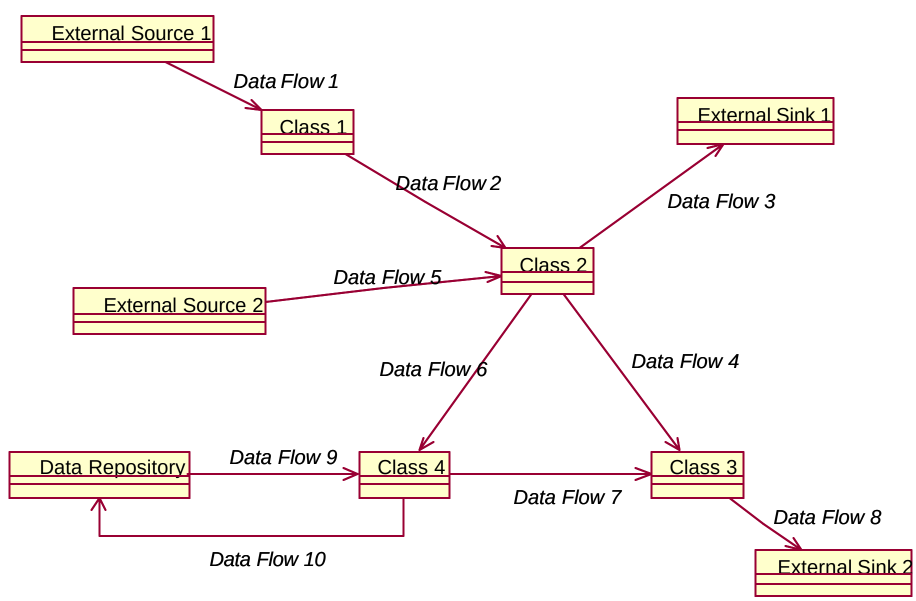


Figure 5-17. UML Class Diagram Showing System Functions and Relationships

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 case(s) 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 be used to 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. Attributes and operations for these classes (that show associated system data types and their permissible operations) may be specified or left out depending on the level of implementation detail needed. Packages and containment relationships can be used in conjunction with use case and class diagrams to show functional decomposition. **Figure 5-18** depicts relationships between SV-4 use cases and the collection (packages) of classes that implement them.

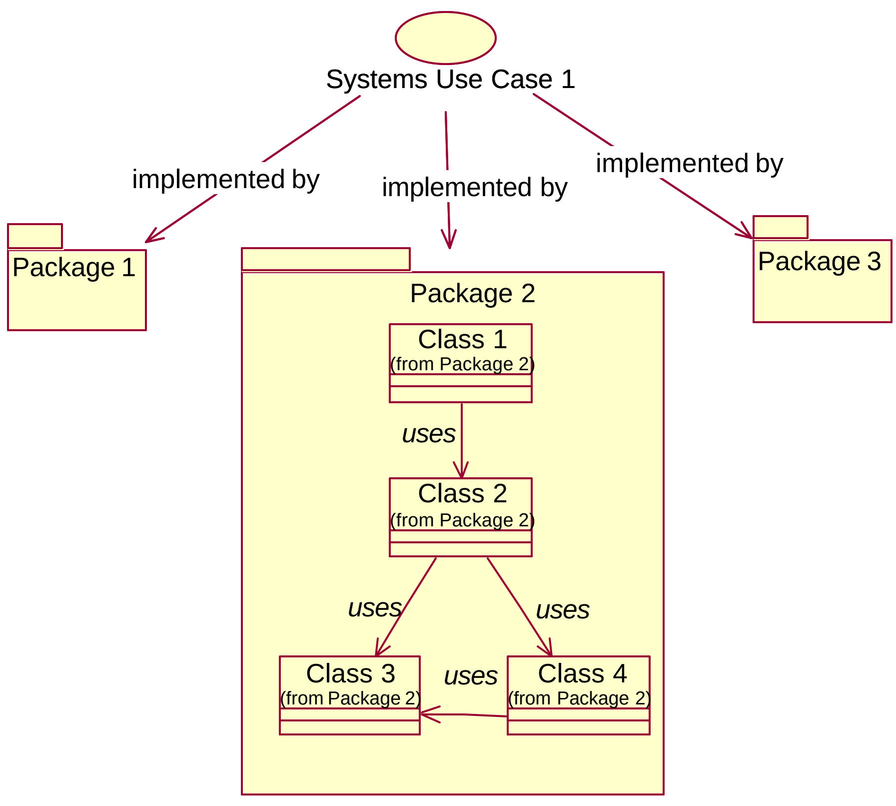


Figure 5-18. UML Class Diagram for Systems Functionality Description (SV-4)

5.4.3 Systems Functionality Description (SV-4) – Data Element Definitions

Table 5-4 defines the architecture data elements for SV-4.

Table 5-4. Data Element Definitions for Systems Functionality Description (SV-4)

Data Elements	Attributes	Example Values/Explanation
Graphical Box Types		
System Function*		See SV-1 Definition Table
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 system(s) 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
	Sub-Function	Name/Identifier of system sub-function into which the super-function decomposes
Referenced Types		
Systems Node		See SV-1 Definition Table
System (or Subsystem)*		See SV-1 Definition Table
Relationships		
System Data Repository is Sink 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 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

Data Elements	Attributes	Example Values/Explanation
	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
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 JTA standard (e.g., HCI or GUI standard used)

5.4.4 CADM Support for Systems Functionality Description (SV-4)

SV-4 is stored as an instance of DOCUMENT with a category code designating it as a SYSTEM-FUNCTIONALITY-SPECIFICATION (a subtype of DOCUMENT). DFDs can be specified by using the same entities and attributes as for ACTIVITY-MODEL. The specification of the activity model is thus stored as an instance of INFORMATION-ASSET with a category code designating it as an ACTIVITY-MODEL (a subtype of INFORMATION-ASSET). The key entities associated with defining system functions and their relationships to ACTIVITY-MODEL are shown in **Figure 5-19**.

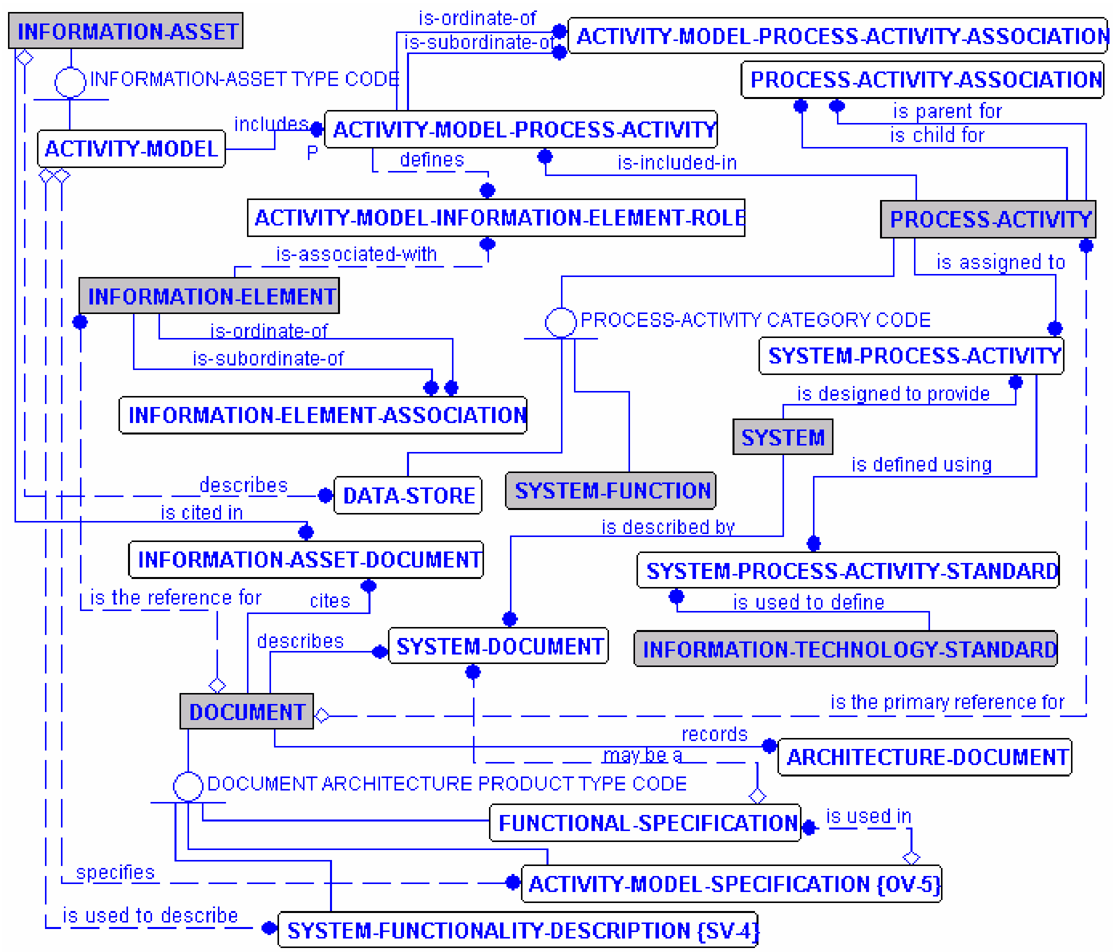


Figure 5-19. CADM ER Diagram for Systems Functionality Description (SV-4)

A SYSTEM-FUNCTIONALITY-SPECIFICATION can apply to many architectures. All the ARCHITECTURE instances to which that DOCUMENT applies are specified in ARCHITECTURE-DOCUMENT.

Each function and subfunction cited in the ACTIVITY-MODEL for a SV-4 is stored as an instance of PROCESS-ACTIVITY with a category code designating it, as appropriate, as a SYSTEM-FUNCTION (a subtype of PROCESS-ACTIVITY), and its occurrence in the ACTIVITY-MODEL for the DFD is stored as an instance of ACTIVITY-MODEL-PROCESS-ACTIVITY. The hierarchical breakdown of the functions in a specific SV-4 is specified by instances of ACTIVITY-MODEL-PROCESS-ACTIVITY-ASSOCIATION. Note that different SV-4s can depict *different* hierarchical breakdowns of the same system functions; thus, PROCESS-ACTIVITY-ASSOCIATION is *not* used for this purpose.

A unique element of a SV-4 not found in an IDEF0 activity modeling is a system data store, which is used as the source or destination (sink) of an information flow in the form of an information repository. For convenience and consistency, DATA-STORE has been incorporated in the CADM as an additional subtype of PROCESS-ACTIVITY. In addition, for modelers who wish to have an explicit source and destination for *every* information flow, two further categories of PROCESS-ACTIVITY are defined: (1) Start State/Condition and (2) End State/Condition (subtypes are not defined since these currently have no special attributes). These same two values are also used to identify an external source PROCESS-ACTIVITY (start condition) and an external sink PROCESS-ACTIVITY (end condition) when they are not considered data stores.

The information flows and their component information flows cited in the ACTIVITY-MODEL for a SV-4 are stored as instances of INFORMATION-ELEMENT, and their occurrences in the DFD for a specific ACTIVITY-MODEL-PROCESS-ACTIVITY are stored as an instance of ACTIVITY-MODEL-INFORMATION-ELEMENT-ROLE. As noted in the discussion for OV-5, the decomposition of an INFORMATION-ELEMENT into its components is specified by using INFORMATION-ELEMENT-ASSOCIATION (such decomposition is *not* dependent on the activity model or on the activities specified in the activity model but is expected to be consistent across activity models).

As each INFORMATION-ELEMENT occurs in a SV-4 for a specific function, using MODEL-INFORMATION-ELEMENT-ROLE, there are two roles that can occur: input and output. Note that these roles are for the *use* of the INFORMATION-ELEMENT in relation to a specific ACTIVITY-MODEL-PROCESS-ACTIVITY and are not directly associated to the INFORMATION-ELEMENT itself. The two roles are specified by use of an attribute (ACTIVITY-MODEL-INFORMATION-ELEMENT-ROLE Type Code) to designate exactly one of the following subtypes:

- ACTIVITY-MODEL-INFORMATION-ELEMENT-ROLE-INPUT
- ACTIVITY-MODEL-INFORMATION-ELEMENT-ROLE-OUTPUT

It is optional to incorporate some or all of the details of a SV-4 into a CADM-structured database. INFORMATION-ELEMENT and SYSTEM-FUNCTION should be populated, together with associations between them, so that the system functions and information flows required for SV-5 and SV-6, respectively, are available for cross reference. Because system functions are specified by use of SYSTEM-FUNCTION, a subtype of PROCESS-ACTIVITY, the entity PROCESS-ACTIVITY-ASSOCIATION is used to record the cross reference of SYSTEM-FUNCTION instances to other instances of PROCESS-ACTIVITY.

A key element of operational requirement traceability is the relation of SYSTEM-FUNCTION instances and other PROCESS-ACTIVITY instances to TASK instances. This traceability is provided by populating PROCESS-ACTIVITY-TASK.

Documents cited for details of an ACTIVITY-MODEL (or other subtypes of INFORMATION-ASSET) defining a SV-4 are stored as instances of INFORMATION-ASSET-DOCUMENT.

Figure 5-20 provides a core diagram from the CADM showing key entities that are used to store system functions in relation to human factors and lines of business.

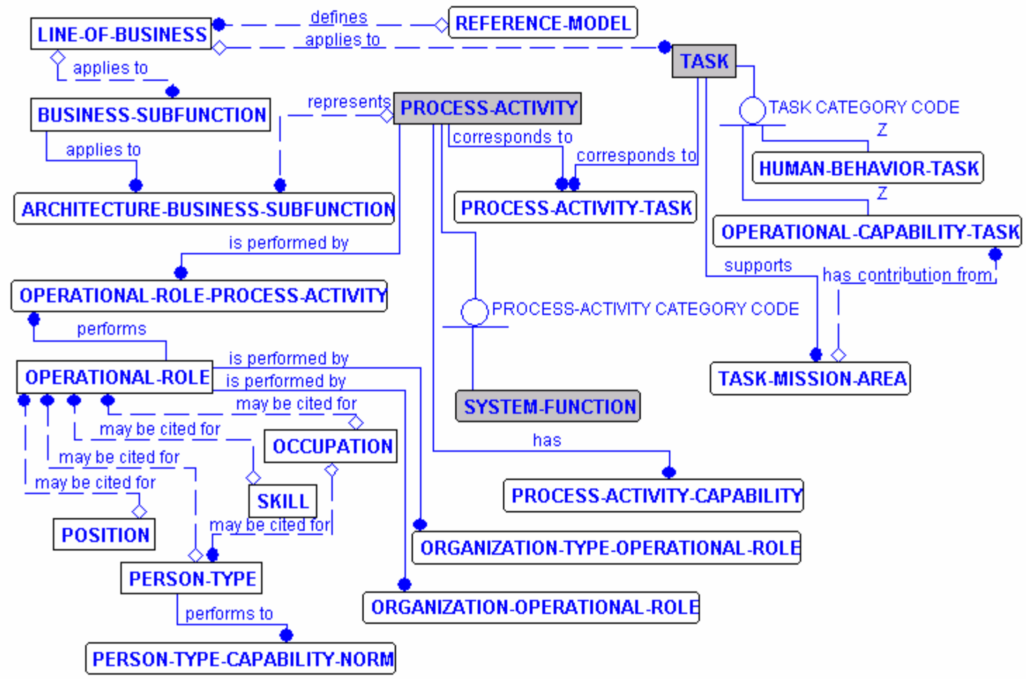


Figure 5-20. CADM Human Factors and Lines and Business for SV-4

Figure 5-21 provides a high-level diagram from the CADM showing key entities that are used to store data for the SYSTEM-FUNCTIONALITY-SPECIFICATION architecture product in a CADM-conformant database.

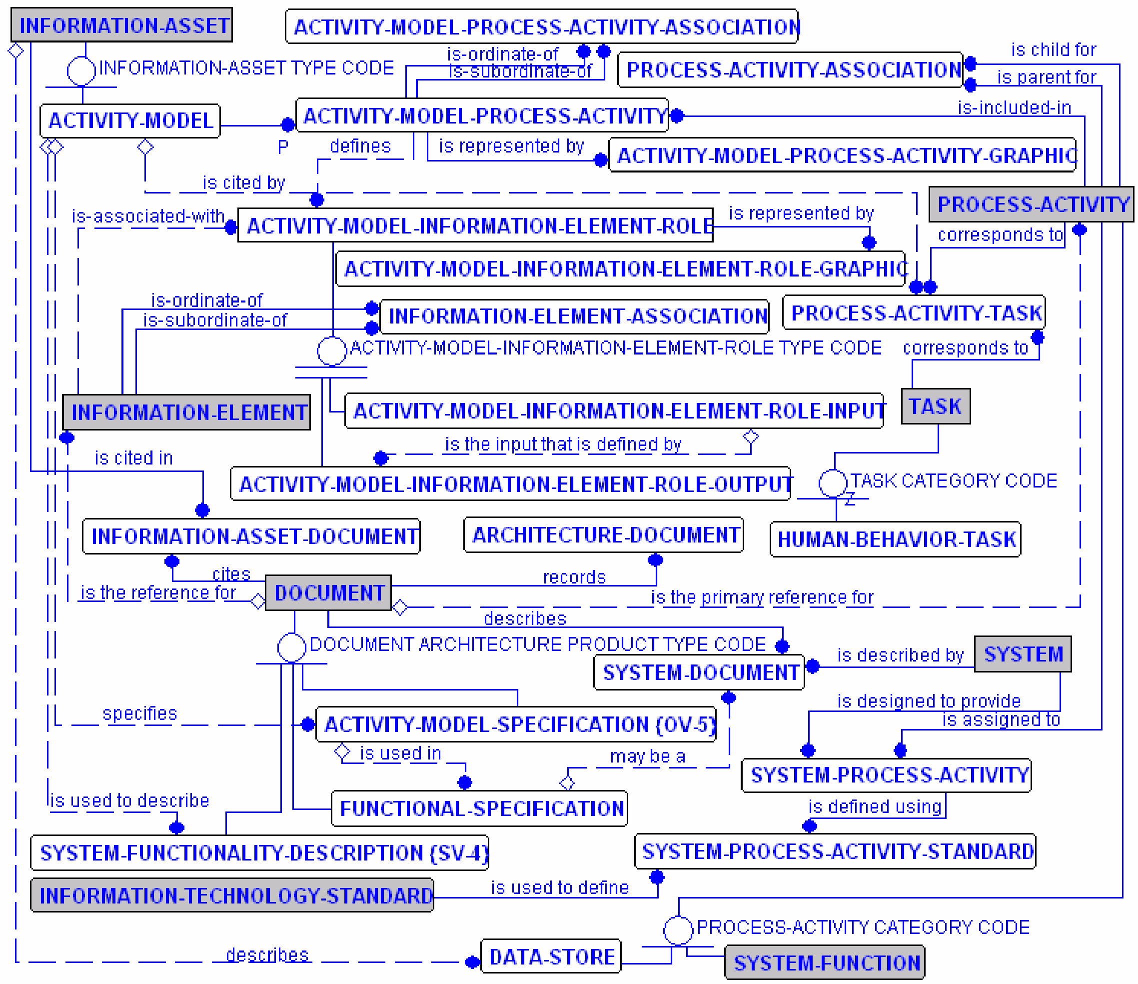


Figure 5-21. CADM ER Diagram for Systems Functionality Description (SV-4)

5.5 OPERATIONAL ACTIVITY TO SYSTEMS FUNCTION TRACEABILITY MATRIX (SV-5)

5.5.1 Operational Activity to Systems Function Traceability Matrix (SV-5) – Product Description

Product Definition. Operational Activity to Systems Function Traceability Matrix 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.

Product Purpose. SV-5 depicts the mapping of operational activities to system functions and thus identifies the transformation of an operational need into a purposeful action performed by a system.

SV-5 can be extended to depict the mapping of capabilities to operational activities, operational activities to system functions, system functions to systems, and thus relates the capabilities to the systems that support them. Such a matrix allows decision makers and planners to quickly identify stovepiped systems, redundant/duplicative systems, gaps in capability, and possible future investment strategies all in accordance with the time stamp given to the architecture. SV-5 correlates capability requirements that would *not* be satisfied if a specific system is *not* fielded to a specific DoD unit.

Product Detailed Description. The Framework 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, accept inputs, and 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 Framework'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 (i.e., one operational activity may be supported by multiple system functions, and one system function may support multiple operational activities). **Figure 5-22** provides a notional example of SV-5.

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		

Figure 5-22. Operational Activity to Systems Function Traceability Matrix (SV-5)

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 materiel that support them) in order to achieve a given mission objective. Capability-related attributes may be associated with specific activities or with the information exchange between activities, or both. A particular operational thread may be used to depict a capability. An operational thread is defined as 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. SV-5 can be used 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, the activities are related to the system functions that automate them (wholly or partially). Then a set of activities are associated 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), SV-5 can be used as the 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.

Each mapping between an operational activity to a system function is described by a stoplight colored circle 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 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.

Figure 5-23 provides a notional example of the extended SV-5 that provides a mapping between capabilities and systems.

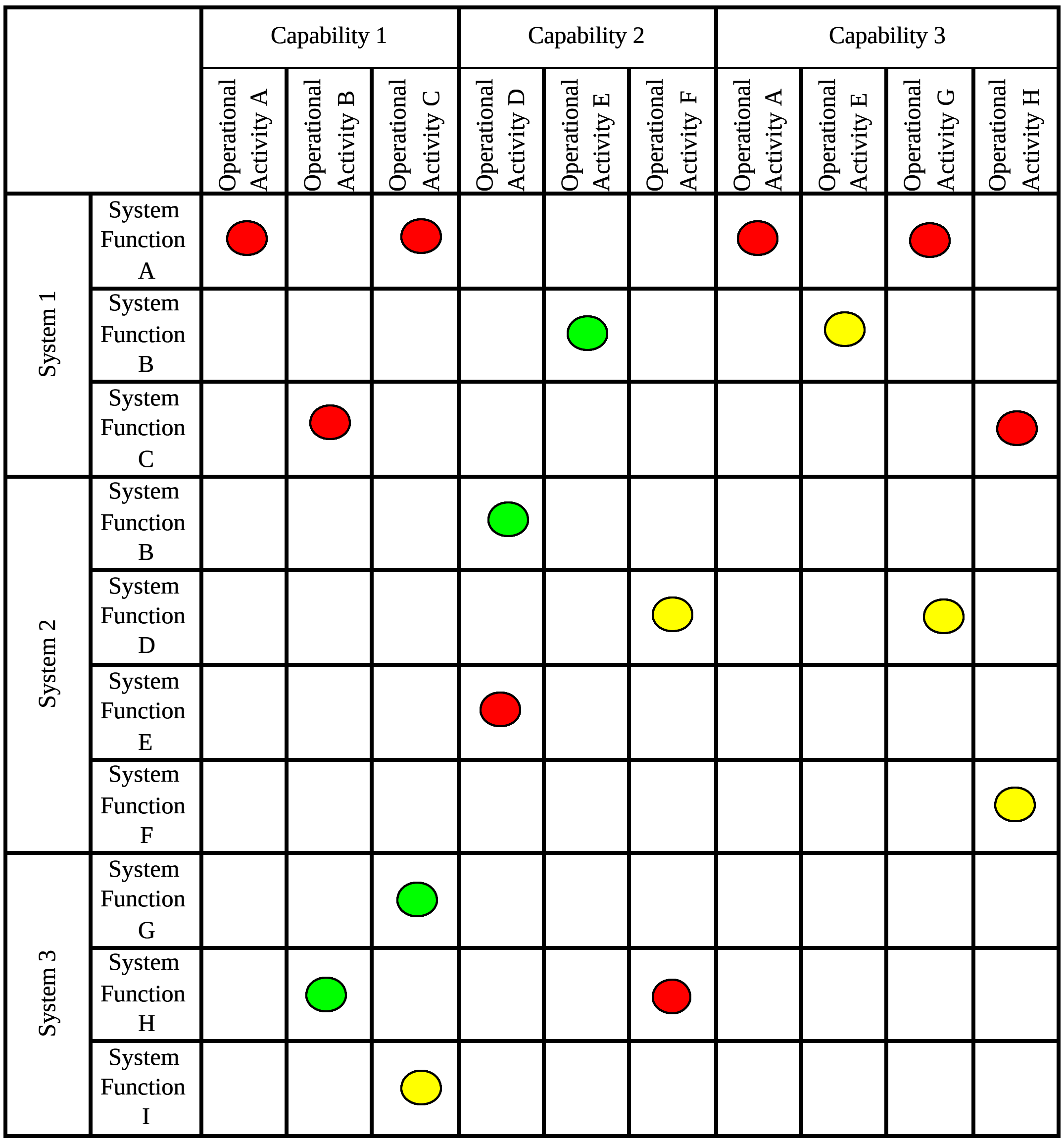


Figure 5-23. Capability to System Traceability Matrix (SV-5)

5.5.2 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 adornments (e.g., <<include>> relationships between OV-5 use cases and SV-4 use cases). A UML modeling tool script may be used 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. System functions can also be allocated to systems in UML by allocating SV-4 classes and packages to UML Components (from SV-1) accomplishing the intent of SV-5.

5.5.3 Operational Activity to Systems Function Traceability Matrix (SV-5) – Data Element Definitions

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

Table 5-5. Data Element Definitions for Operational Activity to Systems Function Traceability Matrix (SV-5)

Data Elements	Attributes	Example Values/Explanation
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; and 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 (at least partially) implemented by the system function

5.5.4 CADM Support for Operational Activity to Systems Function Traceability Matrix (SV-5)

SV-5 is stored as an instance of DOCUMENT with a category code designating it as a SYSTEM-FUNCTION-TRACEABILITY-MATRIX (a subtype of DOCUMENT). The properties of each cell of the SYSTEM-FUNCTION-TRACEABILITY-MATRIX are specified using SYSTEM-FUNCTION-TRACEABILITY-MATRIX-ELEMENT.

A SYSTEM-FUNCTION-TRACEABILITY-MATRIX can apply to many architectures. All the ARCHITECTURE instances to which that DOCUMENT applies are specified in ARCHITECTURE-DOCUMENT.

Each function cited in the SYSTEM-FUNCTION-TRACEABILITY-MATRIX is stored as an instance of PROCESS-ACTIVITY with a category code designating it as a SYSTEM-FUNCTION (a subtype of PROCESS-ACTIVITY).

SYSTEM-PROCESS-ACTIVITY identifies all the SYSTEM-FUNCTION instances and other PROCESS-ACTIVITY instances that are associated with a specific SYSTEM. Each such association is provided one of the following type codes: 1 = Is designed to carry out; 2 = Performs (i.e., has a function represented by); 3 = Provides partial support for; or 4 = Provides an alternate capability for.

A key element of operational requirement traceability is the relation of SYSTEM-FUNCTION instances and other PROCESS-ACTIVITY instances to TASK instances. This traceability is provided by populating PROCESS-ACTIVITY-TASK.

The CADM provides two levels of association of System Functions to Operational Activities. The first is independent of architecture and architecture product and thus applies globally in all situations. This global level of association is stored as instances of PROCESS-ACTIVITY-ASSOCIATION and can be designated by a code in this entity as the following kind of association: *Is derived from; Is identical to; Is part of; Directly supports; and Indirectly supports* (additional types of association can be defined and agreed to by modifying the values for PROCESS-ACTIVITY-ASSOCIATION Role Code). Ideally, the global level of association of System Functions to Operational Activities can be specified as System Functions or Operational Activities are defined (whichever comes second). At a minimum, the global association would address specific Operational Activities that would *not* be supported if *no* system providing the System Functions is fielded.

The second level of association can be specific to an architecture and its architecture product(s). This architecture-level association is stored as instances of SYSTEM-FUNCTION-TRACEABILITY-MATRIX-ELEMENT. At present, the architecture-level association is weak in that it only establishes an association but does not categorize it (in contrast to the global level of association).

Each row of the SYSTEM-FUNCTION-TRACEABILITY-MATRIX-ELEMENT has all the properties of the following entities that are directly referenced (by foreign key attributes) in this entity:

- Operational PROCESS-ACTIVITY
- SYSTEM
- System PROCESS-ACTIVITY
- SYSTEM-PROCESS-ACTIVITY
- TASK

Figure 5-24 provides a high-level diagram from the CADM showing key entities that are used to store architecture data for SV-5 in a CADM-conformant database.

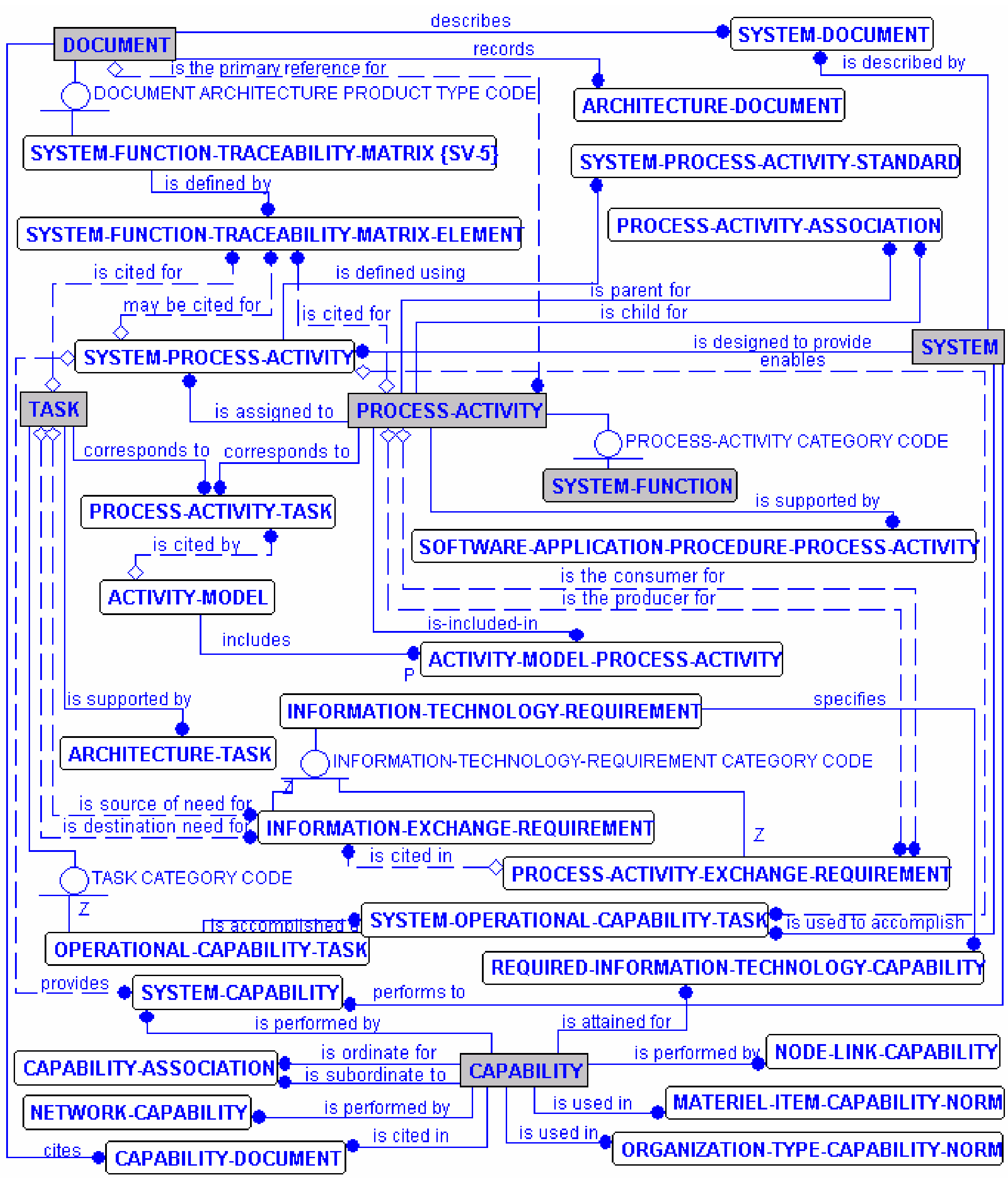


Figure 5-24. CADM ER Diagram for Operational Activity to Systems Function Traceability Matrix (SV-5)

5.6 SYSTEMS DATA EXCHANGE MATRIX (SV-6)

5.6.1 Systems Data Exchange Matrix (SV-6) – Product Description

Product Definition. The Systems Data Exchange Matrix 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.

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) and 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 and are critical to understanding the potential for overhead and constraints introduced by the physical aspects of the implementation.

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 security characteristics of the exchange. In addition, the system data elements, their format and media type, accuracy, units of measurement, and system data standard are also described in the matrix.

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, the Levels of Information Systems Interoperability (LISI) level required for the operational information exchange is replaced by the LISI level achieved through the system data exchange(s). Similarly, performance attributes for the operational information exchanges are replaced by the actual system data exchange performance attributes for the automated portion(s) 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 and constitute the automated portion(s) of the OV-3 information elements.

A partial format for the SV-6 matrix can be found in CJCSI 6212.01B, and that format is required for C4ISP development. However additions to the CJCSI 6212.01B matrix to meet program-unique needs should also be allowed. **Figure 5-25** shows a template for this product. The data element definition table for SV-6 contains detailed descriptions or references for each matrix column.

Interface Identifier	Data Exchange Identifier	Data Description							Producer		Consumer		Nature of Transaction			
System Interface Name and Identifier	System Data Exchange Name and Identifier	Data Element Name and Identifier	Content	Format Type	Media Type	Accuracy	Units of Measurement	Data Standard	Sending System Name and Identifier	Sending System Function Name and Identifier	Receiving System Name and Identifier	Receiving System Function Name and Identifier	Transaction Type	Triggering Event	Interoperability Level Achieved	Criticality

Interface Identifier	Data Exchange Identifier	Performance Attributes				Information Assurance							Security				
System Interface Name and Identifier	System Data Exchange Name and Identifier	Periodicity	Timeliness	Throughput	Size	Access Control	Availability	Confidentiality	Dissemination Control	Integrity	Non-Repudiation Producer	Non-Repudiation Consumer	Protection (Type Name, Duration, Date)	Classification	Classification Caveat	Releasability	Security Standard

Figure 5-25. Systems Data Exchange Matrix (SV-6) – Template

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 (i.e., 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, and multiple system data flows may map to a single system data element.

5.6.2 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 an automated tool is used to create the other products in UML, then SV-6 expanded definitions can be generated from adornments to the applicable SV-1 systems and interfaces and SV-4 system functions and system data flows.

5.6.3 Systems Data Exchange Matrix (SV-6) – Data Element Definitions

SV-6 Template (Figure 5-25) illustrates how the end product should look to the user. **Table 5-6** describes the architecture data elements for SV-6.

Table 5-6. Data Element Definitions for Systems Data Exchange Matrix (SV-6)

Data Elements	Attributes	Example Values/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 event(s) that triggers the system data exchange
	Interoperability Level Achievable	LISI achieved or achievable through the exchange
	Periodicity*	Frequency of system data exchange transmission – may be expressed in terms of worst case or average frequency
	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
	IA – Access Control	The class of mechanisms used to ensure only those authorized can access a specific system data element
	IA – Availability	The relative level of effort required to be expended to ensure that the system data can be accessed
	IA – Confidentiality	The kind of protection required for system data to prevent unintended disclosure
	IA – Dissemination Control	The kind of restrictions on receivers of system data based on sensitivity of system data
	IA – Integrity	The kind of requirements for checks that the content of the system data element has not been altered
	IA – Non-Repudiation Consumer	The requirements for unassailable knowledge that the system data sent was consumed by the intended recipient
	IA – Non-Repudiation Producer	The requirements for unassailable knowledge that the system data received was produced by the stated source
	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
	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

Data Elements	Attributes	Example Values/Explanation
	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/DTD, 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*	e.g., DoD XML Registry (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
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
	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

Data Elements	Attributes	Example Values/Explanation
	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

5.6.4 CADM Support for Systems Data Exchange Matrix (SV-6)

SV-6 is stored as an instance of DOCUMENT with a category code designating it as an INFORMATION-EXCHANGE-MATRIX (a subtype of DOCUMENT). Each row of SV-6 is specified as an instance of INFORMATION-EXCHANGE-MATRIX-ELEMENT. The details of each row are specified by the references (foreign keys) contained in INFORMATION-EXCHANGE-MATRIX-ELEMENT (if a complete OV-3 is provided, SV-6 may be limited to providing implementation-level references only, highlighted in bold):

- COMMUNICATION-MEDIUM Identifier
- Data Item MATERIEL-ITEM Identifier
- Exchange Needline Requirement GUIDANCE Identifier (links to two instances of ORGANIZATION and/or ORGANIZATION-TYPE)
- Information Exchange Requirement GUIDANCE Identifier (whose identity provides links between rows of OV-3 and SV-6 for the same IER)
- Information Requirement GUIDANCE Identifier (links to INFORMATION-ELEMENT)
- **INTERFACE Identifier**
- Message Standard AGREEMENT Identifier
- Process Activity Exchange Requirement GUIDANCE Identifier (links to two instances of PROCESS-ACTIVITY, each of which can be a SYSTEM-FUNCTION)
- **Provider Software Item MATERIEL-ITEM Identifier**
- **Provider SYSTEM Identifier**
- **Recipient Software Item MATERIEL-ITEM Identifier**
- **Recipient SYSTEM Identifier**
- **Time Frame PERIOD Identifier**

Using these references, all the attributes of GUIDANCE, INFORMATION-TECHNOLOGY-REQUIREMENT, INFORMATION-EXCHANGE-REQUIREMENT, EXCHANGE-NEEDLINE-REQUIREMENT, INFORMATION-REQUIREMENT, AGREEMENT, INFORMATION-TECHNOLOGY-STANDARD, and MESSAGE-STANDARD can be inserted into SV-6 from a CADM-structured database. In addition, these entities contain references (foreign keys) that allow all of the attributes of the following additional entities to be inserted into SV-6 (implementation-level references are highlighted in bold):

- **Alternate COMMUNICATION-SYSTEM**
- **Alternate Transmission MATERIEL-ITEM**
- CAVEATED-SECURITY-CLASSIFICATION
- Consumer Battlefield Function TASK
- Consumer DEPLOYMENT-LOCATION-TYPE
- Consumer ORGANIZATION-TYPE (e.g., OPERATIONAL-FACILITY)
- DATA-ITEM-TYPE
- Destination ORGANIZATION
- Destination ORGANIZATION-TYPE
- Destination Resourcing ORGANIZATION
- Destination TASK
- EXCHANGE-RELATIONSHIP-TYPE
- INFORMATION-EXCHANGE-REQUIREMENT-ELEMENT
- INFORMATION-EXCHANGE-REQUIREMENT-ELEMENT-DEPLOYMENT-MISSION-TYPE
- INFORMATION-EXCHANGE-REQUIREMENT-ELEMENT-DEPLOYMENT-PHASE
- **INFORMATION-EXCHANGE-REQUIREMENT-ELEMENT-METHOD**
- INFORMATION-EXCHANGE-REQUIREMENT-ELEMENT-PRODUCT
- INFORMATION-ELEMENT
- **NETWORK**
- NODE-ASSOCIATION
- **Primary COMMUNICATION-SYSTEM**
- **Primary Transmission MATERIEL-ITEM**
- Producer Battlefield Function TASK
- Producer DEPLOYMENT-LOCATION-TYPE
- Producer ORGANIZATION-TYPE (e.g., OPERATIONAL-FACILITY)
- **Receiver INFORMATION-PROCESSING-SYSTEM**
- Receiver ORGANIZATION-TYPE (e.g., OPERATIONAL-FACILITY)
- **Receiver Terminal MATERIEL-ITEM**

- SECURITY-CLASSIFICATION
- **Sender INFORMATION-PROCESSING-SYSTEM**
- Sender ORGANIZATION-TYPE (e.g., OPERATIONAL-FACILITY)
- **Sender Terminal MATERIEL-ITEM**
- Source ORGANIZATION
- Source ORGANIZATION-TYPE
- Source Resourcing ORGANIZATION
- Source TASK
- **TRANSMISSION-MEANS-TYPE**

The relation (e.g., function of the sender, function of the receiver) of a SYSTEM-FUNCTION is specified using INFORMATION-EXCHANGE-MATRIX-ELEMENT-SYSTEM-FUNCTION. Relationships of a specific row in an SV-6 to AGREEMENT (e.g., standard), GUIDANCE, TASK, and TECHNICAL-INTERFACE are specified using the following entities:

- INFORMATION-EXCHANGE-MATRIX-ELEMENT-AGREEMENT
- INFORMATION-EXCHANGE-MATRIX-ELEMENT-GUIDANCE
- INFORMATION-EXCHANGE-MATRIX-ELEMENT-TASK
- INFORMATION-EXCHANGE-MATRIX-ELEMENT-TECHNICAL-INTERFACE

Note that both OV-3 and SV-6 cite instances of INFORMATION-EXCHANGE-REQUIREMENT. OV-3 specifies in one or many rows of the matrix what is required. SV-6 specifies in one or many rows how each instance of INFORMATION-EXCHANGE-REQUIREMENT and, thereby, the rows of OV-3 are supported by the Systems View. The correlation is maintained by citing exactly the same INFORMATION-EXCHANGE-REQUIREMENT.

In addition, note that in a net-centric environment one but not necessarily both of the Provider SYSTEM and Receiver SYSTEM may be known. The associated identifiers in such a case would be null.

Figure 5-26 provides a high-level diagram from the CADM showing key entities that are used to store architecture data for SV-6 in a CADM-conformant database.

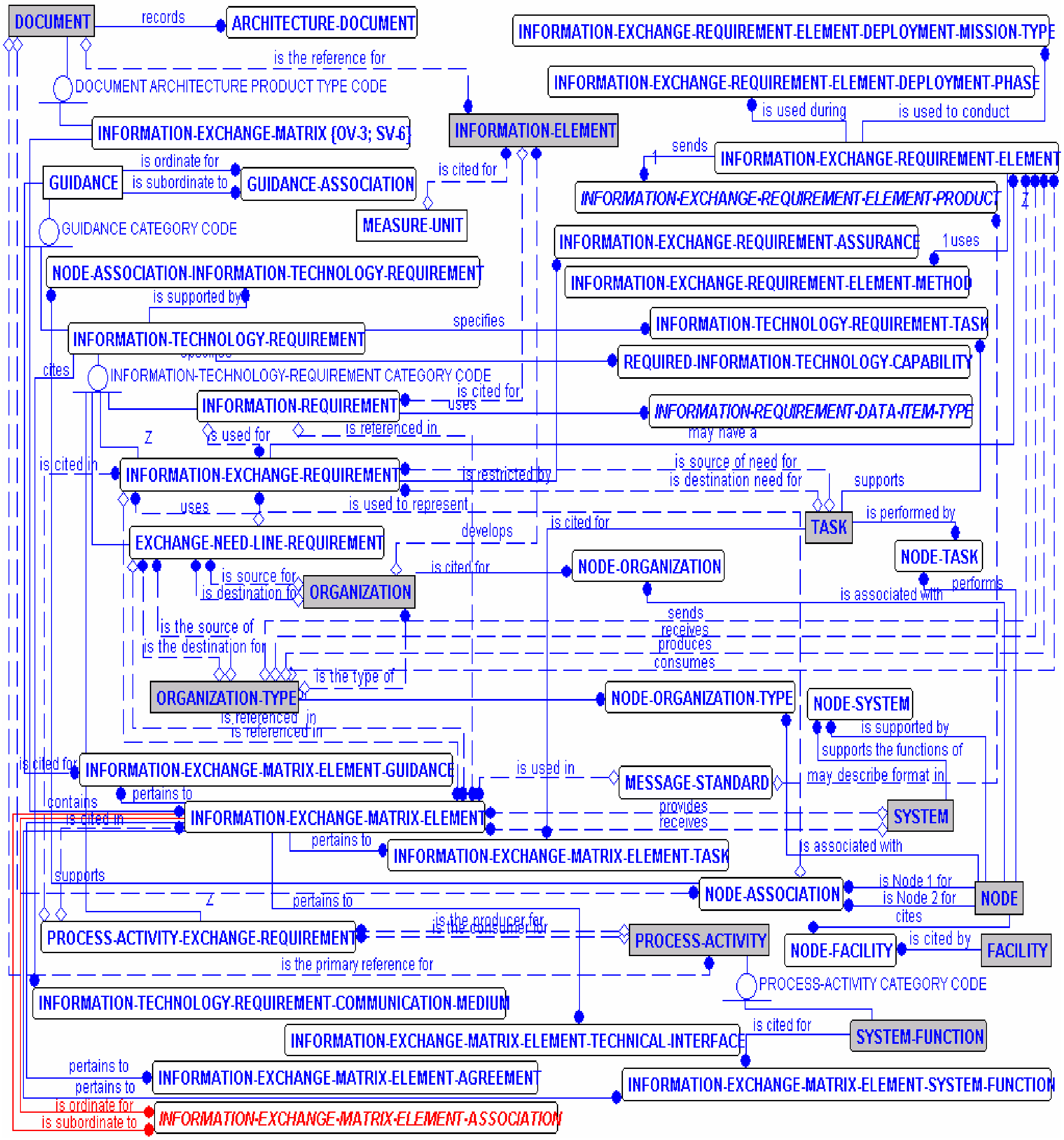


Figure 5-26. CADM ER Diagram for Systems Data Exchange Matrix (SV-6)

5.7 SYSTEMS PERFORMANCE PARAMETERS MATRIX (SV-7)

5.7.1 Systems Performance Parameters Matrix (SV-7) – Product Description

Product Definition. The Systems Performance Parameters Matrix product specifies the quantitative characteristics of systems and system hardware/software items, their interfaces (system data carried by the interface as well as communications link details that implement the interface), and their functions. It 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 requirements can be developed and specification defined. The complete set of performance parameters may not be known at the early stages of architecture definition, so it should be expected that this product will be updated throughout the system's specification, design, development, testing, and possibly even its deployment and operations life-cycle phases.

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, and will figure strongly in systems analyses and simulations done to support the acquisition decision processes and system design refinement.

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/software items and their interfaces (defined in SV-1), communications details (defined in SV-2), their functions (defined in SV-4), and their system data exchanges (defined in SV-6). The term *system*, as defined for this product and all others in the Framework, may represent a family of systems (FoS), system of systems (SoS), network of systems, or an individual system. Performance parameters for system hardware/software items (the hardware and software elements comprising a system) are also described in this product. In addition, performance parameters often relate to a system function being performed. Therefore, system functions and their performance attributes may also be shown in this product. If the future performance expectations are based on expected technology improvements, then the performance parameters and their time periods should be coordinated with a Systems Technology Forecast (SV-9). If performance improvements are associated with an overall system evolution or migration plan, then the time periods in SV-7 should be coordinated with the milestones in an Systems Evolution Description (SV-8).

Figure 5-27 is a template of SV-7, listing notional performance characteristics with a time period association.

	Performance Range (Threshold and Objective) Measures		
	$Time_0$ (Baseline Architecture Time Period)	$Time_1$	$Time_n$ (Target Architecture Time Period)
System Name			
Hardware Element 1			
Maintainability			
Availability			
System Initialization Time			
Architecture data Transfer Rate			
Program Restart Time			
S/W Element 1 / H/W Element 1			
Architecture Data Capacity (e.g., throughput or # of input types)			
Automatic Processing Responses (by input type, # processed/unit time)			
Operator Interaction Response Times (by type)			
Availability			
Effectiveness			
Mean Time Between S/W Failures			
Organic Training			
S/W Element 2 / H/W Element 1			
Hardware Element 2			

Figure 5-27. Systems Performance Parameters Matrix (SV-7) – Notional Example

5.7.2 UML Representation

There is no equivalent to this product in UML.

5.7.3 Systems Performance Parameters Matrix (SV-7) – Data Element Definitions

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

Table 5-7. Data Element Definitions for Systems Performance Matrix (SV-7)

Data Elements	Attributes	Example values/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/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)

Data Elements	Attributes	Example values/explanation
	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/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*		
	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 changes to objective values in future increments
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		
	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

Data Elements	Attributes	Example values/explanation
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		
	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

5.7.4 CADM Support for Systems Performance Parameters Matrix (SV-7)

Figure 5-28 provides a high-level diagram from the CADM showing key entities that are used to store architecture data for SV-7 in a CADM-conformant database. Each SV-7 is represented as an instance of SYSTEM-PERFORMANCE-PARAMETER-MATRIX, a subtype of DOCUMENT, which has attributes for Name, Time Frame, Category Code (with domain value Systems Performance Parameters Matrix), and so forth.

SV-7 is an instance of SYSTEM-PERFORMANCE-PARAMETER-MATRIX, which is defined by one or more instances of SYSTEM-PERFORMANCE-PARAMETER-MATRIX-ELEMENT. The latter contains identifiers for a DIRECTED-CONSTRAINT, a SYSTEM, a SYSTEM-CAPABILITY, and a Time

Frame PERIOD. Each SYSTEM has separate specifications in terms of its EQUIPMENT-TYPE instances (through SYSTEM-EQUIPMENT-TYPE and the Equipment Type MATERIEL-ITEM Identifier), SOFTWARE-ITEM instances (through SYSTEM-SOFTWARE-ITEM and the Software Item MATERIEL-ITEM Identifier), and capabilities (MATERIEL-ITEM-CAPABILITY-NORM, SYSTEM-CAPABILITY).

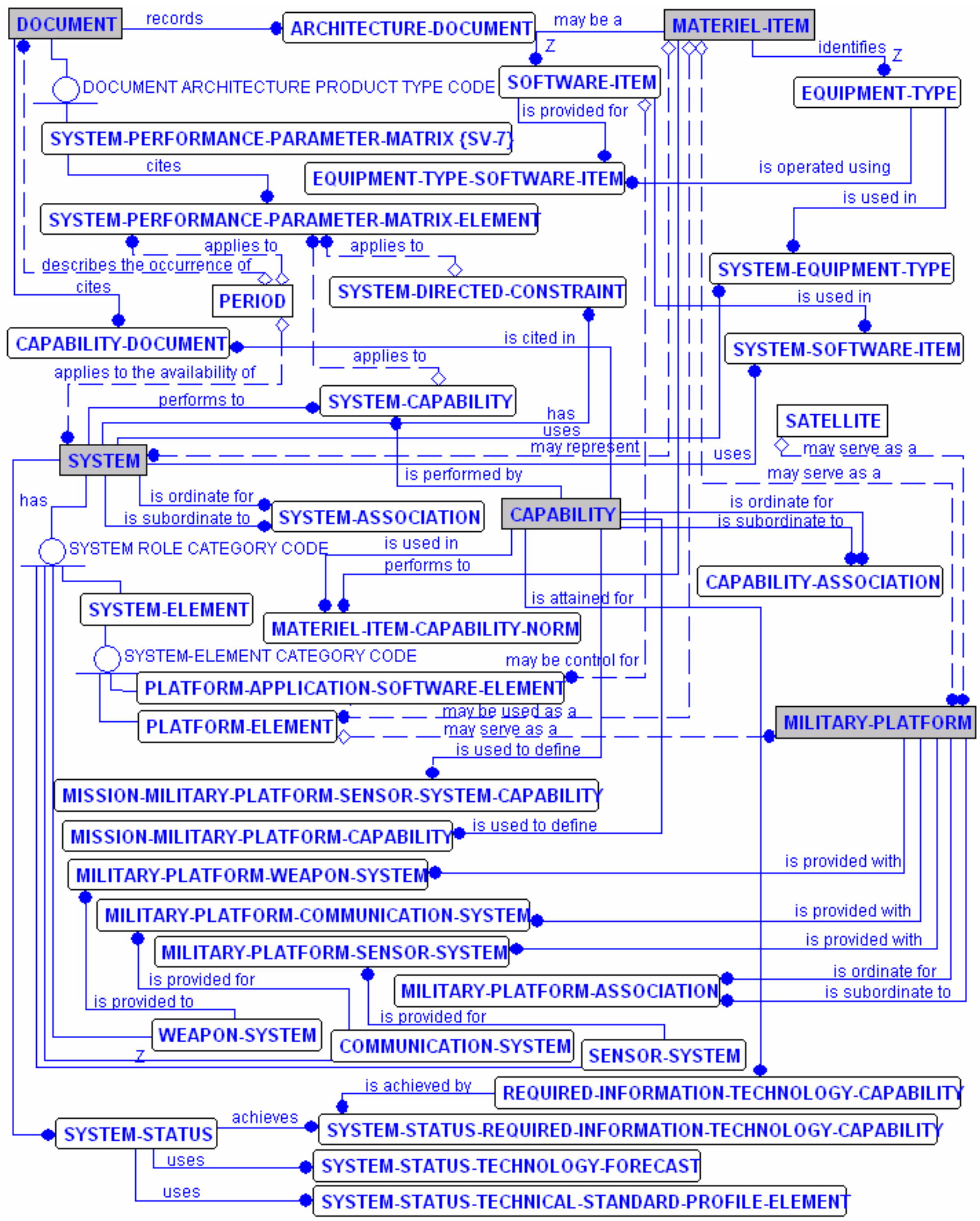


Figure 5-28. CADM ER Diagram for Technology Systems Performance Parameters Matrix (SV-7)

5.8 SYSTEMS EVOLUTION DESCRIPTION (SV-8)

5.8.1 Systems Evolution Description (SV-8) – Product Description

Product Definition. The Systems Evolution Description 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.

Product Purpose. SV-8, when linked together 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, the product can be used as an architecture evolution project plan or transition plan.

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) and will often combine the two thrusts. 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 a planned evolution or migration of communications systems, communication links, and their associated standards can also be described in this product. **Figure 5-29** illustrates a migration description, while **Figure 5-30** illustrates evolution. All entries in the graphics are for illustration only.

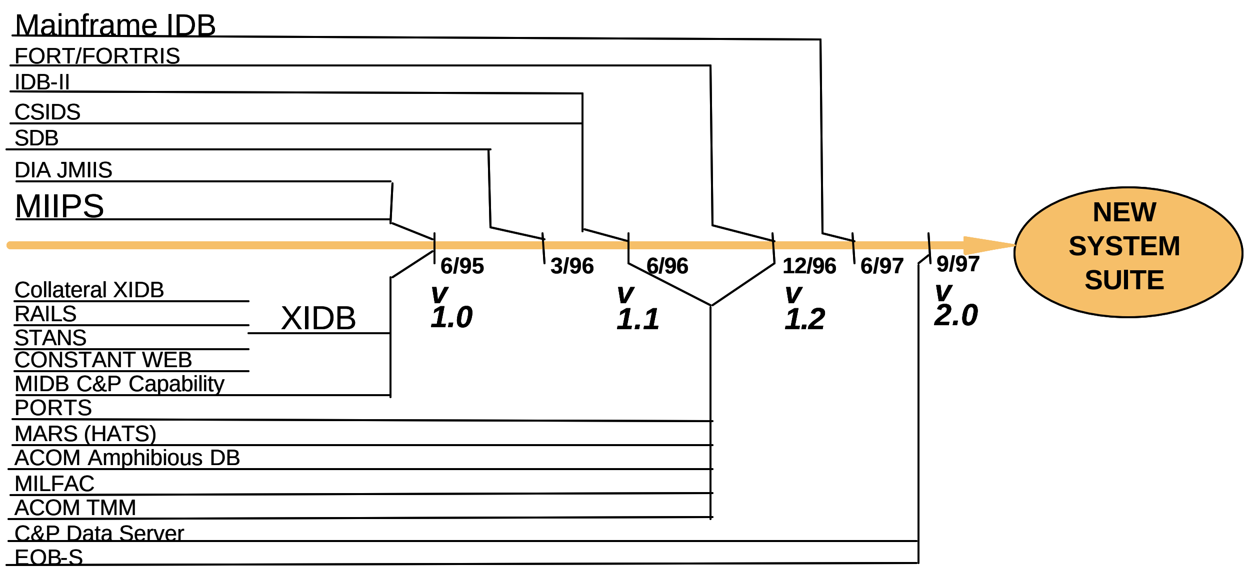


Figure 5-29. Systems Evolution Description (SV-8) – Migration

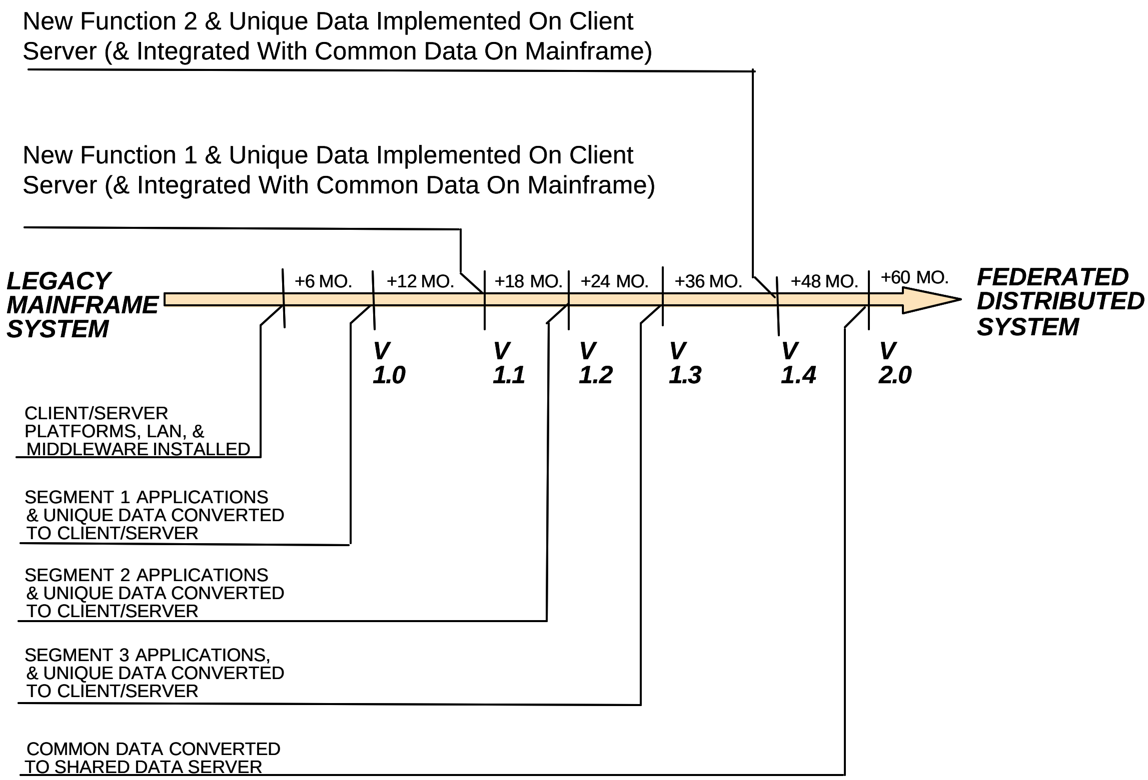


Figure 5-30. Systems Evolution Description (SV-8) – Evolution

5.8.2 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.

5.8.3 Systems Evolution Description (SV-8) – Data Element Definitions

Table 5-8 describes the architecture data elements for SV-8.

Table 5-8. Data Element Definitions for Systems Evolution Description (SV-8)

Data Elements	Attributes	Example Values/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

Data Elements	Attributes	Example Values/Explanation
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
	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*		
	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

Data Elements	Attributes	Example Values/Explanation
	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
	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

5.8.4 CADM Support for Systems Evolution Description (SV-8)

Figure 5-31 provides a high-level diagram from the CADM showing key entities that are used to store architecture data for SV-8 in a CADM-conformant database. Each SV-8 is represented as an instance of SYSTEM-EVOLUTION-DESCRIPTION, a subtype of DOCUMENT, which has attributes for Name, Time Frame, Category Code (with domain value Systems Evolution Description), etc.

SV-8 is an instance of SYSTEM-EVOLUTION-DESCRIPTION, which is defined by one or more instances of SYSTEM-MIGRATION-EVOLUTION. The latter cites a specific SYSTEM and SYSTEM-DIRECTED-CONSTRAINT, and it is in turn defined by one or more instances of SYSTEM-ASSOCIATION-MIGRATION. SYSTEM-ASSOCIATION-MIGRATION cites a specific GUIDANCE (e.g., INFORMATION-TECHNOLOGY-REQUIREMENT, TECHNICAL-CRITERION), a specific AGREEMENT (e.g., a standard or standard profile), and a Time Frame PERIOD. As a child of SYSTEM-ASSOCIATION, SYSTEM-ASSOCIATION-MIGRATION has all the characteristics of SYSTEM-ASSOCIATION (e.g., interface status, interoperability level), SYSTEM-ASSOCIATION-DIRECTED-CONSTRAINT, and SYSTEM-ASSOCIATION-MEANS. The Time Frame for a SYSTEM-ASSOCIATION-MIGRATION can be matched with the Time Frames for TECHNOLOGY-FORECAST and TECHNICAL-STANDARD-FORECAST (TV-2).

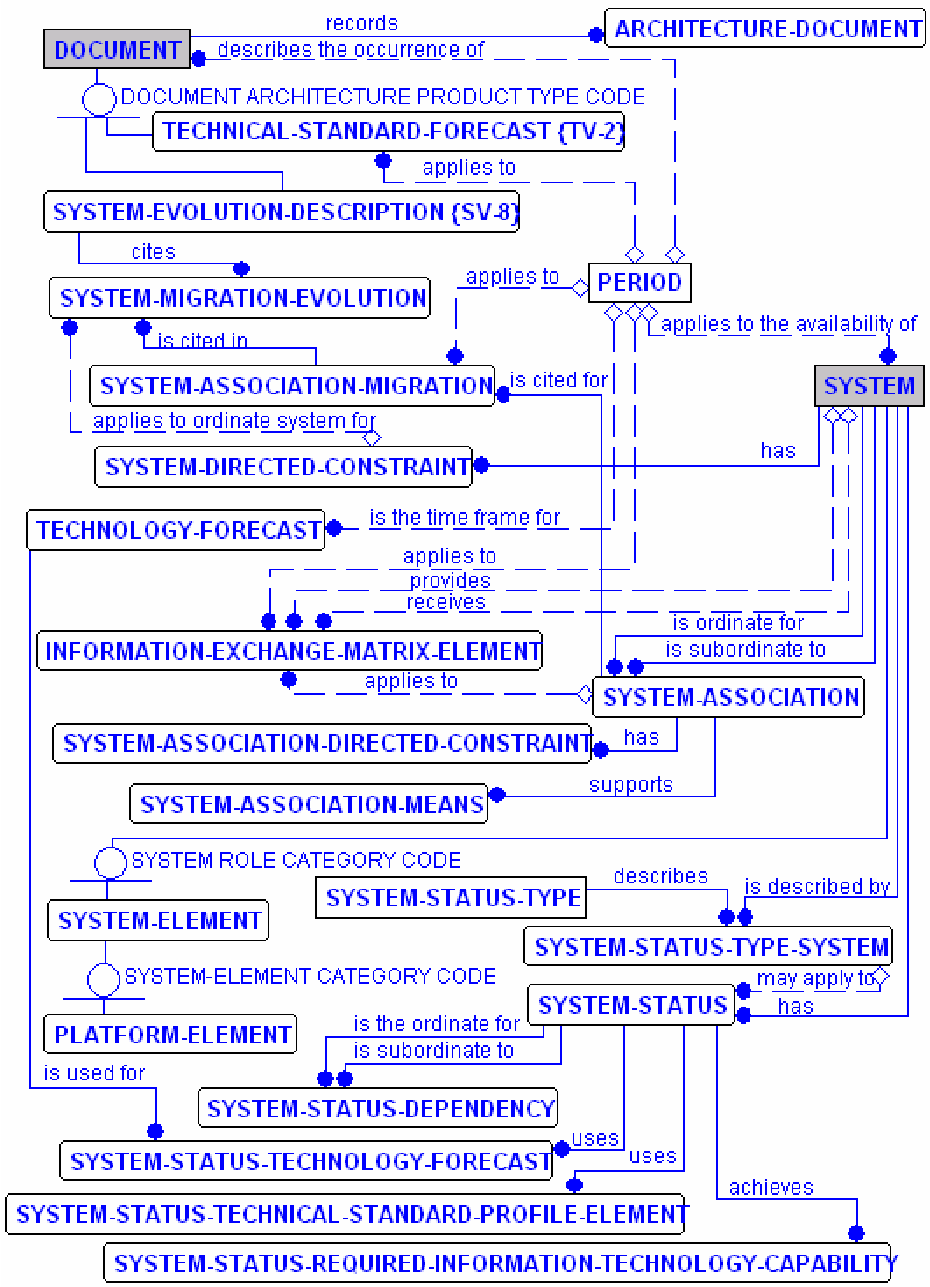


Figure 5-31. CADM ER Diagram for Systems Evolution Description (SV-8)

5.9 SYSTEMS TECHNOLOGY FORECAST (SV-9)

5.9.1 Systems Technology Forecast (SV-9) – Product Description

Product Definition. The Systems Technology Forecast defines the underlying current and expected supporting technologies. It is not expected to include predictions of technologies as with a crystal ball. 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.

Product Purpose. SV-9 provides a summary of emerging technologies that impact 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.

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). That is, 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 (i.e., target) architectures. The forecast should be focused on technology areas that are related to the purpose for which a given architecture is being described and should 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 to which the architecture is subject to using, such as the DoD Technical Reference Model (TRM) [DoD TRM, 2001] or the Joint Technical Architecture (JTA) [DISA, 2002]. Using these reference models or standards profiles, the architecture should select the service areas and services relevant to the architecture. SV-9 forecasts relate to the Technical Standards Profile (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).

A template for SV-9 is shown in **Table 5-9**. The template organization is based on JTA service categories. The template entries contain the names of example technologies for the JTA service areas and services, and summary status predictions.

Table 5-9. Systems Technology Forecast (SV-9) – Notional Example

JTA 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 (Open Source DBMS) available		
Operating System		Next MS Windows desktop 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 CRT monitors for PC desktops become price competitive	Thin screen LED monitors become price competitive for desktops Conventional CRT technology monitors for desktops become obsolete
Persistent Storage	5G PCMCIA type 2 card available		Disk storage capacity doubles 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 (e.g., systems) where applicable. The list of systems potentially impacted by the technology can be included directly in SV-9 by specifying a time period in the cell corresponding to the system element and the applicable JTA service area and service. **Table 5-10** is a template showing this variant of SV-9.

Table 5-10. Systems Technology Forecast (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
JTA Service Area	Service	Technology Forecast (Summary Prediction)	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)

5.9.2 UML Representation

There is no equivalent to this product in UML.

5.9.3 Systems Technology Forecast (SV-9) – Data Element Definitions

Table 5-11 describes the architecture data elements for SV-9.

Table 5-11. Data Element Definitions for Systems Technology Forecast (SV-9)

Data Elements	Attributes	Example Values/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
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

Data Elements	Attributes	Example Values/Explanation
	Reference Model or Standards Profile Name*	Name/identifier of the reference model or standards profile used to organize the forecast
Timed Technology Forecast Impacts System		
	Timed Technology Forecast Name	Name/identifier of a timed technology forecast
	System	Name/identifier of a system that could be impacted by the technology forecast
Timed Technology Forecast Impacts System Hardware/Software Item		
	Timed Technology Forecast Name	Name/identifier of a timed technology forecast
	System Hardware/Software Item	Name/identifier of a system hardware/software item that could be impacted by the technology forecast
Timed Technology Forecast Impacts Communications Link		
	Timed Technology Forecast Name	Name/identifier of a timed technology forecast
	Communications Link Name	Name/identifier of a communications link that could be impacted by the technology forecast
Timed Technology Forecast Impacts 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 Impacts 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 could be impacted by the technology forecast
Timed Technology Forecast Corresponds to Time Period		
	Timed Technology Forecast Name	Name/identifier of a timed technology forecast
	Time Period Identifier	Identifier of a time period
Timed Technology Forecast May Retire Standard		
	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 s tandards forecast

5.9.4 CADM Support for Systems Technology Forecast (SV-9)

Figure 5-32 provides a high-level diagram from the CADM showing key entities that are used to store architecture data for SV-9 in a CADM-conformant database. Each SV-9 is represented as an instance of SYSTEM-TECHNOLOGY-FORECAST, a subtype of DOCUMENT, which has attributes for Name, Time Frame, Category Code (with domain value Technology Forecast Matrix), and so on.

SV-9 is an instance of DOCUMENT in the subtype SYSTEM-TECHNOLOGY-FORECAST, which is defined by one or more instances of SYSTEM-TECHNOLOGY-FORECAST-PROFILE. The latter contains the identifiers for a specific SYSTEM, a specific TECHNICAL-SERVICE, a TECHNOLOGY-COUNTERMEASURE, a TECHNOLOGY-FORECAST, and a PERIOD. The PERIOD is the same entity used to characterize a TECHNICAL-STANDARD-FORECAST-ELEMENT, TECHNICAL-STANDARD-FORECAST, and TECHNOLOGY-FORECAST.

The CADM provides for specification of TECHNOLOGY, together with associations such as TECHNOLOGY-ASSOCIATION (between two instances of TECHNOLOGY), TECHNOLOGY-COUNTERMEASURE, TECHNICAL-CRITERION, TECHNOLOGY-FORECAST, and TECHNOLOGY-ISSUE. Each TECHNOLOGY-FORECAST-MATRIX (a subtype of DOCUMENT) is specified by citing instances of TECHNOLOGY-FORECAST-PROFILE. In addition, a TECHNICAL-CRITERIA-DOCUMENT (also a subtype of DOCUMENT) can be specified by citing instances of TECHNICAL-CRITERION-PROFILE, which in turn cites instances of TECHNICAL-CRITERION and TECHNICAL-GUIDELINE (a subtype of GUIDANCE).

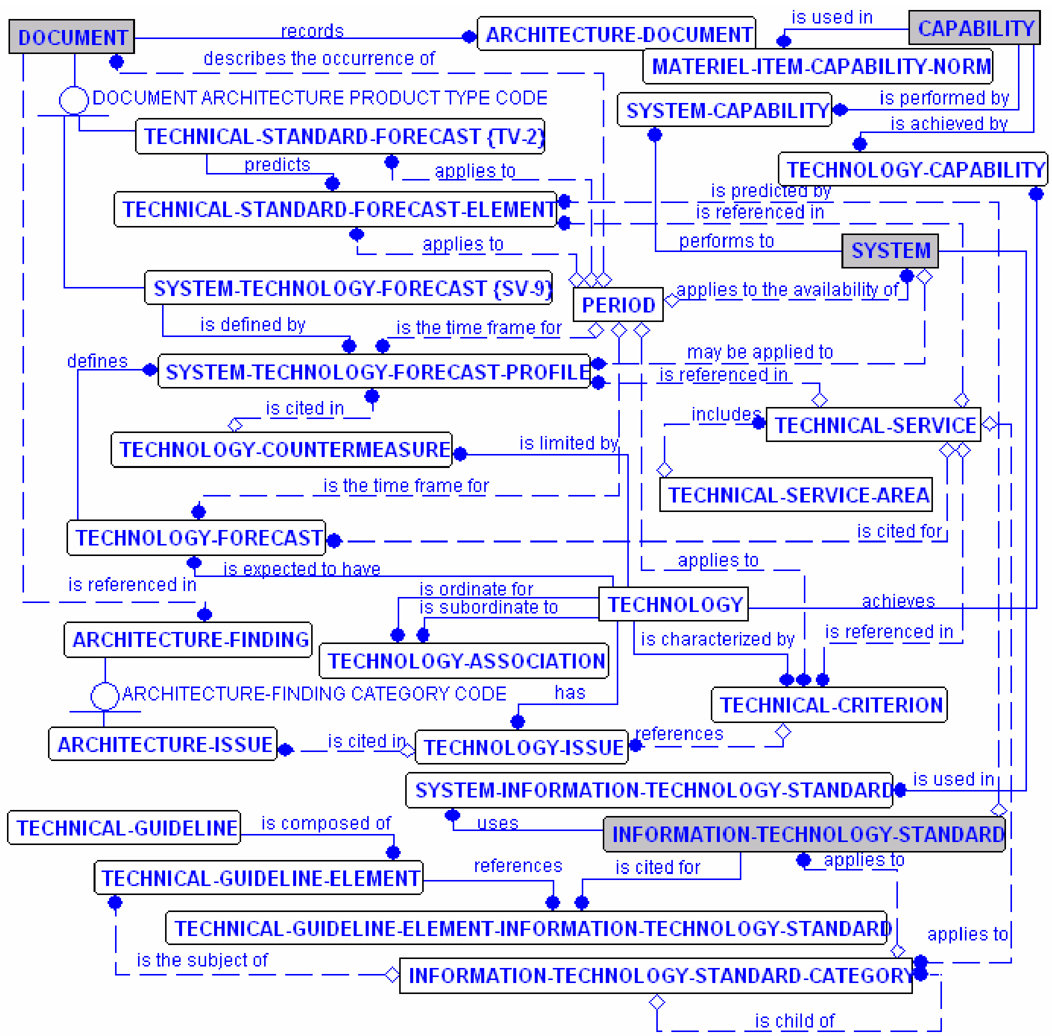


Figure 5-32. CADM ER Diagram for Systems Technology Forecast (SV-9)

5.10 SYSTEMS FUNCTIONALITY SEQUENCE AND TIMING DESCRIPTIONS (SV-10A, 10B, AND 10C)

5.10.1 Overview of Systems Functionality Sequence and Timing Descriptions

Models for SV-10. 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). Behavior modeling and documentation is key to a successful architecture description, because it is how the architecture behaves that 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.

Three types of models may be used to adequately 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)

SV-10b and SV-10c may be used separately or together, as necessary, to describe critical timing and sequencing behavior in the SV. Both types of diagrams are used by a wide variety of different systems methodologies.

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 as described in SV-10a.

5.10.2 CADM Support for Systems Functionality Sequences and Threads

Figure 5-33 provides a high-level diagram from the CADM showing key entities that are used to store architecture data for sequences and threads, which underlies each of SV-10a, SV-10b, and SV-10c. The CADM uses entities such as ACTION and ACTION-ASSOCIATION to support all the temporal and functional relationships among pairs of (operational and other) ACTION instances as shown in the figure. The temporal attributes permit arbitrary sequencing of ACTION instances, as well as specifying timing offsets needed for planning concepts (e.g., a fire plan) such as relative timing from an H-Hour or a D-Day. Threads of activities are specified using OPERATIONAL-MISSION-THREAD (for sequences of information exchange requirements [IERs]), PROCESS-ACTIVITY-THREAD (for sequences of process activities), and ACTIVITY-MODEL-THREAD (for sequences of process activities embedded in a single activity model). These and related entities are also shown in Figure 5-33.

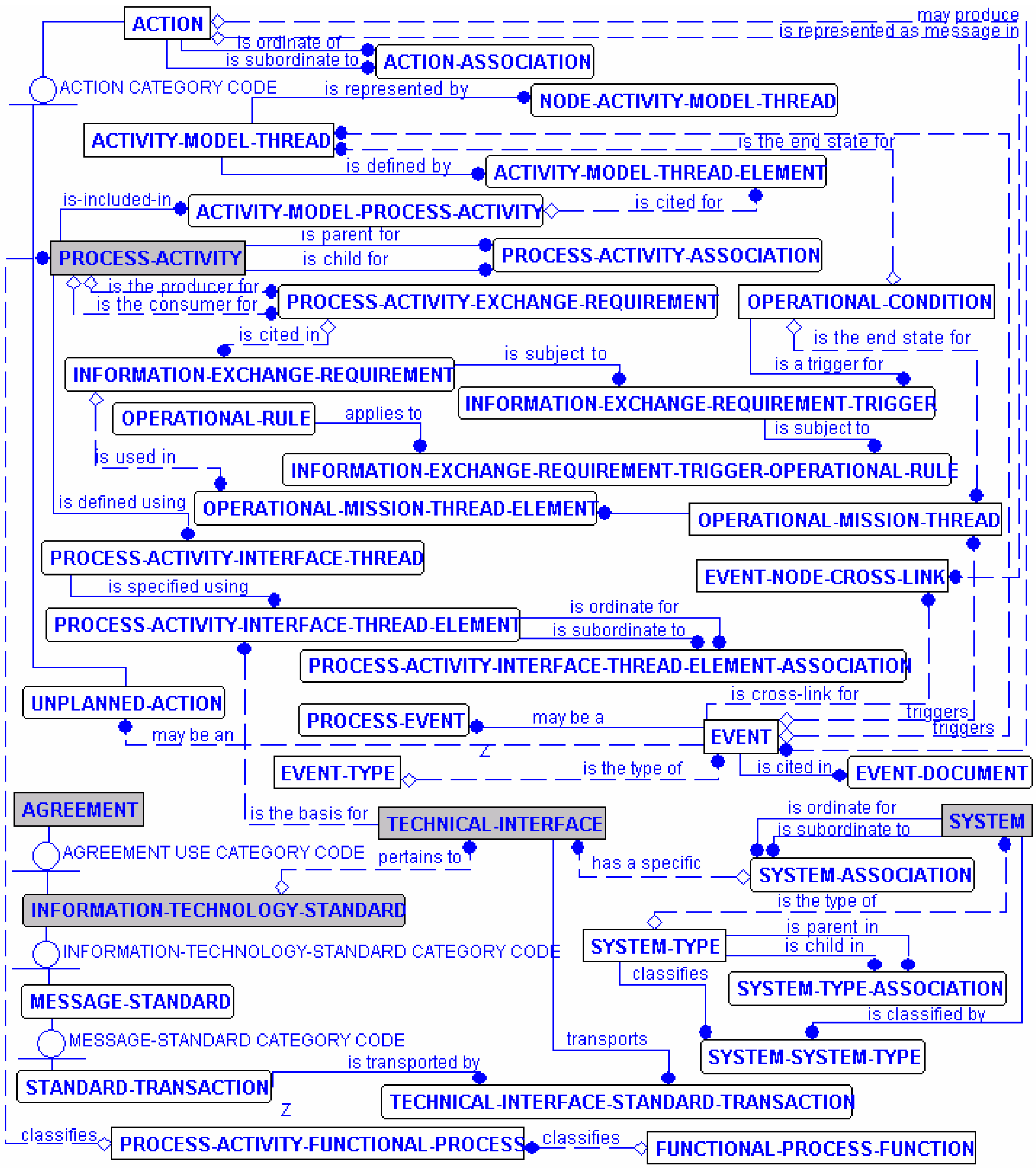


Figure 5-33. CADM ER Diagram for Systems Functionality Sequence and Threads

Three entities—OPERATIONAL-MISSION-THREAD-CAPABILITY, PROCESS-ACTIVITY-CAPABILITY, and REQUIRED-INFORMATION-TECHNOLOGY-CAPABILITY (directly related in to INFORMATION-TECHNOLOGY-REQUIREMENT, the parent of INFORMATION-EXCHANGE-REQUIREMENT and PROCESS-ACTIVITY-EXCHANGE-REQUIREMENT)—provide the ability to express specific capabilities for threads of IERs and sequences of operational activities. These entities are depicted in the diagram for the OV-6 products and are not repeated in Figure 5-33.

5.10.3 Systems Rules Model (SV-10a) – Product Description

Product Definition. Systems rules are constraints on an architecture, on a system(s), or system hardware/software item(s), and/or on a system function(s). While other SV products (e.g., SV-1, SV-2, SV-4, SV-11) describe the static structure of the Systems View (i.e., what the systems can do), they do not describe, for the most part, what the systems *must* do, or what it *cannot* do.

At the systems or system hardware/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. Such rules can be expressed in a textual form, for example, "If (these conditions) exist, and (this event) occurs, then (perform these actions)."

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

Product Detailed Description. *Rules are statements that define or constrain some aspect of the enterprise.* In contrast to the Operational Rules Model (OV-6a), SV-10a focuses on constraints imposed by some aspect of operational performance requirements that translate into system performance requirements. At a lower level of detail, it focuses on some aspects of systems design or implementation. Thus, as the operational rules can be associated with the Operational Activity Model (OV-5), the systems rules in SV-10a can be associated with SV-1 and SV-2 systems and hardware/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 the Logical Data Model (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 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 used 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 5-34 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 related system data synchronized. The example is given in a form of structured English.

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

Figure 5-34. Systems Rules Model (SV-10a) – Action Assertion Example

5.10.4 UML Representation

There is no equivalent diagram in UML. However, if one considers systems rules to be equivalent to complex, nested If-Then-Else and CASE statements, then these statements can be unambiguously derived from UML statechart diagrams for the object classes defined as systems, system functions or system data. Pre- and post-conditions can be specified 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.

5.10.5 Systems Rules Model (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 5-12** describes the architecture data elements for SV-10a.

Table 5-12. Data Element Definitions for Systems Rules Model (SV-10a)

Data Elements	Attributes	Example Values/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

5.10.6 CADM Support for Systems Rule Model (SV-10a)

Figure 5-35 provides a high-level diagram from the CADM showing key entities that are used to store architecture data for SV-10a in a CADM-conformant database. Each SV-10a is represented as an instance of RULE-MODEL, a subtype of DOCUMENT, which has attributes for Name, Time Frame, Category Code (with domain value Rule Model), etc.

The CADM specification for SV-10a also allows the user to describe structural assertions, action assertions with their respective conditions, as well as the applicable integrity constraints, authorizations, and derivations via the instantiation of GUIDANCE Text for OPERATIONAL-RULE Category Code = *Structural Assertion*, *Action Assertion*, *Integrity Rule*, *Authorization*, or *Derivation*. Individual rules can be expressed explicitly, where appropriate, as relationships in a CONCEPTUAL-DATA-MODEL, SYSTEM-FUNCTIONALITY-DESCRIPTION, or ACTIVITY-MODEL. They can also be stated in formal or informal terms, as appropriate, as instances of OPERATIONAL-RULE, TECHNICAL-GUIDELINE, INFORMATION-TECHNOLOGY-REQUIREMENT, and other instances of GUIDANCE. Three subtypes of OPERATION-RULE are defined for the CADM to support specific SV-10a architecture data requirements from the Framework: ACTION-ASSERTION-RULE, STRUCTURAL-ASSERTION-RULE, and DATABASE-RULE. These rules are related to each other through GUIDANCE-ASSOCIATION and to various architecture products through GUIDANCE-DOCUMENT. The specific set of OPERATIONAL-RULE instances for a RULE-MODEL is specified as instances of the associative entity RULE-MODEL-OPERATIONAL-RULE.

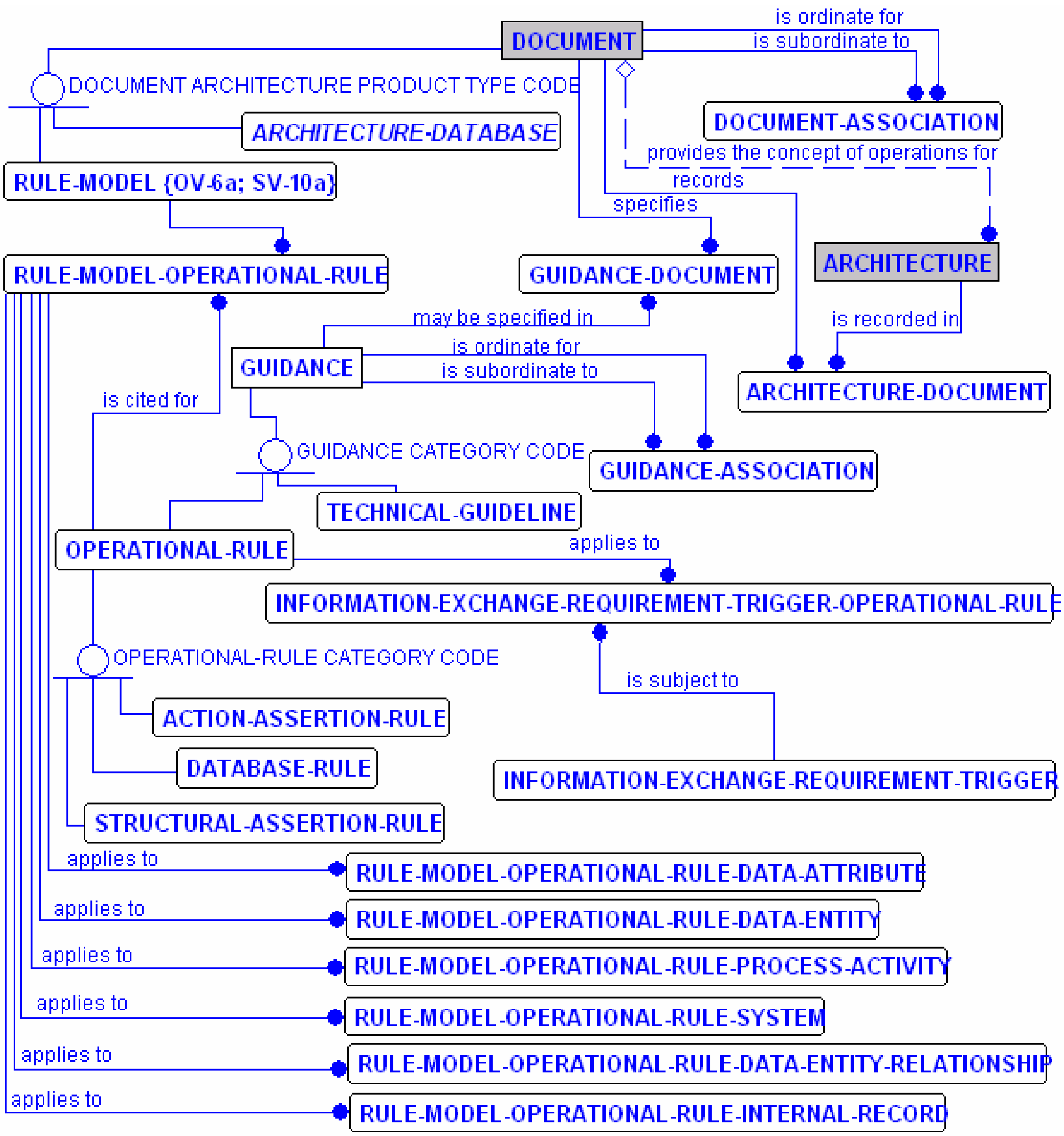


Figure 5-35. CADM ER Diagram for Systems Rules Model (SV-10a)

5.10.7 Systems State Transition Description (SV-10b) – Product Description

Product Definition. The Systems State Transition Description 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.

Product Purpose. The explicit time sequencing of system functions in response to external and internal events is not fully expressed in SV-4. SV-10b can be used to describe the explicit sequencing of the system functions. Alternatively, SV-10b can be used 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.

Product Detailed Description. SV-10b is based on the statechart diagram [OMG, 2003]. A state machine is defined as "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 action(s) 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.

SV-10b can be used 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. SV-10b can be used 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 5-36 provides a template for a simple SV-10b. The black dot and incoming arrow point to initial states (usually one per diagram), while terminal states are identified by an outgoing arrow pointing to a black dot with a circle around it. States are indicated by rounded corner box icons and labeled by name or number and, optionally, any actions associated with that state. Transitions between states are indicated by 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.

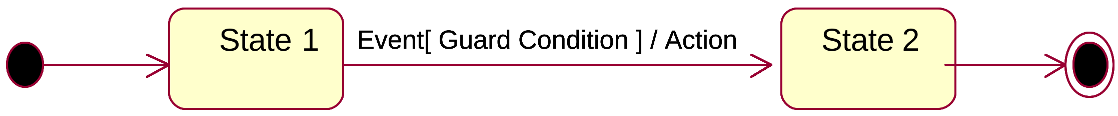


Figure 5-36. Systems State Transition Description (SV-10b) – High-Level Template

5.10.8 UML Representation

SV-10b may be produced 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 5-36). 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.

5.10.9 Systems State Transition Description (SV-10b) – Data Element Definitions

SV-10b describes the detailed sequencing of functions in a system by depicting how the current state of the system changes in response to external and internal events, resulting in time-sequenced activities. **Table 5-13** describes the architecture data elements for SV-10b.

Table 5-13. Data Element Definitions for Systems State Transition Description (SV-10b)

Data Elements	Attributes	Example Values/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
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)

5.10.10 CADM Support for Systems State Transition Description (SV-10b)

Figure 5-37 provides a high-level diagram from the CADM showing key entities that are used to store architecture data for SV-10b in a CADM-conformant database. Each SV-10b is represented as an instance of STATE-TRANSITION-DESCRIPTION, a subtype of DOCUMENT, which has attributes for Name, Time Frame, Category Code (with domain value State Transition Description), etc.

The CADM specification for SV-10b also allows the user to describe all the relevant components of a state transition diagram by the proper instantiation of PROCESS-STATE, PROCESS-STATE-ACTION, TRANSITION-PROCESS, TRANSITION-PROCESS-RESULTING-ACTION, PROCESS-PSEUDO-STATE, STATE-TRANSITION-DESCRIPTION, PROCESS-STATE-VERTEX, COMPOSITE-PROCESS STATE and NESTING-PROCESS-STATE.

SV-10b is an instance of DOCUMENT in the subtype STATE-TRANSITION-DESCRIPTION. To avoid confusion with U.S. State, the term "PROCESS-STATE" has been used (parallel to PROCESS-ACTIVITY) instead of STATE. The key entities in the CADM are independent entities TRANSITION-PROCESS, PROCESS-STATE-VERTEX, ACTION, and EVENT. Instances of TRANSITION-PROCESS are defined in terms of pairs of instances of PROCESS-STATE-VERTEX, each of which may be a PROCESS-PSEUDO-STATE (representing forks, joins, branches, etc.) or PROCESS-STATE. Key associations include TRANSITION-PROCESS-RESULTING-ACTION and PROCESS-STATE-ACTION. The CADM provides explicit specification of four types of events called out in UML: SIGNAL-EVENT, CALL-EVENT, TIME-EVENT, and CHANGE-EVENT. The CADM also explicitly permits a STATE-TRANSITION-DESCRIPTION (SV-10b) to represent a specific SYSTEM (including a specific component), SOFTWARE-ITEM, or ACTION (which may be a TASK or a PROCESS-ACTIVITY).

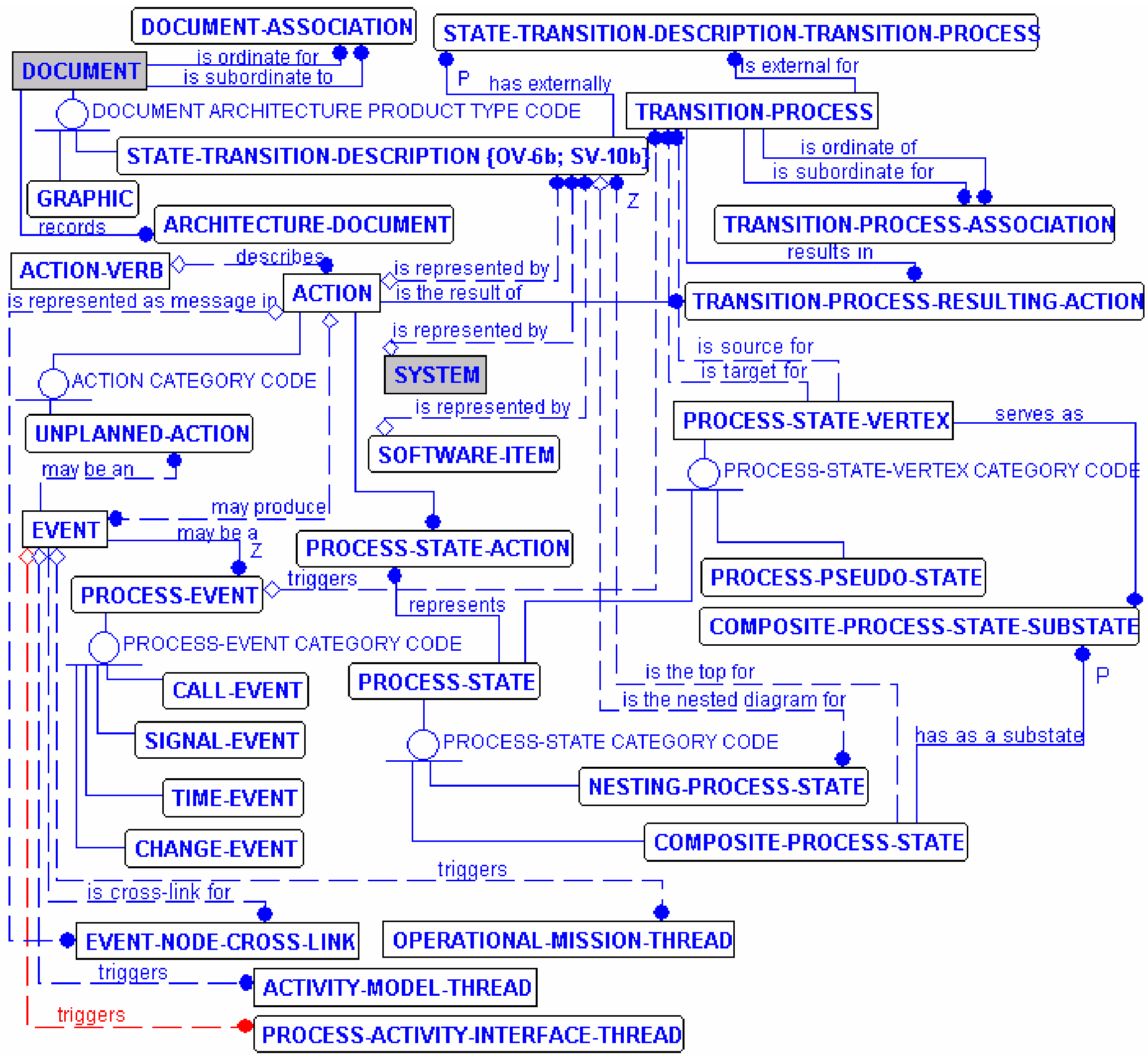


Figure 5-37. CADM ER Diagram for Systems State Transition Description (SV-10b)

5.10.11 Systems Event-Trace Description (SV-10c) – Product Description

Product Definition. The Systems Event-Trace Description provides a time-ordered examination of the system data elements exchanged between participating systems (external and internal), 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.

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, in order to perform its assigned functionality.

Product Detailed Description. SV-10c allows the tracing of actions in a scenario or critical sequence of events. With time proceeding from the top of the diagram to the bottom, a specific diagram lays out the sequence of system data exchanges that occur between systems (external and internal), system functions, or human role for a given scenario. Different scenarios should be depicted by separate diagrams. SV-10c can be used by itself or in conjunction with a SV-10b to describe dynamic behavior of system processes or system function threads.

Figure 5-38 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 event(s) 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/role becomes aware of the events. The direction of the events represents the flow of control from one system/function/role to another based on the event. Each diagram may represent systems (external and internal) or system functions, but not both in the same diagram. Human roles may also be used in the diagram along with either systems or system functions in order to describe the humans' interfaces to the systems or system functions.

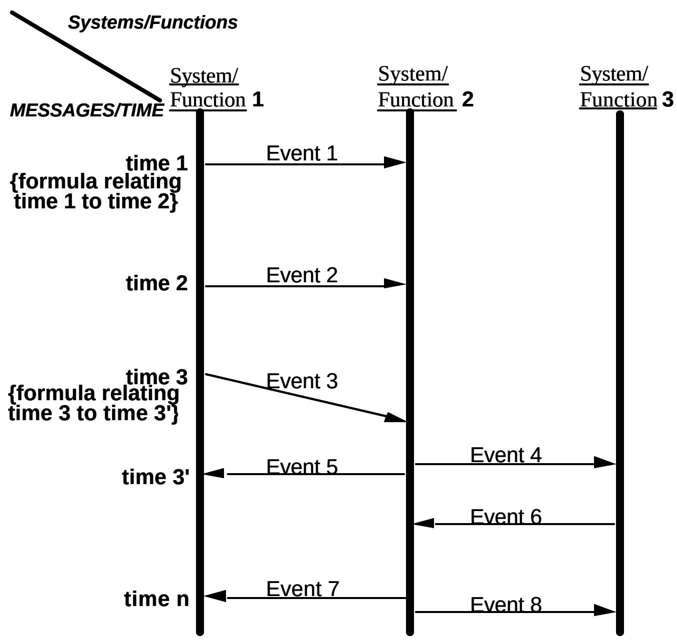


Figure 5-38. Systems Event-Trace Description (SV-10c) – Template

5.10.12 UML Representation

UML sequence diagrams may be used to model SV-10c. Each diagram may represent systems (external and internal) or system functions, but not both in the same diagram.

Figure 5-38 provides a template for a UML sequence diagram.

5.10.13 Systems Event-Trace Description (SV-10c) – Data Element Definitions

Table 5-14 describes the architecture data elements for SV-10c.

Table 5-14. Data Element Definitions for Systems Event-Trace Description (SV-10c)

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

5.10.14 CADM Support for Systems Event-Trace Description (SV-10c)

Figure 5-39 provides a high-level diagram from the CADM showing key entities that are used to store architecture data for SV-10c in a CADM-conformant database. Each SV-10c is represented as an instance of EVENT-TRACE-DESCRIPTION, a subtype of DOCUMENT, which has attributes for Name, Time Frame, Category Code (with domain value Event-Trace Description), etc.

Each NODE is considered to have a (relative) timeline (or lifeline) in the CADM. Each such NODE can be explicitly associated with a SYSTEM or SYSTEM-FUNCTION (a subtype of PROCESS-ACTIVITY).[21] Each cross-link connecting two NODE instances at different times is represented (independently) as an instance of EVENT-NODE-CROSS-LINK (independence enables cross-links to be reused from one scenario to another). Cross-links can start at explicit or implicit (computed) absolute or relative times, the latter in terms of the start or end of another cross-link (using a method analogous to one defined for temporally related ACTION instances). The set of cross-links for a specific SV-10c is specified as a set of instances of EVENT-TRACE-SCENARIO-ELEMENT. Each EVENT-TRACE-DESCRIPTION can be associated directly to an OPERATIONAL-SCENARIO.

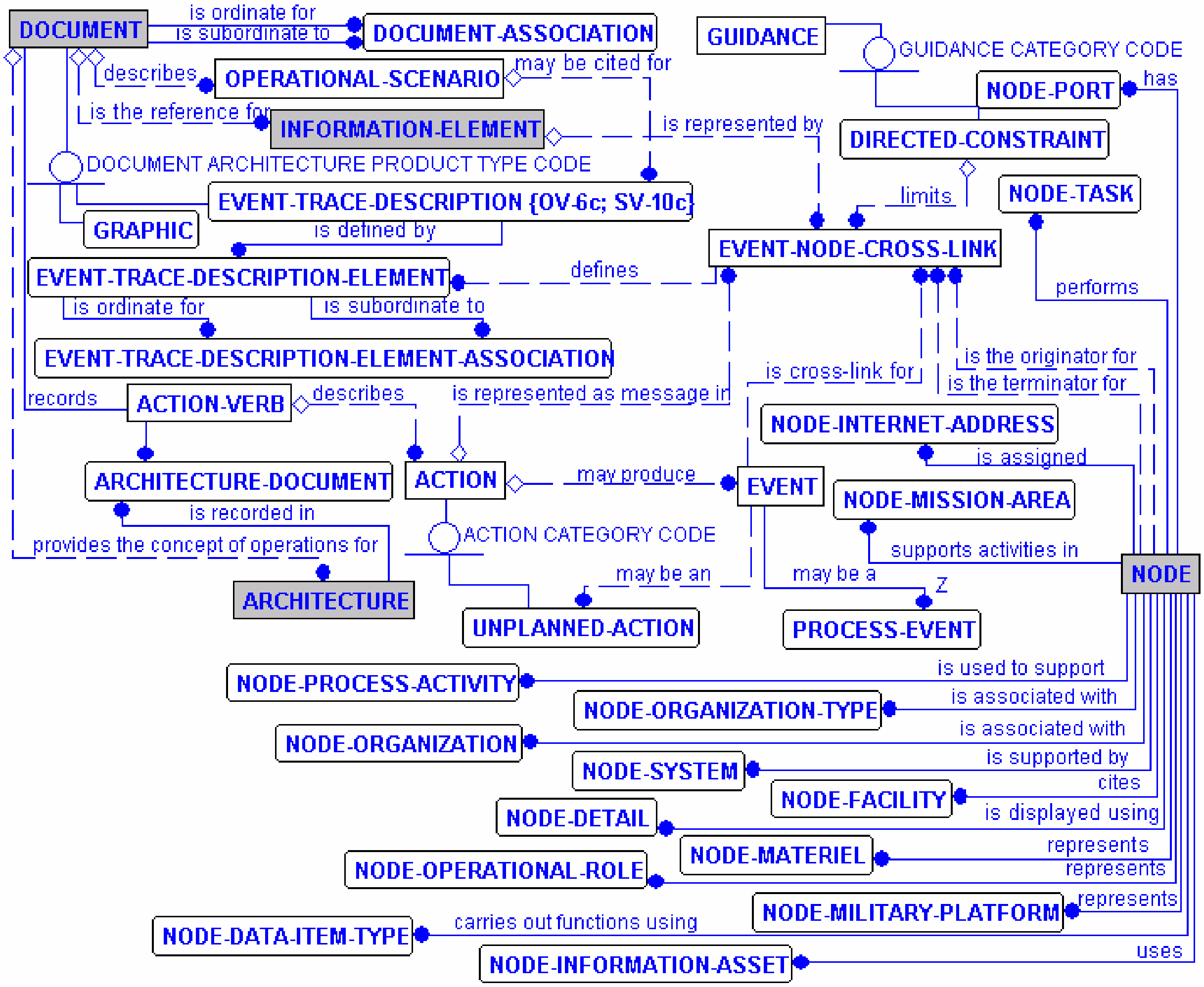


Figure 5-39. CADM ER Diagram for Systems Event-Trace Description (SV-10c)

---

[21] Use of NODE in Systems Event-Trace Description (SV-10c) permits the same data structures to represent many kinds of event-traces, not only for systems and system functions but also for a specific MATERIEL, MILITARY-PLATFORM, or FACILITY, among others. For this reason, many associations with NODE are shown at the bottom of the figure.

5.11 PHYSICAL SCHEMA (SV-11)

5.11.1 Physical Schema (SV-11) – Product Description

Product Definition. The Physical Schema product is one of the architecture products closest to actual system design in the Framework. The product defines the structure of the various kinds of system data that are utilized by the systems in the architecture.

Product Purpose. The product serves several purposes, including (a) providing as much detail as possible on the system data elements exchanged between systems, thus reducing the risk of interoperability errors, and (b) providing system data structures for use in the system design process, if necessary.

Product Detailed Description. SV-11 is an implementation-oriented data model that is used in the Systems View to describe how the information requirements represented in Logical Data Model (OV-7) are actually implemented. Entities represent (a) system data flows in SV-4, (b) system data elements specified in SV-6, (c) triggering events in SV-10b, and/or (d) events in SV-10c.

There should be a mapping from a given OV-7 to SV-11 if both models are used. The form of SV-11 can vary greatly, as shown in **Figure 5-40**. 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 the mode used to exchange information. A Data Definition Language (DDL) (e.g., Structured Query Language [SQL]) may also be used in the cases where shared databases are used to integrate systems. Interoperating systems may use a variety of techniques to exchange system data and have several distinct partitions in their SV-11 with each partition using a different form. Standards associated with entities (e.g., entities are those defined in DoD XML Registry) are also documented in this product.

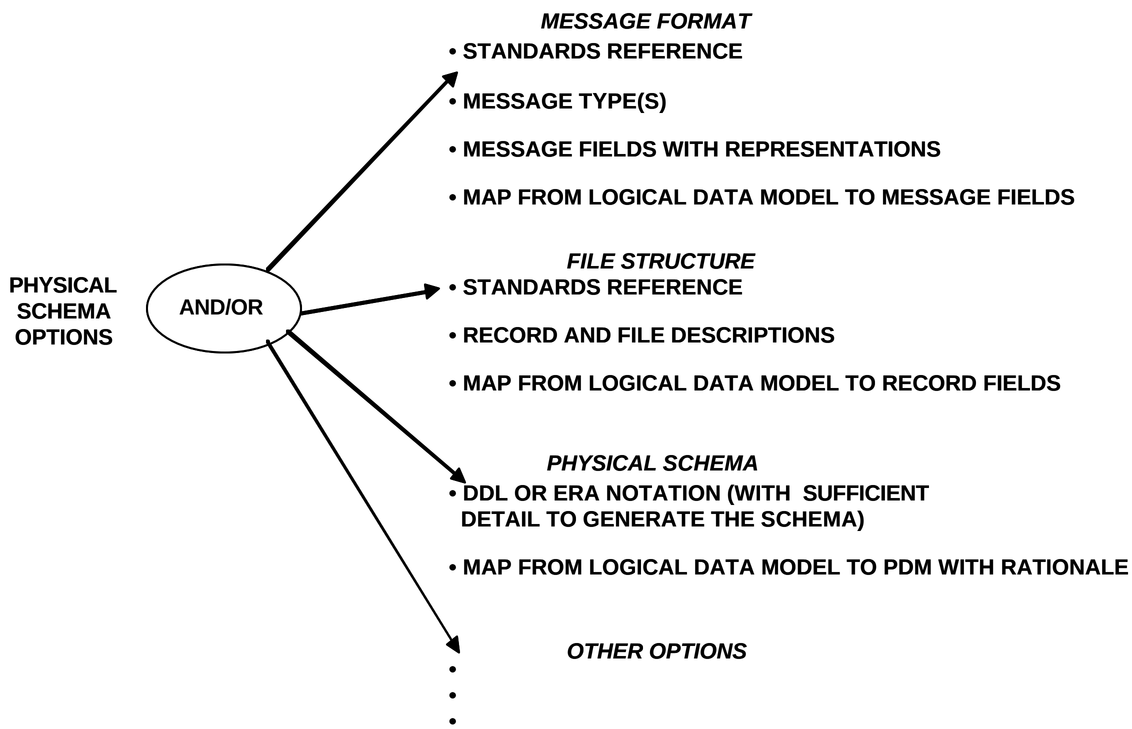


Figure 5-40. Physical Schema (SV-11) – Representation Options

5.11.2 UML Representation

SV-11 may be specified 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, collaborations, 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 5-41** is a template of such a use of a UML class diagram.

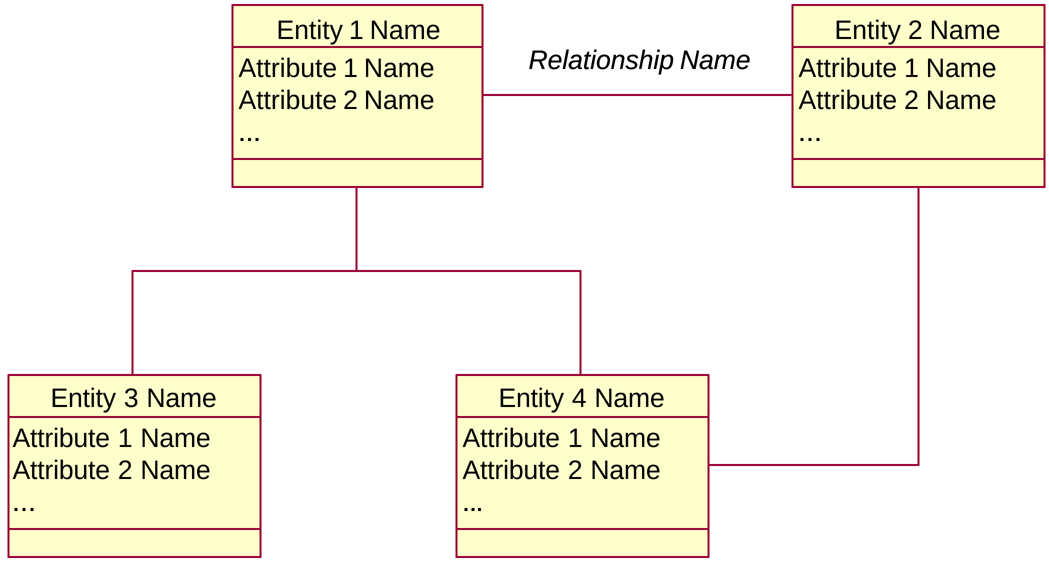


Figure 5-41. UML Class Diagram for Physical Schema (SV-11)

Object classes defined here should trace to (a) system data flows in SV-4, (b) system data elements specified in SV-6, (c) transition triggers in SV-10b, and/or (d) 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.

5.11.3 Physical Schema (SV-11) – Data Element Definitions

Table 5-15 describes the architecture data elements for SV-11.

Table 5-15. Data Element Definitions for Physical Schema (SV-11)

Data Elements	Attributes	Example Values/Explanation
Non-graphical Types		
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 (e.g., USMTF; TADIL A, B, J)
	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 complete ly 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 structure(s) 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 (e.g., IDEF1X); may be tool dependent (e.g., DIAD, 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
	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

Data Elements	Attributes	Example Values/Explanation
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**

5.11.4 CADM Support for Physical Schema (SV-11)

Figure 5-42 provides a high-level diagram from the CADM showing key entities that are used to store architecture data for SV-11 in a CADM-conformant database. Each SV-11 is represented as an instance of PHYSICAL-SCHEMA-SPECIFICATION, a subtype of DOCUMENT, which has attributes for Name, Time Frame, Category Code (with domain value Physical Schema Specification), etc. The same view is used for internal forms of data as actually stored or external forms of data presented to users.

SV-11 is an instance of PHYSICAL-SCHEMA-SPECIFICATION, which cites a specific instance of INTERNAL-DATA-MODEL and USER-PRESENTATION-VIEW (both subtypes of INFORMATION-ASSET). The key entities in this specification include INTERNAL-RECORD, CONCEPTUAL-DATA-MODEL, DATA-ENTITY, DATA-ATTRIBUTE, DATA-DOMAIN, and SYSTEM-PLATFORM (all subtypes of INFORMATION-ASSET), together with other approved entities such as DATA-ENTITY-RELATIONSHIP, USER-PRESENTATION-VIEW-SYSTEM-PLATFORM, and INFORMATION-ASSET-RELATION. Each DATA-ENTITY can be directly related to an ACTIVITY-MODEL-INFORMATION-ELEMENT-ROLE (providing a link between data models and activity models).

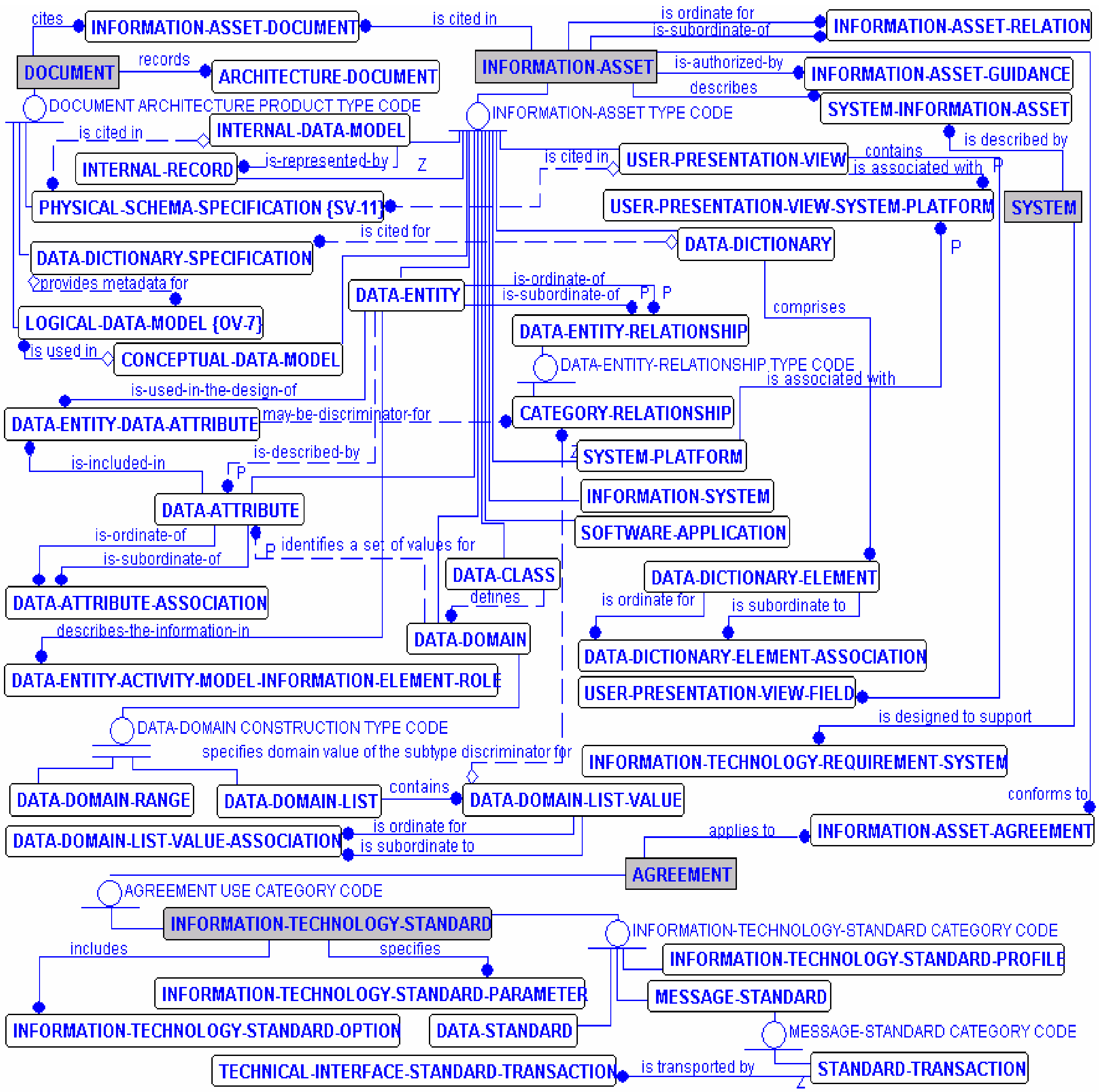


Figure 5-42. CADM ER Diagram for Physical Schema (SV-11)

6 TECHNICAL STANDARDS VIEW PRODUCTS

The Technical Standards View (TV) provides 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)

6.1 TECHNICAL STANDARDS PROFILE (TV-1)

6.1.1 Technical Standards Profile (TV-1) – Product Description

Product Definition. The Technical Standards Profile 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.

Product Purpose. Primarily, this product is concerned with delineating systems standards rules and conventions that apply to architecture implementations. When the standards profile is tied to the system elements to which they apply, TV-1 serves as the bridge between the SV and TV.

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 (i.e., the profile delineates which standards may be used to implement the systems, system hardware/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 TRM [OMB, 2003], DoD TRM, or the Joint Technical Architecture (JTA) [DISA, 2002]. Using these reference models or standards profiles, the architect should select the service areas relevant to the architecture. The identification of relevant services within service areas subsequently points to agreed-upon standards. The source document used for identifying each standard must also be cited.

In most cases, especially in describing architectures with less than a Military Service-wide scope, TV-1 consists of identifying the applicable portions of the JTA and other existing technical standards guidance documents, tailoring those portions, as needed, in accordance with the latitude allowed in these guidance documents, and filling in any gaps. This process of tailoring standards guidance from higher level, more general guidance, is called creating a standards profile. For example, a DoD mission area might create a common mission-area standards profile using TV-1. Each program or project in that mission area would further refine this common profile to create its own standards profile. Care should be taken in the refinement process to ensure that systems compliant with the child profile would continue to be interoperable with systems compliant with the parent profile. If service-level JTA-like documents are used, then the relationship between the JTA and those documents must be stated. Some of the existing technical standards guidance documents are described in the Deskbook [see section on URRs in the Deskbook].

For a given domain, TV-1 may also state a common standard implementation for a particular standard, not just list the standard. Many standards can be implemented in compliant, but not interoperable ways, such as MIL STD 6016.

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, each standard listed in the profile should be associated 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 (e.g., DoD XML Registry), or the data modeling standard used (e.g., IDEF1X).

A template for TV-1 is shown in **Table 6-1**. The template contains a subset of JTA services by way of example.

Table 6-1. Technical Standards Profile (TV-1) Template

JTA Service Area	Service	JTA 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 Interface	
	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 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 in the following ways. 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 6-1** illustrates the bridge concept.

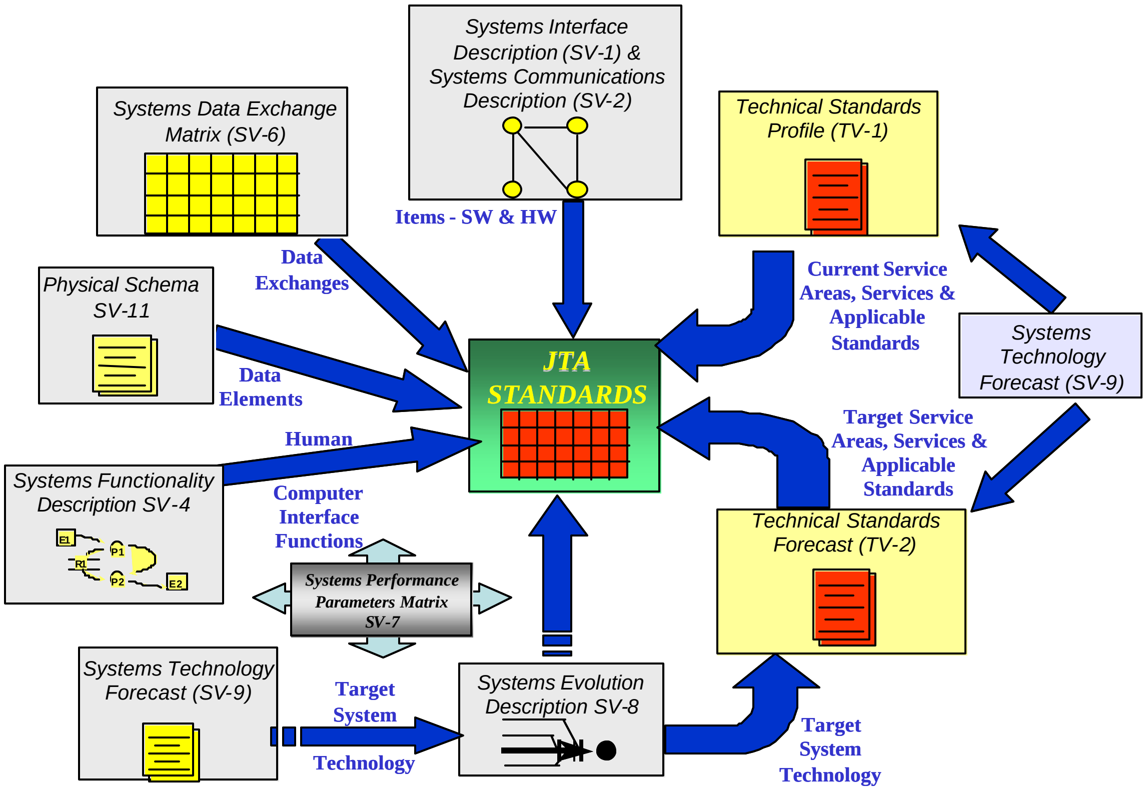


Figure 6-1. Systems Products Associated with Standards

Once the standards profile has been developed, another TV-1 product may be organized 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 6-2**.

Table 6-2. 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
JTA Service Area	**Service**	**JTA 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 shown in **Table 6-3**, 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 6-3. TV-1 Template for Systems with Corresponding Time Periods

Standards Applicable to SV-1 Systems			System A	System B	System C
JTA Service	**Service Area**	**Standard**			
Information-Technology Standards	Operating Environment	The DII COE as mandated by the JTA	Current Baseline: Jan. 12, 2003		
	Operating System Standard	ISO/IEC 9945-1:1996, Information Technology - Portable Operating System Interface (POSIX) - Part 1: System Application Program Interface (API) [C language] (Mandated Services). (JTA v2.0/3.1/4.0) http://webstore.ansi.org/ansidocstore/default.asp		Current Baseline: Jan. 12, 2003	Current Baseline: Jan. 12, 2003
	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). (JTA v2.0/3.1/4.0) http://webstore.ansi.org/ansidocstore/default.asp	6 months from Baseline	6 months from Baseline	
	Operating System Standard	IEEE 1003.2d:1994, POSIX - Part 2: Shell and Utilities - Amendment: Batch Environment. (JTA v2.0/3.1/4.0) http://standards.ieee.org/catalog/olis/search.html	12 months from Baseline	12 months from Baseline	
	Operating System Standard	Win32 APIs, Window Management and Graphics Device Interface, Volume 1 Microsoft Win32 Programmers Reference Manual, 1993 or later, Microsoft Press. (JTA v2.0/3.1) http://msdn.microsoft.com/default.asp			6 months from Baseline
	Operating System Standard	Win32 APIs, as specified in the Microsoft Platform Software Development Kit (SDK). (JTA v4.0) http://msdn.microsoft.com/library/default.asp			12 months from Baseline

6.1.2 UML Representation

There is no equivalent to this product in UML.

6.1.3 Technical Standards Profile (TV-1) – Data Element Definitions

Table 6-4 describes the architecture data elements for TV-1.

Table 6-4. Data Element Definitions for Technical Standards Profile (TV-1)

Data Elements	Attributes	Example Values/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 addressed by the reference model
	Source	Reference to the source documentation and organization supporting the reference model
Service Area*		
	Name*	Name/identifier for service area included in profile or forecast
	Description*	Textual description of service area and included services, including issues for and impacts on system architecture
	Version/Date*	Date or version number for the service area forecast (for use in forecast products)
Service*		
	Name*	Name/identifier for service
	Description*	Text summary description of the service
	Status	Applicability of some standard for this service: for example, now or future, meaning there are current standards for this service or interface to the service; or there are expected to be some in the future
Standard*		
	Standard Name*	Name and identifier for standard, including maintaining organization and relevant revision dates
	Type*	Description of the type of standard (e.g., de jure, 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		
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 OV-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)

Data Elements	Attributes	Example Values/Explanation
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
Service Area Includes Service*		
	Service Area Name*	Name/identifier of a service area
	Service Name*	Name/identifier of a service included in that service area
Standards Profile Includes Service Area*		
	Standards Profile Name*	Name/identifier of a standards profile
	Service Area Name*	Name/identifier of a service area contained in the standards profile
Standard Addresses Service*		
	Standard Name*	Name/identifier of a standard
	Service Name*	Name of the service to which the standard is applicable
Standards Profile Contains Standard*		
	Standards Profile Name*	Name/identifier of a standards profile
	Standard Name*	Name/identifier of a standard contained in the 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
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)

Data Elements	Attributes	Example Values/Explanation
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
	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

6.1.4 CADM Support for the Technical Standards View (TV)

As its name suggests, TV focuses on the state, use, and projected scope of information technology standards. **Figure 6-2** identifies the key CADM entities for specifying characteristics for information technology standards. These include subtypes of INFORMATION-ASSET used to specify domains for data standards. Note that standard is related to the system entities (system, system hardware and software item, communications systems, communications link, and communications network, etc.) and time period.

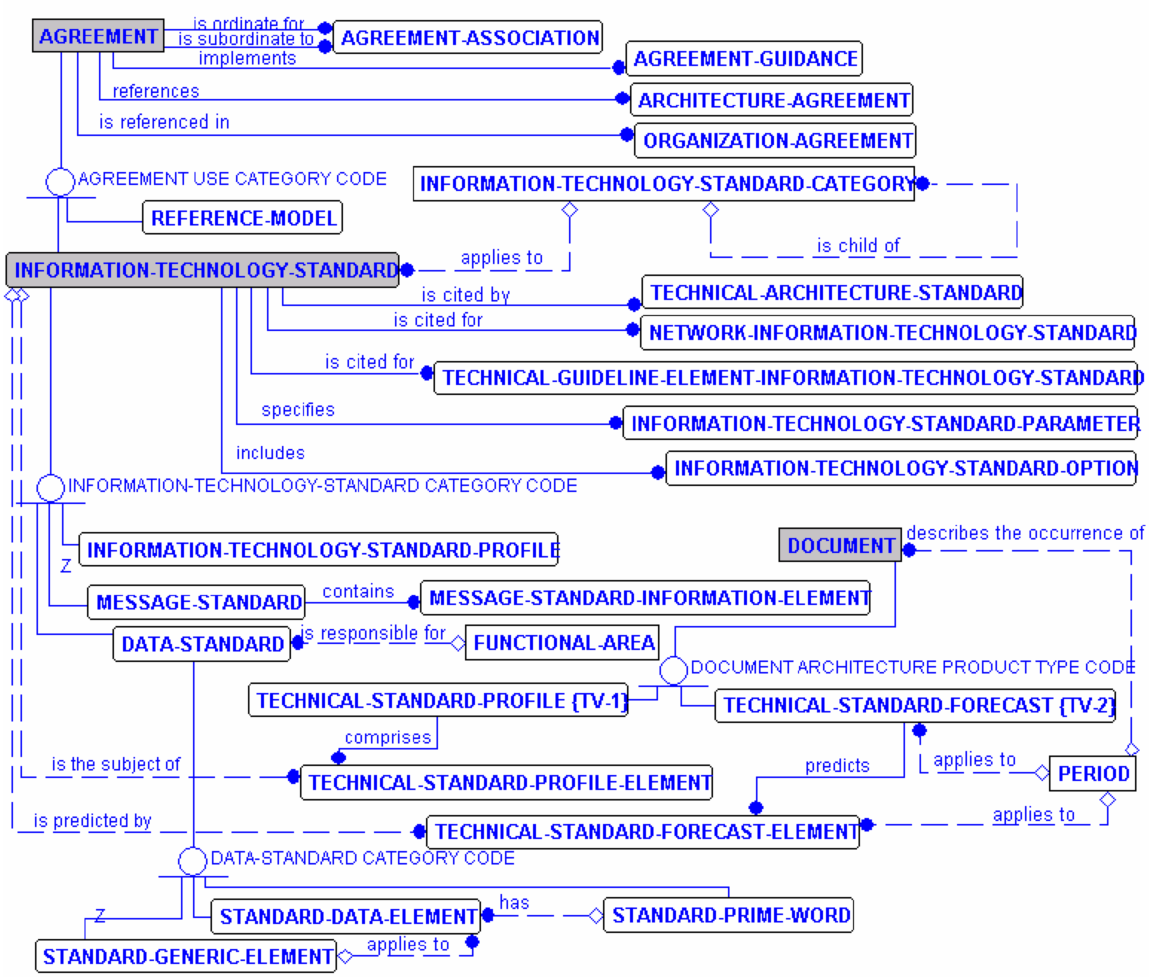


Figure 6-2. CADM ER Diagram for INFORMATION-TECHNOLOGY-STANDARD

Figure 6-3 extends the figure above to show how INFORMATION-TECHNOLOGY-STANDARD bridges SV and TV. Specifically, there are direct links from either INFORMATION-TECHNOLOGY-STANDARD or its parent (AGREEMENT) to all of the following:

- SV: NETWORK, NODE-LINK, NODE-COMMUNICATION-MEDIUM, NODE-PORT, NETWORK-DEVICE-MATERIEL, SYSTEM, SYSTEM-PROCESS-ACTIVITY (which also relates SYSTEM to SYSTEM-FUNCTION, a subtype of PROCESS-ACTIVITY), SOFTWARE-ITEM-INTERFACE, and TECHNICAL-INTERFACE
- TV: TECHNICAL-ARCHITECTURE, TECHNICAL-SERVICE-AREA, TECHNICAL-SERVICE, TECHNICAL-GUIDANCE-ELEMENT, TECHNICAL-STANDARD-PROFILE-ELEMENT, and TECHNICAL-STANDARD-FORECAST-ELEMENT

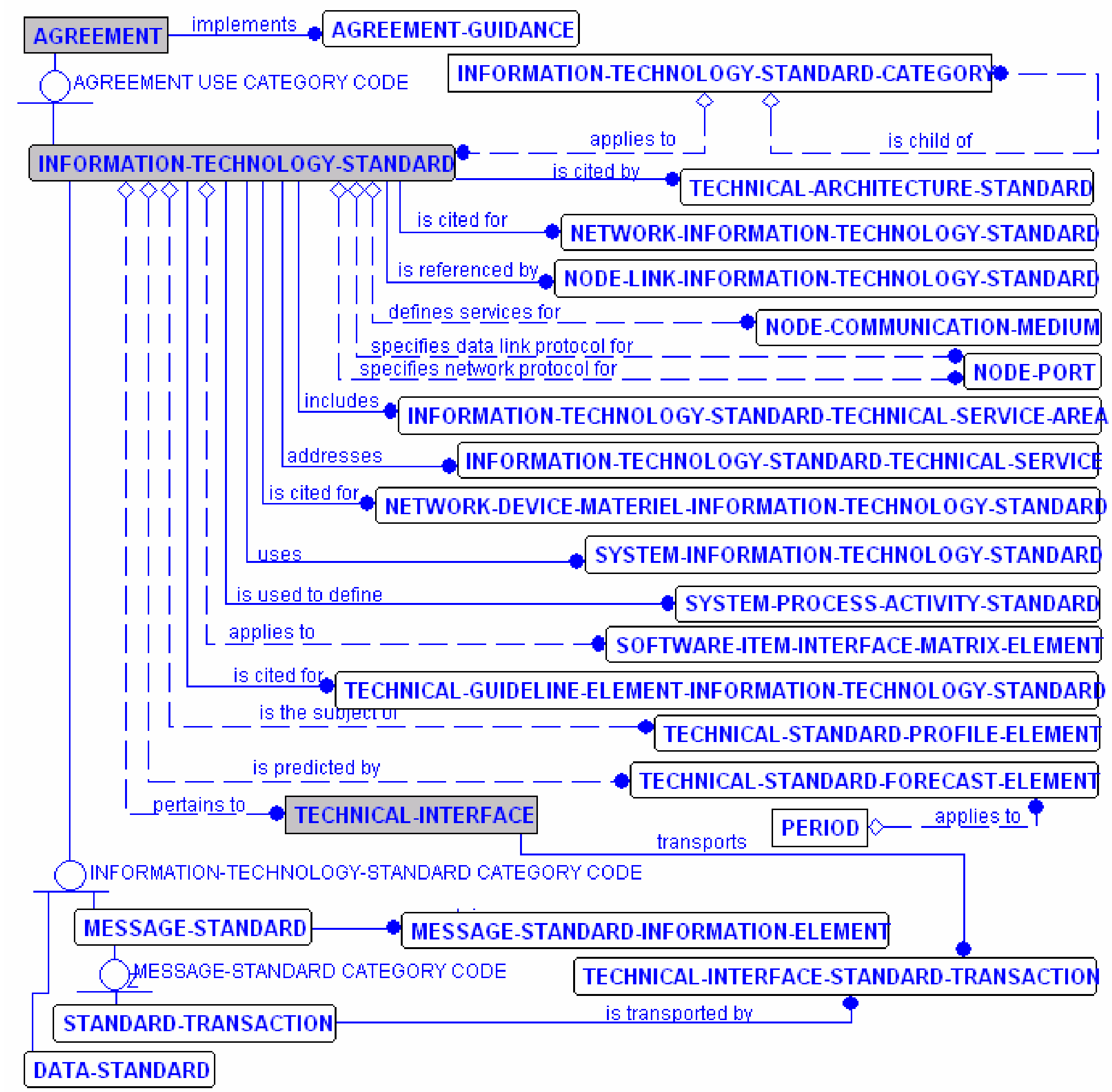


Figure 6-3. INFORMATION-TECHNOLOGY-STANDARD Links SV and TV Data Elements

6.1.5 CADM Support for Technical Standards Profile (TV-1)

TV-1 is stored as an instance of DOCUMENT with a category code designating it as a TECHNICAL-STANDARD-PROFILE (a subtype of DOCUMENT). The properties of each row of the TECHNICAL-STANDARD-PROFILE are specified using TECHNICAL-STANDARD-PROFILE-ELEMENT. Each row of TV-1 is focused on a specific standard. That standard is stored as an instance of AGREEMENT with a category code designating it as an INFORMATION-TECHNOLOGY-STANDARD, which may be further coded as being a MESSAGE-STANDARD, PROTOCOL-STANDARD, or INFORMATION-TECHNOLOGY-STANDARD-PROFILE (which identifies a set of standards, together with the implementation options and parameters chosen for those standards). More than one row of TV-1 may be focused on the same standard. Each row of TV-1 may specify a different IMPLEMENTATION-TIME-FRAME.

The hierarchy of standards (e.g., from the JTA and its annexes) is specified in INFORMATION-TECHNOLOGY-STANDARD-CATEGORY, at most one of which applies to each INFORMATION-TECHNOLOGY-STANDARD.

Where a standard includes specific references to other INFORMATION-TECHNOLOGY-STANDARD instances that are to be recorded in a CADM-structured database, AGREEMENT-ASSOCIATION is used to specify those related standards. AGREEMENT-DOCUMENT stores the identity of the standards document(s) issued by a standards-issuing body.

A TECHNICAL-ARCHITECTURE-PROFILE can apply to many architectures. All the ARCHITECTURE instances to which that DOCUMENT applies are specified in ARCHITECTURE-DOCUMENT.

Figure 6-4 provides a high-level diagram from the CADM showing key entities that are used to store architecture data for TV-1 in an architecture database.

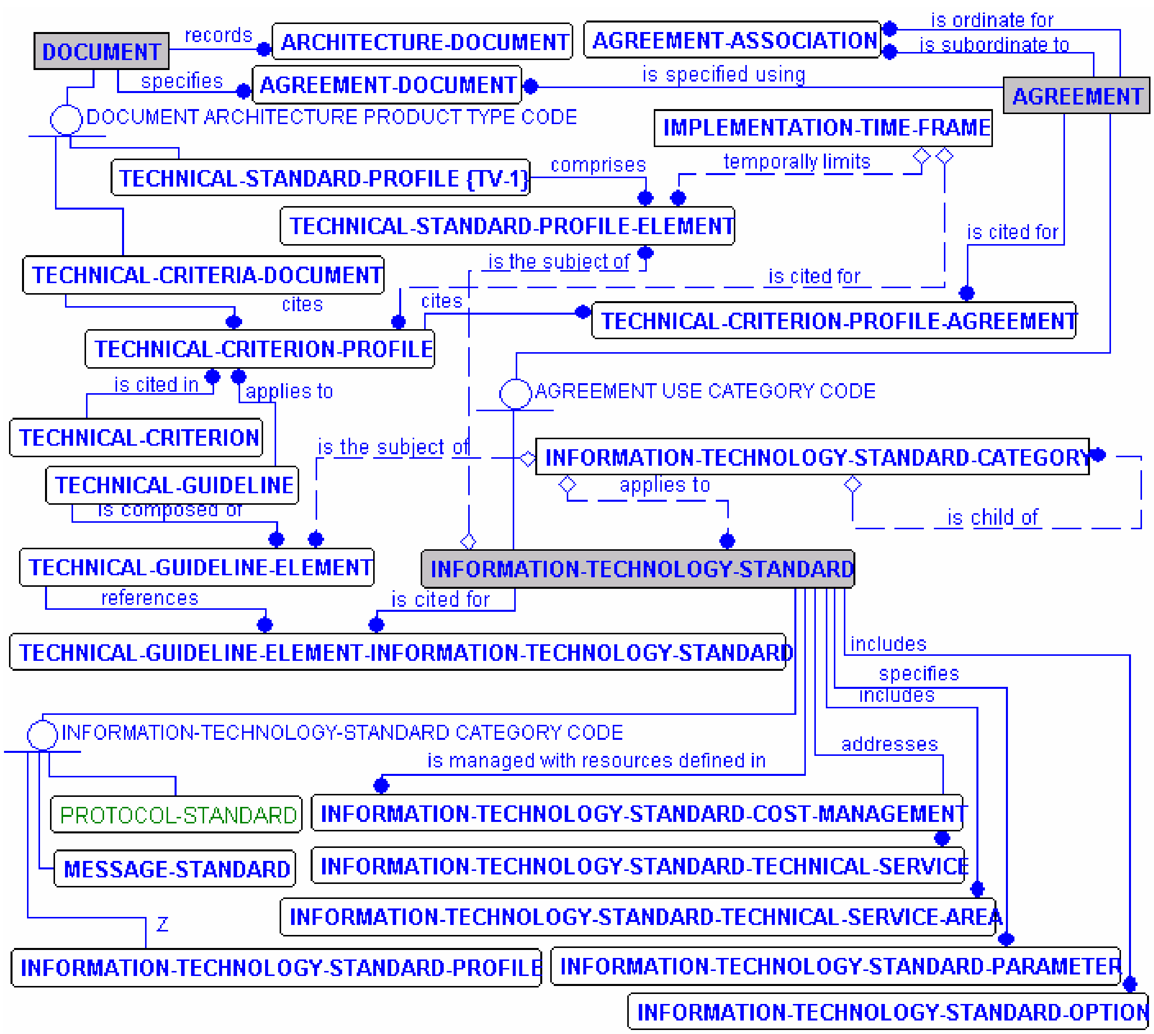


Figure 6-4. CADM ER Diagram for Technical Standards Profile (TV-1)

Figure 6-5 provides a high-level diagram from the CADM showing the entities that are used to store the entirety of the JTA Version 4.0 in an architecture database. The following bullets summarize how the JTA structure is represented in the CADM:

- Each version of the JTA is an instance of TECHNICAL-GUIDELINE.
- Each section of the JTA is a separate instance of TECHNICAL-GUIDELINE-ELEMENT, which has attributes to store the text of a specific section.
- INFORMATION-TECHNOLOGY-STANDARD-CATEGORY stores each instance as well as the hierarchical structure of the Table of Contents of the JTA (the hierarchy is represented using the is child of relationship for this entity and the hierarchy label is stored as an attribute of the entity).
- INFORMATION-TECHNOLOGY-STANDARD stores the individual standards together with the identifier of the applicable INFORMATION-TECHNOLOGY-STANDARD-CATEGORY. INFORMATION-TECHNOLOGY-STANDARD has an attribute that identifies the Open Systems Interconnection level (if applicable) and another attribute that identifies the development status of the standard.
- TECHNICAL-GUIDELINE-ELEMENT-INFORMATION-TECHNOLOGY-STANDARD identifies all instances of INFORMATION-TECHNOLOGY-STANDARD (if any) that are the subject of a specific section of the JTA. An attribute (Citation Code) of this entity indicates whether the standard is identified as mandatory or emerging in the JTA.

The link between the JTA and the templates defined above for TV-1 is through a new relationship from TECHNICAL-SERVICE to INFORMATION-TECHNOLOGY-STANDARD-CATEGORY.

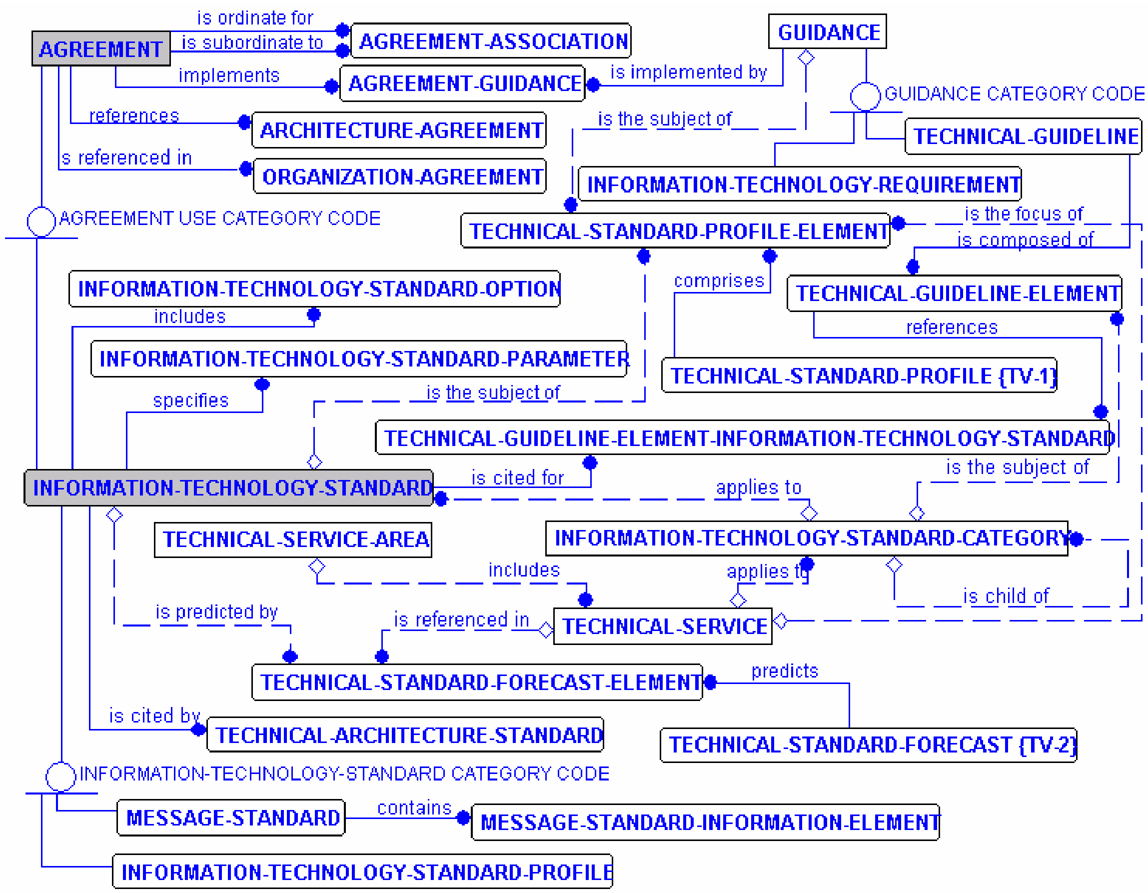


Figure 6-5. CADM ER Diagram for Joint Technical Architecture

6.2 TECHNICAL STANDARDS FORECAST (TV-2)

6.2.1 Technical Standards Forecast (TV-2) – Product Description

Product Definition. The Technical Standards Forecast contains expected changes in technology-related standards and conventions, which are documented in the TV-1 product. The forecast for evolutionary changes in the standards should be correlated against the time periods as mentioned in the SV-8 and SV-9 products.

Product Purpose. One of the prime purposes of this product is to identify critical technology standards, their fragility, and the impact of these standards on the future development and maintainability of the architecture and its constituent elements.

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.

The specific time periods selected (e.g., 6-month, 12-month, 18-month intervals) and the standards being tracked should be coordinated 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 impacts current architectures and influences the development of transition and objective (i.e., target) architectures. The forecast should be tailored to focus on standards areas that are related to the purpose for which a given architecture is being described and should 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 impact 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. A system's evolution, specified in SV-8, may be tied 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 (i.e., 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. The same template (see Table 6-2) shown in TV-1 may be used to describe TV-2.

6.2.2 UML Representation

There is no equivalent to this product in UML.

6.2.3 Technical Standards Forecast (TV-2) – Data Element Definitions

Table 6-5 describes the architecture data elements for TV-2.

Table 6-5. Data Element Definitions for Technical Standards Forecast (TV-2)

Data Elements	Attributes	Example Values/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 standard
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
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
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 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

Data Elements	Attributes	Example Values/Explanation
Timed Standards Forecast Impacts 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 impacted by the timed standards forecast
Timed Standards Forecast Impacts Communications System		
	Timed Standards Forecast Name	Name/identifier of a timed standards forecast
	Communications System Name	Name/identifier of a communications system (e.g., gateway, router, satellite) that could be impacted by the timed standards forecast
Timed Standards Forecast Impacts Communications Link		
	Timed Standards Forecast Name	Name/identifier of a timed standards forecast
	Communications Link Name	Name/identifier of a communications link that could be impacted by the timed standards forecast
Timed Standards Forecast Impacts Communications Network		
	Timed Standards Forecast Name	Name/identifier of a timed standards forecast
	Communications Network Name	Name/identifier of a communications network that could be impacted by the timed standards forecast
Timed Standards Forecast Impacts 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 impacted 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

6.2.4 CADM Support for Technical Standards Forecast (TV-2)

Figure 6-6 provides a high-level diagram from the CADM showing key entities that are used to store architecture data for TV-2 in a CADM-conformant database.

Each TV-2 is represented as an instance of TECHNICAL-STANDARD-FORECAST, a subtype of DOCUMENT, which has attributes for Name, Time Frame, Category Code (with domain value Technical Standards Forecast), etc. TV-2 is an instance of TECHNICAL-STANDARD-FORECAST, which is defined by one or more instances of TECHNICAL-STANDARD-FORECAST-ELEMENT. The latter contains identifiers for TECHNICAL-SERVICE, Time Frame PERIOD, INFORMATION-TECHNOLOGY-STANDARD-PROFILE, and INFORMATION-TECHNOLOGY-STANDARD.

As noted for TV-1, standards are separately specified in a subtype hierarchy for AGREEMENT. These include at the first level INFORMATION-TECHNOLOGY-STANDARD, REFERENCE-MODEL, and INFORMATION-TECHNOLOGY-STANDARD-PROFILE. Subtypes of INFORMATION-TECHNOLOGY-STANDARD are MESSAGE-STANDARD, PROTOCOL-STANDARD, DATA-MODEL-STANDARD, and DATA-STANDARD (for data elements and data entities). Each INFORMATION-TECHNOLOGY-STANDARD may have one or more instances of INFORMATION-TECHNOLOGY-STANDARD-OPTION and INFORMATION-TECHNOLOGY-STANDARD-PARAMETER. DOCUMENT may be used to specify an ARCHITECTURE-FINDING, such as a constraint, issue, impact, or recommendation.

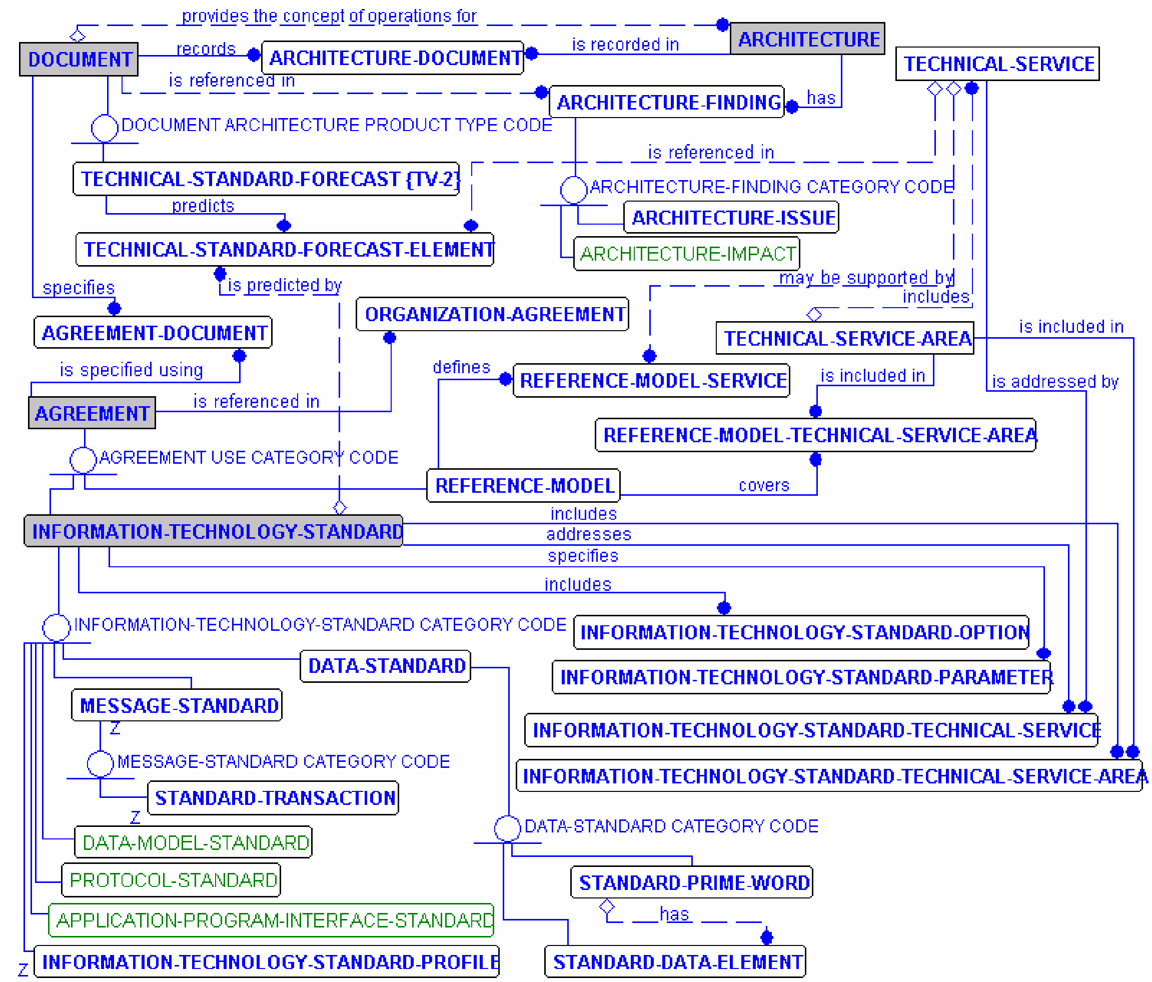


Figure 6-6. CADM ER Diagram for Technical Standards Forecast (TV-2)

7 FRAMEWORK ARCHITECTURE DATA ELEMENT RELATIONSHIPS

7.1 OVERVIEW

The three views (Operational View [OV], Systems View [SV], and Technical Standards View [TV]) of the DoD Architecture Framework (DoDAF) provide different areas of emphasis into a single, integrated architecture, including defined work processes and supporting Information Technology (IT). The logical linkages among the architecture data elements underlying the products and the views serve to ensure that the single architecture so described can actually be built and operated. In particular, these linkages ensure that the architecture remains mutually consistent. The linkages provide traceability from view to view, from product to product within a view, and across views that ensures:

- 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

This section discusses these linkages by summarizing the major relationships among the various DoDAF products.

7.2 ARCHITECTURE DATA ELEMENT RELATIONSHIPS ACROSS THREE VIEWS

Figure 7-1 summarizes major relationships among the OV, SV, and TV architecture products. Major architecture data element relationships (belonging to these products) include:

- Operational nodes in the Operational Node Connectivity Description (OV-2) map to operational activities of the Operational Activity Model (OV-5), which are conducted by these operational nodes.
- Needlines in the Operational Node Connectivity Description (OV-2) map to information exchanges in the Operational Information Exchange Matrix (OV-3), which detail the information exchanges between these operational nodes.
- Operational Information Exchange Matrix (OV-3) is constructed from system data elements described in the Logical Data Model (OV-7).
- The Operational Rules Model (OV-6a) contains structural and validity rules constraining the Operational Activity Model (OV-5).
- The Operational Rules Model (OV-6a) also contains structural and validity rules constraining the Logical Data Model (OV-7).
- The Systems Rules Model (SV-10a) contains structural and validity rules constraining Systems Functionality Description (SV-4).
- Operational nodes in the Operational Node Connectivity Description (OV-2) map to the systems in Systems Interface Description (SV-1), which support these operational nodes by providing automated support.
- Needlines in the Operational Node Connectivity Description (OV-2) map to interfaces in the Systems Interface Description (SV-1), which represent an automated needline.
- Interfaces in a Systems Interface Description (SV-1) map to system data elements exchanged in the Systems Data Exchange Matrix (SV-6), which in turn map to system data flows in the System Functionality Description (SV-4).
- System functions in a Systems Functionality Description (SV-4) map to the systems in Systems Interface Description (SV-1), which execute these functions.
- System functions in a Systems Functionality Description (SV-4) implement automated portions of the operational activities in an Operational Activity Model (OV-5). The Operational Activity to Systems Function Traceability Matrix (SV-5) documents this relationship.
- System data exchanges of the Systems Data Exchange Matrix (SV-6) map to the information exchanges of the Operational Information Exchange Matrix (OV-3). System data exchanges constitute the automated portions of the operational information exchanges.
- Physical Schema (SV-11) implements and details the elements of the Logical Data Model (OV-7).

- The Technical Standards Profile (TV-1) documents standards that constrain the Systems Interface Description (SV-1), the Systems Communications Description (SV-2), the Systems Functionality Description (SV-4), the Systems Data Exchange Matrix (SV-6), the Logical Data Model (OV-7), and the Physical Schema (SV-11).
- The Systems Evolution Description (SV-8) documents system evolution timelines and milestones that have corresponding time periods associated with timed technology forecasts in the Systems Technology Forecast (SV-9) and timed standard forecasts in Technical Standards Forecast (TV-2).

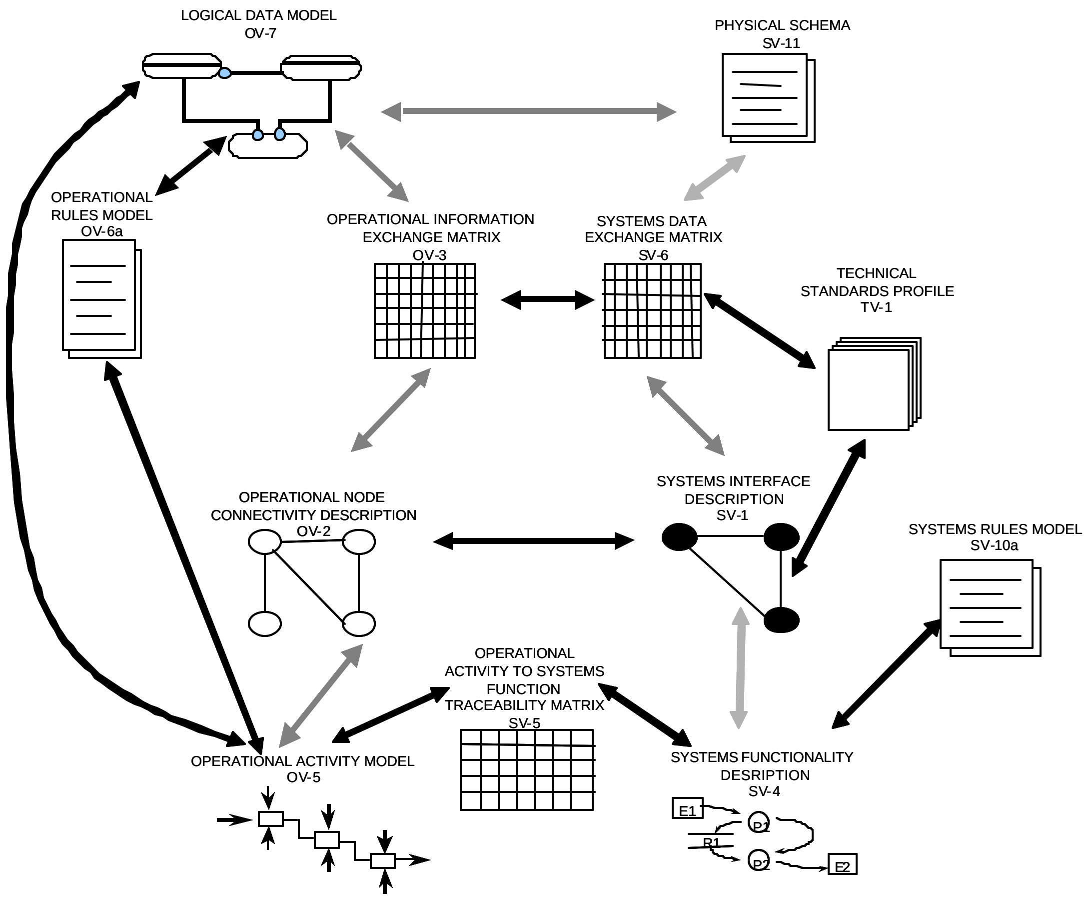


Figure 7-1. Major Product Relationships

7.3 OPERATIONAL VIEW ARCHITECTURE DATA ELEMENT RELATIONSHIPS

Figure 7-2 covers relationships among products describing operations in the architecture's OV. Some relationships that are illustrated in Figure 7-2 may not be listed below, because they have already been stated in section 7.2. Figure 7-2 summarizes the following major architecture data element relationships:

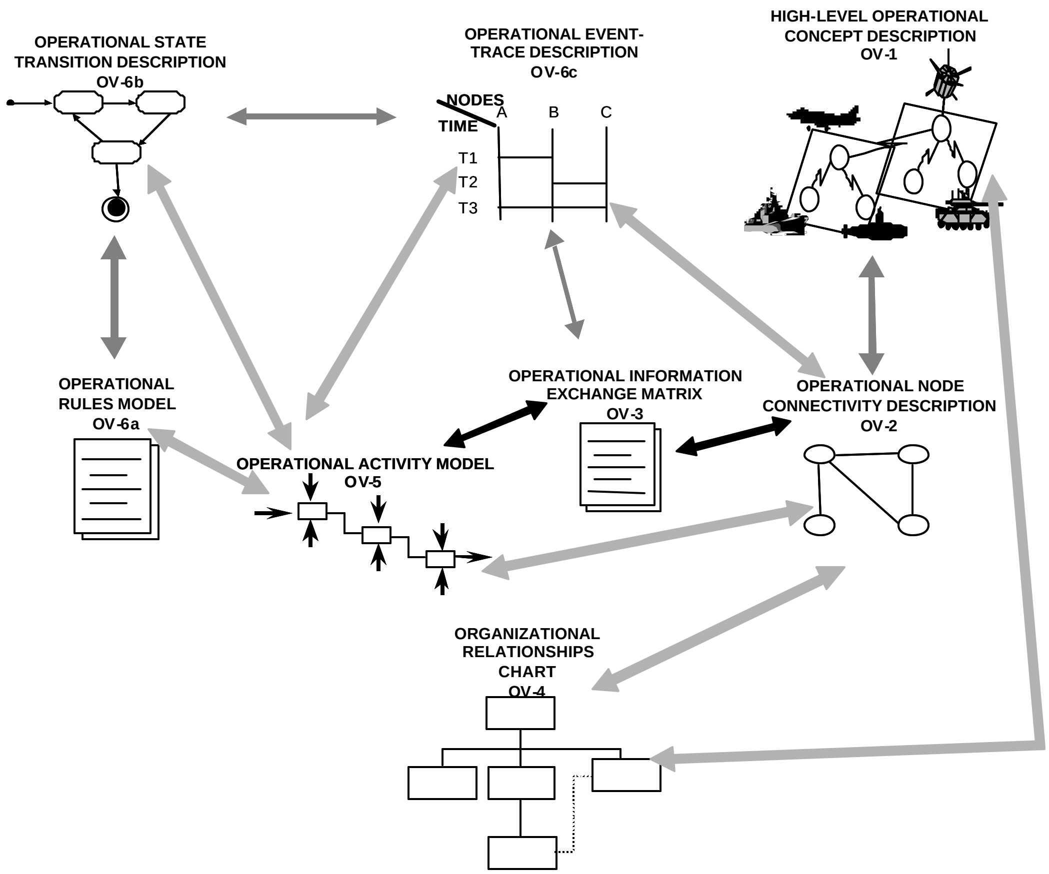


Figure 7-2. Operational View Product Relationships

- Organizations in the High-Level Operational Concept Graphic (OV-1) should match those in the Operational Node Connectivity Description (OV-2) and annotations identifying responsible operational nodes in the Operational Activity Model (OV-5).
- Operational nodes in the Operational Node Connectivity Description (OV-2) match the lifelines of the Operational Event-Trace Description (OV-6c).
- Organizations in the Organizational Relationships Chart (OV-4) map to the operational nodes in the Operational Node Connectivity Description (OV-2).
- Information elements of the Information Exchange Matrix (OV-3) map to the inputs and outputs (I/Os) that belong to activities of the Operational Activity Model (OV-5) conducted across two or more operational nodes of the Operational Node Connectivity Description (OV-2).

- Events in the Operational State Transition Description (OV-6b) map to events in the Operational Event-Trace Description (OV-6c).
- State transitions in the Operational State Transition Description (OV-6b) and events in the Operational Event-Trace Description (OV-6c) should be consistent with activities in the Operational Activity Model (OV-5).
- Dynamic rules in the Operational Rules Model (OV-6a) may constrain state transitions in the State Transition Description (OV-6b) or decision points in the Operational Activity Model (OV-5).
- Events in the Operational State Transition Description (OV-6b) and the Operational Event-Trace Description (OV-6c) map to triggering events in the Information Exchange Matrix (OV-3).

7.4 SYSTEMS VIEW ARCHITECTURE DATA ELEMENT RELATIONSHIPS

Figure 7-3 summarizes relationships among products of the SV. Major architecture data element relationships include:

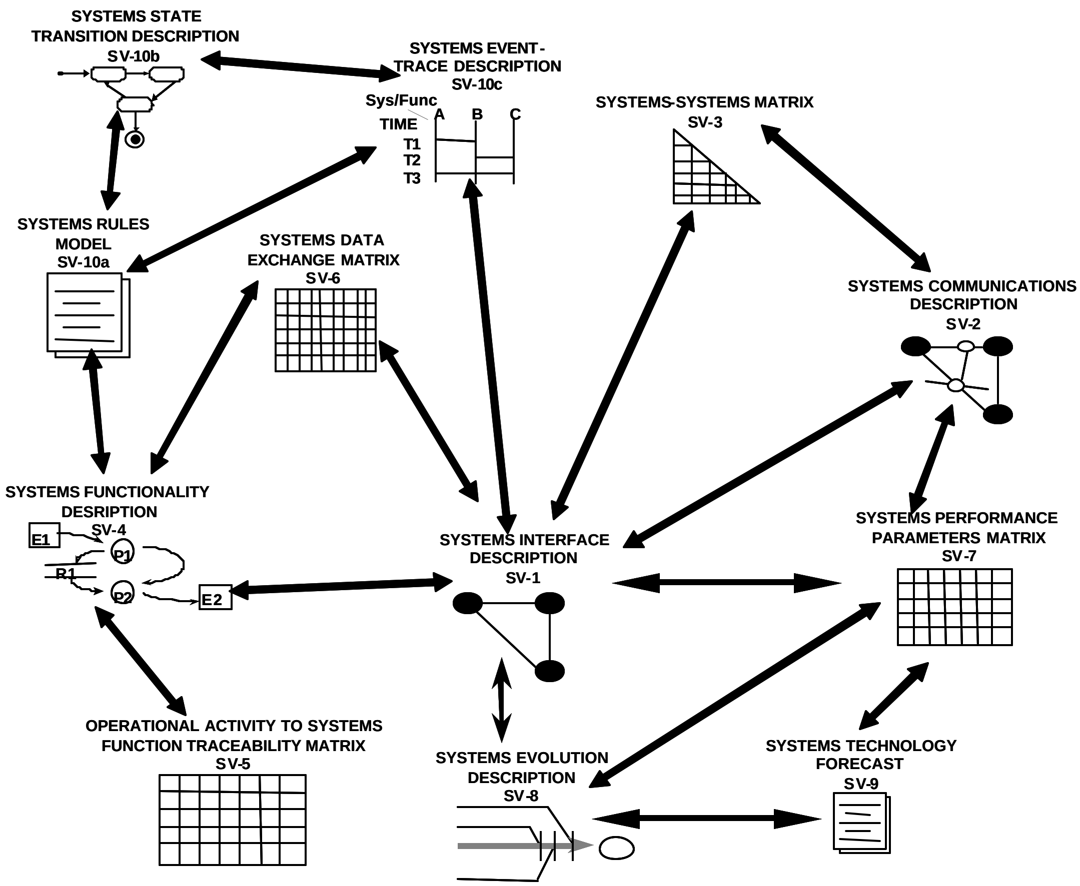


Figure 7-3. Systems View Product Relationships

- Interfaces in a Systems Interface Description (SV-1) map to system data exchanges in the Systems Data Exchange Matrix (SV-6).
- System data input or output by system functions in the Systems Functionality Description (SV-4) map to system data elements of the Systems Data Exchange Matrix (SV-6).
- Interfaces in the Systems-Systems Matrix (SV-3), communications links in the System Communications Description (SV-2), and some performance requirements in Systems Performance Parameters Matrix (SV-7) map to interfaces in the Systems Interface Description (SV-1).
- The Systems Evolution Description (SV-8) phases interface requirements and implementation in the Systems Interface Description (SV-1).
- The Systems Evolution Description (SV-8) phases the applicability of performance requirements in the Systems Performance Parameters Matrix (SV-7) and the availability of new products and technologies in the Systems Technology Forecast (SV-9).

- Rules in the Systems Rules Model (SV-10a) constrain transitions in the Systems State Transition Description (SV-10b), events in Systems Event-Trace Description (SV-10c), and functions in the Systems Functionality Description (SV-4).
- Events in the Systems State Transition Description (SV-10b) map to events in the Systems Event-Trace Description (SV-10c).
- Lifelines in the Systems Event-Trace Description (SV-10c) map to systems in the Systems Interface Description (SV-1) and system functions in the Systems Functionality Description (SV-4).

7.5 SYSTEM ELEMENTS THAT MAP TO STANDARDS

Figure 7-4 illustrates the systems products (and associated systems elements) that have technical standards current (TV-1) or forecast (TV-2) associated with them. These architecture data elements include:

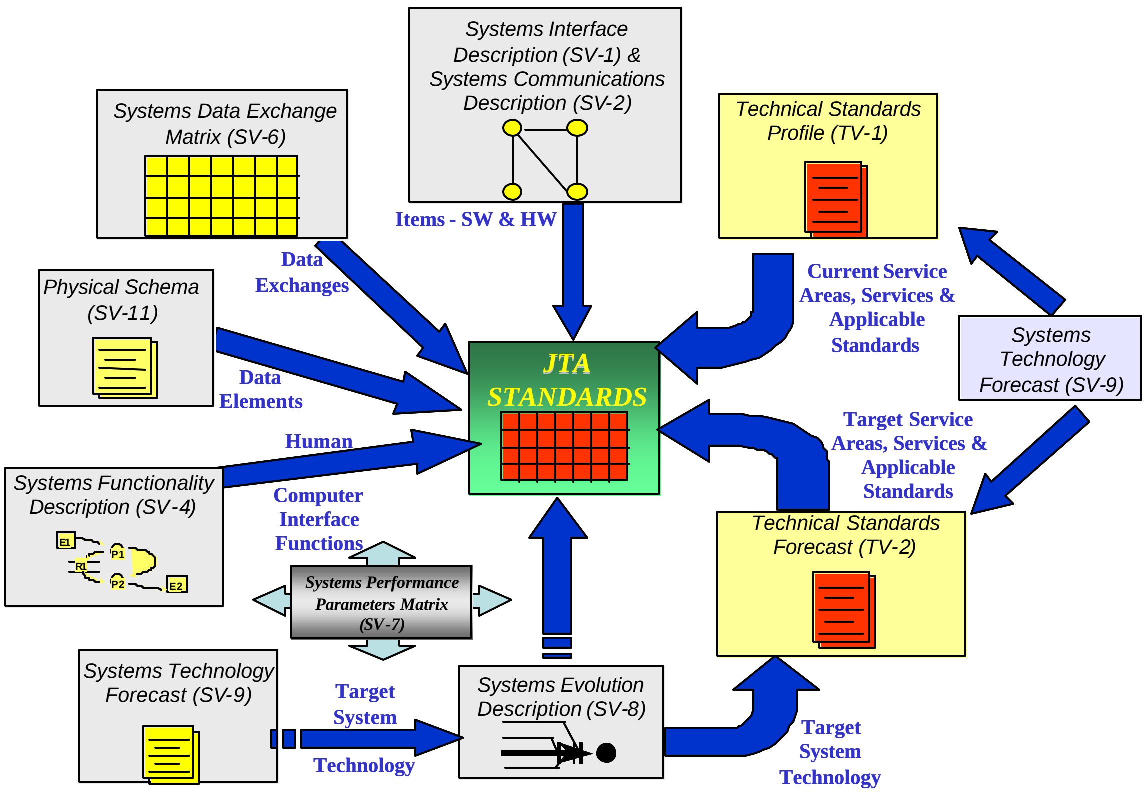


Figure 7-4. Detail of Systems Elements that are Associated with Standards

- Systems and system hardware/software items (software and hardware) from the Systems Interface Description (SV-1)
- Communications links, communications paths, communications networks, and communications systems from the System Communications Description (SV-2)
- System functions from the Systems Functionality Description (SV-4)
- System data elements from the Systems Data Exchange Matrix (SV-6)
- Evolving systems from the Systems Evolution Description (SV-8)
- Data Modeling techniques used in the Logical Data Model (OV-7) and the Physical Schema (SV-11)
- Timed technology forecasts in Systems Technology Forecast (SV-9)

7.6 SUMMARY OF ARCHITECTURE DATA ELEMENT RELATIONSHIPS

Table 7-1 details these relationships and others among the products in the Framework. 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 twice.

Table 7-1. Detailed Architecture Data Element Relationships

LINK	TO...	BY:
OV-1		
	OV-2	Organizations, organization types, and/or 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/or 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/or 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	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-6c	Lifelines in OV-6c should map to operational nodes in OV-2.
	SV-1	Operational node in an OV-2 may be supported by one or more systems in SV-1 (indicating that the operational node owns/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 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) in OV-5, 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.
	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/or 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 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) in OV-5, if OV-5 decomposes to a level that permits such a mapping. 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 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…	BY:
	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	Operational node in OV-2 may be supported by one or more systems in SV-1 (indicating that the operational node owns/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 link(s) or communications network(s) in SV-2.
	SV-3	One entry in 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 impact systems, subsystems and system hardware/software items in SV-1.
SV-2		
	SV-1	An interface in SV-1 is implemented by communications link(s) or communications network(s) in SV-2.
	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 impact 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 impact communications systems, communications links, and communications networks in SV-2.
SV-3		
	SV-1	One entry in 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 impact 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		
	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 depicted by one of the interfaces in SV-1; an interface supports one or more system data exchanges.

LINK	TO…	BY:
	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 impact 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 (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.
	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 impact systems, subsystems, and system hardware/software items of SV-1.
	SV-2	Timed technology forecasts in SV-9 impact communications systems, communications links, and communications networks in SV-2.
	SV-7	If the future performance expectations (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		
	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.

LINK	TO…	BY:
	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 impact systems, subsystems and system hardware/software items in SV-1.
	SV-2	Timed standard forecasts in TV-2 impact communications systems, communications links, and communications networks in SV-2.
	SV-4	Timed standard forecasts in TV-2 impact system functions in SV-4.
	SV-6	Timed standard forecasts in TV-2 impact 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.

ANNEX A
GLOSSARY

A

A&I	Architecture and Interoperability
A&T	Acquisition and Technology
ACC	Architecture Coordination Council
ACL	Access Control List
AFWG	Architecture Framework Working Group
AMS	Acquisition Management System
AoA	Analysis of Alternatives
API	Application Program Interface
ASD(C3I)	Assistant Secretary of Defense for Command, Control, Communications, and Intelligence
ASD(NII)	Assistant Secretary of Defense for Networks and Information Integration
AV	All-Views
AWG	Architecture Working Group

B

BPR	Business Process Reengineering
BRM	Business Reference Model

C

C2	Command and Control
C3	Command, Control, and Communications
C3	Command, Control, and Consultation (NATO usage)
C3I	Command, Control, Communications, and Intelligence
C4I	Command, Control, Communications, Computers, and Intelligence
C4ISP	Command, Control, Communications, Computers, and Intelligence Support Plan
C4ISR	Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance
CADM	Core Architecture Data Model
CC	Combatant Commander
CCA	Clinger-Cohen Act of 1996 (also referred to as ITMRA)
CDD	Capability Development Document
CED	Capability Evolution Description

CIO	Chief Information Officer
CJCS	Chairman Joint Chiefs of Staff
CJCSI	Chairman of the Joint Chiefs of Staff Instruction
CJCSM	Chairman of the Joint Chiefs of Staff Manual
CONOPS	Concept of Operations
COTS	Commercial Off-the-Shelf
CPD	Capability Production Document
CPN	Colored PetriNet
CRD	Capstone Requirements Document
C/S/As	Commands, Services, and Agencies

D

DARS	DoD Architecture Repository System
DBMS	Database Management System
DDDS	DoD Data Dictionary System
DDL	Data Definition Language
DFD	Data Flow Diagram
DIAD	Department of the Navy Integrated Architecture Database
DoD	Department of Defense
DoDAF	DoD Architecture Framework
DoDD	DoD Directive
DoDI	DoD Instruction
DON CIO	Department of the Navy Chief Information Officer
DOTLPF	Doctrine, Organization, Training, Leadership & education, Personnel, and Facilities
DOTMLPF	Doctrine, Organization, Training, Materiel, Leadership & education, Personnel, and Facilities
DSS	Decision Support System
DTLOMS	Doctrine, Training, Leader Development, Organizations, Materiel, Soldiers

E

EEI	External Environment Interface
ER	Entity-Relationship
ERD	Entity Relationship Diagram

F

FAA	Functional Area Analysis
FEA	Federal Enterprise Architecture

FIPS	Federal Information Processing Standard
FNA	Functional Needs Analysis
FOC	Full Operational Capability
FoS	Family of Systems
FPI	Functional Process Improvement
FRP	Full Rate Production
FSA	Functional Solution Analysis

G

GCCS	Global Command and Control System
GIG	Global Information Grid
GUI	Graphical User Interface

H

HCI	Human Computer Interface
HR	Human Resources

I

IAP	Integrated Architectures Panel
IC	Intelligence Community
ICD	Initial Capabilities Document
ICD	Interface Control Document
ICOM	Input, Control, Output, and Mechanism
IDEF0	Integrated Definition for Activity Modeling
IDEF1X	Integrated Definition for Data Modeling
IER	Information Exchange Requirement
I/O	Input and Output
IOC	Initial Operational Capability
IOT&E	Initial Operational Test & Evaluation
IRM	Information Resources Management
ISP	Information Support Plan
IT	Information Technology
ITF	Integration Task Force
ITMRA	Information Technology Management Reform Act

J

JCIDS	Joint Capabilities Integration and Development System
JCS	Joint Chiefs of Staff

JF	Joint Forces
JIC	Joint Intelligence Center
JMA	Joint Mission Area
JOA	Joint Operational Architecture
JOC	Joint Operations Center
JROC	Joint Requirements Oversight Council
JSA	Joint Systems Architecture
JTA	Joint Technical Architecture
JTF	Joint Task Force
JWCA	Joint Warfighting Capability Assessment

K

KI	Key Interface
KIP	Key Interface Profile
KPP	Key Performance Parameters

L

LAN	Local Area Network
LISI	Levels of Information Systems Interoperability
LRIP	Low Rate Initial Production

M

MAIS	Major Automated Information System
MCEB	Military Communications Electronics Board
MCP	Mission Capability Package
MDAPS	Major Defense Acquisition Programs
MOE	Measure of Effectiveness
MOP	Measure of Performance
M&S	Modeling and Simulation
MTOE	Modified Table of Organization and Establishment

N

NATO	North Atlantic Treaty Organization
NCO	Net-Centric Operations
NCOW	Net-Centric Operations and Warfare
NIPRNET	Non-Secure Internet Protocol Router Network
NRO	National Reconnaissance Office
NSS	National Security Systems

O

OA	Operational Activity
OASD	Office of the Assistant Secretary of Defense
OIEM	Operational Information Exchange Matrix
OMB	Office of Management and Budget
OMG	Object Management Group
ONCD	Operational Node Connectivity Description
OO	Object-Oriented
OPLAN	Operational Plan
OSD	Office of the Secretary of Defense
OV	Operational View

P

PM	Program Manager
PPBE	Planning, Programming, Budgeting, and Execution
PPBS	Planning, Programming, and Budgeting System
PRM	Performance Reference Model
PSA	Principal Staff Assistants (OSD officials holding Presidential appointments, Assistants to the Secretary of Defense, and OSD Directors or equivalents who report directly to the Secretary or Deputy Secretary of Defense [DoDI 5025.1])

R

RGS	Requirements Generation System
ROI	Return On Investment

S

SA	Structured Analysis
SDEM	Systems Data Exchange Matrix
SECDEF	Secretary of Defense
SF	System Function
SIPRNET	Secret Internet Protocol Router Network
SoS	System of Systems
SRM	Service Component Reference Model
SSL	Secure Sockets Layer
SSM	Systems-Systems Matrix
SV	Systems View

T

TOE	Tables of Organization and Establishment
TPPU	Task, Post, Process, and Use
TRM	Technical Reference Model
TTP	Tactics, Techniques, and Procedures
TV	Technical Standards View

U

UJTL	Universal Joint Task List
UML	Unified Modeling Language
UOB	Unit of Behavior
URR	Universal Reference Resource
U.S.	United States
USD(A&T)	Under Secretary of Defense for Acquisition and Technology
USD(AT&L)	Under Secretary of Defense for Acquisition, Technology, and Logistics

V

VJTA	Virtual Joint Technical Architecture
VPN	Virtual Private Network

W

WAN	Wide Area Network

X

XML	Extensible Markup Language

ANNEX B

DICTIONARY OF TERMS

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

*** Definitions shared between the Framework and CADM documents**

Analysis of Alternatives	The evaluation of operational effectiveness, operational suitability, and estimated costs of alternative systems to meet a mission capability. The analysis assesses the advantages and disadvantages of alternatives being considered to satisfy capabilities, including the sensitivity of each alternative to possible changes in key assumptions or variables. (CJCSI 3170.01C)
Analysis of Materiel Approaches	The JCIDS analysis to determine the best materiel approach or combination of approaches to provide the desired capability or capabilities. Though the AMA is similar to an AoA, it occurs earlier in the analytical process. Subsequent to approval of an ICD, which may lead to a potential ACAT I/IA program, Director Program Analysis & Evaluation provides specific guidance to refine this initial AMA into an AoA. (CJCSI 3170.01C)
Architecture Data Element	One of the data elements that make up the Framework products. Also referred to as architecture data type. (DoDAF)
Attribute	A property or characteristic. (Derived from DATA-ATTRIBUTE, DDDS 4363 (A)) A testable or measurable characteristic that describes an aspect of a system or capability. (CJCSI 3170.01C)
Capability	The ability to execute a specified course of action. (JP 1-02) It is defined by an operational user and expressed in broad operational terms in the format of an initial capabilities document or a DOTMLPF change recommendation. In the case of materiel proposals, the definition will progressively evolve to DOTMLPF performance attributes identified in the CDD and CPD. (CJCSI 3170.01C)
Capability Gaps	Those synergistic resources (DOTMLPF) that are unavailable but potentially attainable to the operational user for effective task execution. (CJCSI 3170.01C)
Capability Development Document	A document that captures the information necessary to develop a proposed program(s), normally using an evolutionary acquisition strategy. The CDD outlines an affordable increment of military useful, logistically supportable, and technically mature capability. (CJCSI 3170.01C)
Capability Production Document	A document that addresses the production elements specific to a single increment of an acquisition program. (CJCSI 3170.01C)
Capstone Requirements Document	A document that contains capability-based requirements that facilitates the development of CDDs and CPDs by providing a common framework and operational concept to guide their development. (CJCSI 3170.01C)

Communications Medium*	A means of data transmission.
Data	A representation of individual facts, concepts, or instructions in a manner suitable for communication, interpretation, or processing by humans or by automatic means. (IEEE 610.12)
Data Model	A representation of the data elements pertinent to an architecture, often including the relationships among the elements and their attributes or characteristics. (DoDAF)
Data-Entity*	The representation of a set of people, objects, places, events or ideas that share the same characteristic relationships. (DDDS 4362 (A))
Defense Acquisition System	The management process by which the Department of Defense provides effective, affordable, and timely systems to the users. (DoDD 5000.1)
DoD Component	The DoD Components consist of the Office of the Secretary of Defense, the Military Departments, the Chairman of the Joint Chiefs of Staff, the combatant commands, the Office of the Inspector General of the Department of Defense, the Defense agencies, the DoD field activities, and all other organizational entities within the Department of Defense. (DoDD 8100.01)
Family of Systems	A set or arrangement of independent systems that can be arranged or interconnected in various ways to provide different capabilities. (DoDD 4630.5)
Format	The arrangement, order, or layout of data/information. (Derived from IEEE 610.5)
Functional Area*	A major area of related activity (e.g., Ballistic Missile Defense, Logistics, or C2 support). (DDDS 4198 (A))
Information	The refinement of data through known conventions and context for purposes of imparting knowledge.
Information 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. (DoDAF)
Information Exchange	The collection of information elements and their performance attributes such as timeliness, quality, and quantity values. (DoDAF)
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 executive agency. This includes equipment used by a DoD Component directly, or used by a contractor under a contract with the Component, which (i) requires the use of such equipment, or (ii) requires the use, to a significant extent, of such equipment in the performance of a service or the furnishing of a product. The term "IT" also includes computers, ancillary equipment, software, firmware and similar procedures, services (including support services), and related resources. Notwithstanding the above, the term "IT" does not include any equipment that is acquired by a Federal contractor incidental to a Federal contract. The term "IT" includes National Security Systems (NSS). (DoDD 4630.5)

Initial Capabilities Document	Documents the need for a materiel approach to a specific capability gap derived from an initial analysis of materiel approaches executed by the operational user and, as required, an independent analysis of materiel alternatives. It defines the capability gap in terms of the functional area, the relevant range of military operations, desired effects and time. The ICD summarizes the results of the DOTMLPF analysis and describes why non-materiel changes alone have been judged inadequate in fully providing the capability. (CJCSI 3170.01C)
Integrated Architecture	An architecture consisting of multiple views or perspectives (Operational View, Systems View, and Technical Standards View) that facilitates integration and promotes interoperability across family of systems and system of systems and compatibility among related architectures (DoDD 4630.5) 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 defined to be 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. (DoDAF)
Interoperability	The ability of systems, units, or forces to provide data, information, materiel, and services to and accept the same from other systems, units, or forces and to use the data, information, materiel, and services so exchanged to enable them to operate effectively together. IT and NSS interoperability includes both the technical exchange of information and the end-to-end operational effectiveness of that exchange of information, as required, for mission accomplishment. (DoDD 4630.5)
Joint Capabilities Integrated Development System	Policy and procedures that support the Chairman of the Joint Chiefs of Staff and the Joint Requirements Oversight Council in identifying, assessing, and prioritizing joint military capability needs. (CJCSI 3170.01C)
Key Performance Parameters	Those minimum attributes or characteristics considered most essential for an effective military capability. KPPs are validated by the JROC for JROC interest documents, by the Functional Capabilities Board for Joint Impact documents, and by the DoD Component for Joint Integration or Independent documents. CDD and CPD KPPs are included verbatim in the Acquisition Program Baseline. (CJCSI 3170.01C)
Link	A representation of the physical realization of connectivity between systems nodes.
Mission Area*	The general class to which an operational mission belongs. (DDDS 2305(A)) Note: Within a class, the missions have common objectives.
Mission*	An objective together with the purpose of the intended action. (Extension of DDDS 1(A)) Note: Multiple tasks accomplish a mission. (Space and Naval Warfare Systems Command)
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.
National Security Systems	Telecommunications and information systems operated by the Department of Defense – the functions, operation, or use of which (1) involves intelligence activities, (2) involves cryptologic activities related to national security, (3) involves the command and control of military forces, (4) involves equipment that is an integral part of a weapon or weapons systems, or (5) is critical to the direct fulfillment of military or intelligence missions. Subsection (5) in the preceding sentence does not include procurement of automatic data processing equipment or services to be used for routine administrative and business applications (including payroll, finance, logistics, and personnel management applications). (DoDD 4630.5)
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. (DoDAF)
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. (DoDAF)
Operational Node	A node that performs a role or mission. (DoDAF)
Organization*	An administrative structure with a mission. (DDDS 345 (A))
Planning, Programming, Budgeting, and Execution Process	The primary resource allocation process of the DoD. One of three major decision support systems for defense acquisition, PPBE is a systematic process that guides DoD's strategy development, identification of needs for military capabilities, program planning, resource estimation and allocation, acquisition, and other decision processes.
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. (Army Systems Architecture Framework) Note: Multiple activities make up a process. (Space and Naval Warfare Systems Command)
Report	The DoDAF defines a report to be architecture data elements from one or more products combined with additional information. Reports provide a different way of looking at architecture data.
Requirement*	A need or demand. (DDDS 12451/1 (D))
Role	A function or position. (Webster's)
Rule	Statement that defines or constrains some aspect of the enterprise.
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. (DoDAF)
System Data Element	A basic unit of data having a meaning and distinct units and values. (Derived from 8320.1) 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. (DoDAF)
System Data Exchange	The collection of System Data Elements and their performance attributes such as timeliness, quality, and quantity values. (DoDAF)
System Function*	A data transform that supports the automation of activities or information elements exchange. (DoDAF)
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. (DoDAF)
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. (DoDD 4630.5)
Task	A discrete unit of work, not specific to a single organization, weapon system, or individual, that enables missions or functions to be accomplished. (Extension from UJTL, JCSM 3500.04A, 1996). Note: Multiple processes accomplish a task; a single process may support multiple tasks. (Space and Naval Warfare Systems Command)
Universal Reference Resources	Reference models and information standards that serve as sources for guidelines and attributes that must be consulted while building architecture products. (DoDAF)

ANNEX C

DICTIONARY OF UML TERMS

The terms included here are UML terms. They convey some restrictive or special sense in this section. The sources for these definitions are [Booch, 1999] and [Rumbaugh, 1999].

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 distinguish 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. See: active object.
Active Object	An object that owns a thread and can initiate control activity. An instance of active class. See: active class, thread.
Activity Graph	A special case of a state machine that is used to model processes involving one or more classifiers. Contrast: statechart diagram.
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. See: aggregation.
Aggregation	A special form of association that specifies a whole-part relationship between the aggregate (whole) and a component part. See: composition.
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	An attribute is a named property of a class that describes a range of values that instances of the property may hold.
Building Blocks	There are three kinds of building blocks in UML: things, relationships, and diagrams.
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. See: interface.
Class Diagram	A diagram that shows a collection of declarative (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. See: 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. See: sequence diagram.
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). Contrast: artifact.
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 UML. See: tagged value, stereotype.
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.
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. See: component diagrams.
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	A [sic] 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). UML supports the following diagrams: class diagram, object diagram, use case diagram, sequence diagram, collaboration diagram, state diagram, activity diagram, component diagram, and deployment diagram.
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 state 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 defined 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. See: extension point, include.
Guard	A Boolean predicate that provides a fine-grained control over the firing of the transition. It must be true for the transition to be fired. 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. See: generalization.
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.
Inheritance	The mechanism by which more specific elements incorporate structure and behavior of more general elements related by behavior. See: generalization.
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. See: collaboration.
Interaction Diagram	A generic term that applies to several types of diagrams that emphasize object interactions. These include collaboration diagrams and sequence diagrams.
Interface	A named set of operations that characterize the behavior of an element.
Link	A semantic connection among a tuple of objects. An instance of an association. See: association.
Link End	An instance of an association end. See: association end.
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.
Model	A semantically complete abstraction of a system.
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 things like requirements, observations, reviews, and explanations, in addition to rendering constraints.
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. See: class, instance.
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. See: class diagram, collaboration diagram.
Operations	An operation is the implementation of a service that can be requested from any object of the class to affect behavior.
Package	A package is a general-purpose mechanism for organizing elements into groups. Graphically, a package is rendered as a tabbed folder.
Postcondition	A constraint that must be true at the completion of an operation.
Precondition	A constraint that must be true when an operation is invoked.
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 fuller 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 of relationships include associations and generalizations.
Relationships	There are four kinds of relationships in the UML: Dependency, Association, Generalization, Realization.
Sequence Diagram	A diagram that shows object interactions arranged in 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. See: collaboration diagram.
Signal	The specification of an asynchronous stimulus communicated between instances. Signals may have parameters.
Specification	A declarative description of what something is or does. Contrast: implementation (one of two Annotation mechanisms in UML).
Source	Designates the originating state vertex (state or pseudostate) 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. Contrast: state [OMA].
State Machine	A behavior that specifies the sequences of states that an object or an 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. See: 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 UML. See: constraint, tagged value.
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. See: message.
Swimlane	A partition on an activity diagram for organizing the responsibilities for actions. Swimlanes typically correspond to organizational units in a business model. See: partition.
Tagged Values	Everything in the UML has its own set of properties: classes have names, attributes, and operations, and so on. With stereotypes, you can add new things to the UML; with tagged values, you can add new properties.
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 interesting 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 lightweight process. See process.
Time Event	An event that denotes the time elapsed since the current state was entered. See: event.
Time Expression	An expression that resolves to an absolute or relative value of time.
Trace	A dependency that indicates a 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. See also: implementation class Contrast: interface.
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. See: use case instances.
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. See: use case class.

Use Case Model	A model that describes a system’s functional requirements in terms of use cases.

ANNEX D

CADM KEY ENTITY DEFINITIONS

Source: DoD Data Dictionary System (DDDS).

ACTION	(325/1) (A) AN ACTIVITY.
ACTION-VERB	(11373/1) (A) A FUNCTION TO BE PERFORMED.
ACTIVITY-MODEL-INFORMATION-ELEMENT-ROLE	(4182/2) (A) THE ROLE ASSIGNED TO AN INFORMATION-ELEMENT FOR A PROCESS-ACTIVITY IN A SPECIFIC ACTIVITY-MODEL.
ACTIVITY-MODEL-THREAD	(20160/1) (A) A PATH IN AN ACTIVITY-MODEL CONSISTING OF SEQUENTIAL INFORMATION FLOWS FROM ONE PROCESS-ACTIVITY TO ANOTHER.
AGREEMENT	(332/1) (A) AN ARRANGEMENT BETWEEN PARTIES.
ANTENNA-TYPE	(6542/2) (A) THE CLASSIFICATION OF A DEVICE FOR THE COLLECTION OR RADIATION OF ELECTROMAGNETIC SIGNALS.
ARCHITECTURE	(19524/1) (A) THE STRUCTURE OF COMPONENTS, THEIR RELATIONSHIPS, AND THE PRINCIPLES AND GUIDELINES GOVERNING THEIR DESIGN AND EVOLUTION OVER TIME.
ARCHITECTURE-CHANGE-PROPOSAL-REVIEW	(22443/1) (A) THE CHARACTERIZATION OF A CONFIGURATION MANAGEMENT ACTIVITY FOR CHANGES TO ARCHITECTURE.
ARCHITECTURE-ORGANIZATION	(19546/1) (A) THE RELATION OF AN ARCHITECTURE TO A SPECIFIC ORGANIZATION.
AUTOMATED-INFORMATION-SYSTEM	(8020/1) (A) AN INTEGRATED SET OF COMPONENTS USED TO ELECTRONICALLY MANAGE DATA.
BATTLEFIELD-FUNCTIONAL-AREA-PROPONENT	(19563/1) (A) A DISCRETE AREA OF RESPONSIBILITY READILY IDENTIFIABLE BY FUNCTION PERFORMED WHICH CONTRIBUTES DIRECTLY TO BATTLEFIELD MANAGEMENT.
BUSINESS-SUBFUNCTION	(22594/1) (A) THE LOWER-LEVEL SET OF FUNCTIONS PERFORMED BY THE FEDERAL GOVERNMENT FOR A SPECIFIC LINE-OF-BUSINESS.
CAPABILITY	(333/1) (A) AN ABILITY TO ACHIEVE AN OBJECTIVE.
COMMUNICATION-CIRCUIT	(19575/1) (A) A PATH USED FOR TRANSMITTING DATA.
COMMUNICATION-CIRCUIT-TYPE	(19576/1) (A) A KIND OF PATH USED FOR TRANSMITTING DATA.
COMMUNICATION-LINK-TYPE	(19579/1) (A) A GENERIC KIND OF COMMUNICATION-LINK.
COMMUNICATION-MEANS	(19580/1) (A) A PHYSICAL OR ELECTROMAGNETIC INSTANTIATION OF TELECOMMUNICATIONS.
COMMUNICATION-MEDIUM	(19582/1) (A) A MODE OF DATA TRANSMISSION.
COMMUNICATION-SPACE-USE-CLASS	(19585/1) (A) THE SPECIFICATION OF CATEGORIES OF UTILIZATION OF SPACE FOR TELECOMMUNICATION IN BUILDINGS AND OTHER FACILITIES.

COST-BASIS	(19590/1) (A) THE SPECIFICATION USED TO DETERMINE AN UNDERLYING EXPENSE.
COUNTRY	(39/1) (A) A NATION OF THE WORLD.
DATA-ITEM-TYPE	(19595/1) (A) A KIND OF DATA-ITEM.
DECISION-MILESTONE	(20170/1) (A) A DECISION POINT THAT SEPARATES THE PHASES OF A DIRECTED, FUNDED EFFORT THAT IS DESIGNED TO PROVIDE A NEW OR IMPROVED MATERIAL CAPABILITY IN RESPONSE TO A VALIDATED NEED.
DEFENSE-OCCUPATIONAL-SPECIALTY-CROSS-REFERENCE	(22526/1) ® THE RELATIONSHIP OF THE DEPARTMENT OF DEFENSE OCCUPATIONAL CONVERSIONS TO SERVICE-SPECIFIC OCCUPATIONAL SPECIALTIES.
DEPLOYMENT-LOCATION-TYPE	(19596/1) (A) THE CHARACTERIZATION OF A KIND OF GENERIC PLACE FOR DEPLOYED OPERATIONS.
DOCUMENT	(119/1) (A) RECORDED INFORMATION REGARDLESS OF PHYSICAL FORM.
EVENT	(49/1) (A) A SIGNIFICANT OCCURRENCE.
EVENT-NODE-CROSS-LINK	(19978/1) (A) THE SPECIFICATION OF HOW A SPECIFIC EVENT FOR A SPECIFIC ORIGINATOR NODE TEMPORALLY RELATES TO ANOTHER TERMINATOR NODE SUBJECT TO A CONSTRAINT.
EVENT-TYPE	(12341/1) (A) A CATEGORY OF EVENT.
EXCHANGE-RELATIONSHIP-TYPE	(19608/1) (A) THE SPECIFICATION OF A CLASS OF PAIRING FOR INFORMATION EXCHANGE.
FACILITY	(334/1) (A) REAL PROPERTY, HAVING A SPECIFIED USE, THAT IS BUILT OR MAINTAINED BY PEOPLE.
FACILITY-CLASS	(5742/1) (A) THE HIGHEST LEVEL OF REAL PROPERTY CLASSIFICATION BY THE DEPARTMENT OF DEFENSE.
FACILITY-IMPROVEMENT-ACTIVITY	(19541/1) (A) A PROCESS TO IMPROVE CAPABILITIES FOR A SPECIFIC FACILITY.
FACILITY-TYPE	(50/1) (A) A SPECIFIC KIND OF FACILITY.
FEATURE	(4134/2) (A) A SET OF CHARACTERISTICS, STRUCTURES, OR OTHER ENTITIES THAT ARE OF MILITARY SIGNIFICANCE.
FUNCTIONAL-AREA	(4198/2) (A) A MAJOR AREA OF RELATED ACTIVITY.
FUNCTIONAL-PROCESS-FUNCTION	(22044/1) (A) A GENERAL CLASS OF ACTIVITY IN A SPECIFIC FUNCTIONAL-AREA.
GUIDANCE	(336/4) (A) A STATEMENT OF DIRECTION RECEIVED FROM A HIGHER ECHELON.
HAND-RECEIPT	(21353/1) (A) THE SPECIFICATION OF TRANSFER OF PROPERTY RESPONSIBILITY.
ICON-CATALOG	(19625/1) (A) A DIRECTORY OF IMAGES DEPICTED IN GRAPHICAL PRESENTATION SOFTWARE.
ICON-DATA-CATEGORY	(22294/1) (A) A CLASSIFICATION OF ELEMENTS OF INFORMATION THAT APPLY TO ICONS WITHIN AN ICON-CATALOG.
ICON-DATA-REQUIREMENT	(22295/1) (A) THE SPECIFICATION OF WHETHER AN ASSOCIATED ELEMENT OF INFORMATION IS MANDATORY FOR A SPECIFIC ICON.

IDENTIFICATION-FRIEND-FOE	(17031/1) (A) THE RECOGNIZED HOSTILITY CHARACTERIZATION OF A BATTLEFIELD OBJECT.
IMPLEMENTATION-TIME-FRAME	(19731/1) (A) THE SPECIFICATION OF A GENERAL CHRONOLOGICAL PERIOD FOR THE INSTANTIATION OF A CONCEPT, SYSTEM, OR CAPABILITY.
INFLATION-FACTOR	(19732/1) (A) ADJUSTMENTS TO COSTS THAT DEPEND ON FISCAL YEAR.
INFORMATION-ASSET	(4246/3) (A) AN INFORMATION RESOURCE.
INFORMATION-ELEMENT	(4199/2) (A) A FORMALIZED REPRESENTATION OF DATA SUBJECT TO A FUNCTIONAL PROCESS.
INFORMATION-TECHNOLOGY-REGISTRATION	(20501/1) (A) THE IDENTIFICATION OF A MISSION-CRITICAL/MISSION-ESSENTIAL INFORMATION TECHNOLOGY SYSTEM OR OTHER ASSET.
INFORMATION-TECHNOLOGY-STANDARD-CATEGORY	(20513/1) (A) A CLASSIFICATION OF INFORMATION-TECHNOLOGY-STANDARD.
INTERNAL-DATA-MODEL-TYPE	(9289/2) (A) A CLASSIFICATION OF AN INTERNAL-DATA-MODEL.
INTERNET-ADDRESS	(19762/1) (A) THE SPECIFICATION OF A VALUE OR RANGE OF VALUES CONSTITUTING THE LABEL FOR A NODE ON THE INTERNET.
INTEROPERABILITY-DOCUMENT-TYPE	(22390/1) (A) A KIND OF DOCUMENT THAT FOCUSES ON PROPERTIES WHICH ENABLE SYSTEM INTEROPERATION.
LANGUAGE	(2228/1) (A) A MEANS OF COMMUNICATION BASED ON A FORMALIZED SYSTEM OF SOUNDS AND/OR SYMBOLS.
LINE-OF-BUSINESS	(22593/1) (A) THE TOP-LEVEL SET OF FUNCTIONS PERFORMED BY THE FEDERAL GOVERNMENT.
LOCATION	(343/2) (A) A SPECIFIC PLACE.
MATERIEL	(337/1) (A) AN OBJECT OF INTEREST THAT IS NON-HUMAN, MOBILE, AND PHYSICAL.
MATERIEL-ITEM	(787/1) (A) A CHARACTERIZATION OF A MATERIEL ASSET.
MEASURE-UNIT	(2482/2) (A) THE INCREMENT BY WHICH MATTER IS MEASURED.
MILITARY-PLATFORM	(22100/1) (A) AN OBJECT FROM WHICH OR THROUGH WHICH MILITARY TASKS CAN BE CONDUCTED.
MILITARY-TELECOMMUNICATION-USE	(19773/1) (A) THE CHARACTERIZATION OF SPECIFIC USE-DEPENDENT BUT FACILITY-INDEPENDENT PARAMETERS FOR ESTIMATING THE COMMUNICATIONS, WIRING, AND EQUIPMENT REQUIRED BY MILITARY OCCUPANTS OF FACILITIES.
MILITARY-UNIT-LEVEL	(42/2) (A) A MILITARY-UNIT ACCORDING TO A STRATUM, ECHELON, OR POINT WITHIN THE MILITARY COMMAND HIERARCHY AT WHICH CONTROL OR AUTHORITY IS CONCENTRATED.
MISSION	(1/3) (A) THE TASK, TOGETHER WITH THE PURPOSE, THAT CLEARLY INDICATES THE ACTION TO BE TAKEN.
MISSION-AREA	(2305/1) (A) THE GENERAL CLASS TO WHICH AN OPERATIONAL MISSION BELONGS.

MODELING-AND-SIMULATION-JUSTIFICATION	(19776/1) (A) A STATEMENT PROVIDING RATIONALE TO JUSTIFY REQUIREMENTS FROM THE POINT OF VIEW OF MODELING AND SIMULATION.
NETWORK	(10972/1) (A) THE SPECIFICATION FOR THE JOINING OF TWO OR MORE NODES FOR A SPECIFIC PURPOSE.
NETWORK-CONTROLLER-TYPE	(20591/2) (A) THE KIND OF FUNCTIONAL PROPONENT WHO EXERCISES AUTHORITY OVER A NETWORK.
NETWORK-ECHELON	(22486/1) (A) THE NORMAL OPERATIONAL LEVEL SUPPORTED BY A NETWORK.
NETWORK-TYPE	(11570/1) (A) A SPECIFIC KIND OF NETWORK.
NODE	(956/1) (A) A ZERO DIMENSIONAL TOPOLOGICAL PRIMITIVE THAT DEFINES TOPOLOGICAL RELATIONSHIPS.
NODE-SYSTEM-ASSET-OWNERSHIP	(20009/1) (A) THE POSSESSION, IN WHOLE OR PART, OF THE OBJECTS OF VALUE ASSOCIATED TO A SPECIFIC NODE-SYSTEM.
NODE-SYSTEM-COST-MANAGEMENT	(20011/1) (A) THE AMOUNTS ASSOCIATED WITH VARIOUS ASPECTS OF THE MANAGEMENT OF A NODE-SYSTEM.
OCCUPATION	(2009/1) (A) A FIELD OF WORK.
OPERATIONAL-CONDITION	(19589/1) (A) A VARIABLE OF THE OPERATIONAL ENVIRONMENT OR SITUATION IN WHICH A UNIT, SYSTEM, OR INDIVIDUAL IS EXPECTED TO OPERATE THAT MAY AFFECT PERFORMANCE.
OPERATIONAL-DEPLOYMENT-MISSION-TYPE	(19848/1) (A) THE KIND OF HIGH-LEVEL TASKING FOR DEPLOYED OPERATIONS.
OPERATIONAL-DEPLOYMENT-PHASE	(19849/1) (A) A STAGE OF THE OPERATIONAL ACTIVITIES CONDUCTED FOR DEPLOYED OPERATIONS.
OPERATIONAL-FACILITY-ECHELON	(19853/1) (A) A SUBDIVISION OF A HEADQUARTERS (OR) A SEPARATE LEVEL OF COMMAND AS IT APPLIES TO AN OPERATIONAL-FACILITY.
OPERATIONAL-FACILITY-PROPONENT	(19854/2) (A) THE AGENT RESPONSIBLE FOR REQUIREMENTS DEVELOPMENT OF OPERATIONAL FACILITIES.
OPERATIONAL-MISSION-THREAD	(19857/1) (A) AN IDENTIFIED INFORMATION EXCHANGE SEQUENTIAL PROCEDURE TO SUPPORT TASK EXECUTION BY INFORMATION SYSTEMS AND ORGANIZATION-TYPES.
OPERATIONAL-ROLE	(22459/1) (A) THE SPECIFICATION OF A SET OF ABILITIES REQUIRED FOR PERFORMING ASSIGNED ACTIVITIES AND ACHIEVING AN OBJECTIVE.
OPERATIONAL-SCENARIO	(19860/1) (A) A CONCEPT AND SCRIPT FOR POSSIBLE EVENTS AND ACTIONS FOR MILITARY OPERATIONS.
ORGANIZATION	(345/1) (A) AN ADMINISTRATIVE STRUCTURE WITH A MISSION.
ORGANIZATION-TYPE	(892/2) (A) A CLASS OF ORGANIZATIONS.
PERIOD	(1321/1) (A) INTERVAL OF TIME.
PERSON-TYPE	(897/2) (A) A CLASS OF PERSONS.
POINT-OF-CONTACT	(19867/1) (A) A REFERENCE TO A POSITION, PLACE, OFFICE, OR INDIVIDUAL ROLE IDENTIFIED AS A PRIMARY SOURCE FOR OBTAINING INFORMATION.

POINT-OF-CONTACT-TYPE	(22039/1) (A) A KIND OF POINT-OF-CONTACT.
POSITION	(2112/1) (A) A SET OF ESTABLISHED DUTIES.
PROCESS-ACTIVITY	(4204/3) (A) THE REPRESENTATION OF A MEANS BY WHICH A PROCESS ACTS ON SOME INPUT TO PRODUCE A SPECIFIC OUTPUT.
PROCESS-ACTIVITY-FUNCTIONAL-PROCESS	(22043/1) (A) THE MEANS BY WHICH TO CARRY OUT A HIGH-LEVEL FUNCTION.
PROCESS-STATE-VERTEX	(20025/1) (A) THE ABSTRACTION OF AN OBSERVABLE MODE OF BEHAVIOR.
RECORD-TRACKING	(19871/1) (A) INFORMATION REGARDING A SPECIFIC RECORD IN A TABLE OF DATA.
REGIONAL-COST-FACTOR	(19544/1) (A) THE EXPECTED EXPENSE MODIFICATION FOR A GEOGRAPHIC AREA THAT ACCOUNTS FOR SPECIFIC LOCAL COSTS IN RELATION TO A NATIONAL AVERAGE.
RELATION-TYPE	(6515/2) (A) AN ASSOCIATION BETWEEN OBJECTS THAT DEFINES AN INFORMATION ASSET.
ROOM-TYPE	(5605/1) (A) A KIND OF A ROOM.
SATELLITE	(14361/1) (A) A MAN-MADE BODY WHICH REVOLVES AROUND AN ASTROMETRIC-ELEMENT AND WHICH HAS A MOTION PRIMARILY DETERMINED BY THE FORCE OF ATTRACTION OF THAT ASTROMETRIC-ELEMENT.
SECURITY-ACCESS-COMPARTMENT	(16224/2) (A) THE SPECIFICATION OF AN EXCLUSION DOMAIN FOR INFORMATION RELEASED ON A FORMALLY RESTRICTED BASIS (E.G., TO PROTECT SOURCES OR POTENTIAL USE).
SECURITY-CLASSIFICATION	(940/2) (A) THE LEVEL ASSIGNED TO NATIONAL SECURITY INFORMATION AND MATERIAL THAT DENOTES THE DEGREE OF DAMAGE THAT ITS UNAUTHORIZED DISCLOSURE WOULD CAUSE TO NATIONAL DEFENSE OR FOREIGN RELATIONS OF THE UNITED STATES AND THE DEGREE OF PROTECTION REQUIRED.
SKILL	(2226/1) (A) AN ABILITY.
SOFTWARE-LICENSE	(1856/1) (A) THE STIPULATION(S) (AND LEGAL TERMS) BY WHICH THE SOFTWARE MAY BE USED.
SOFTWARE-SERIES	(18977/1) (A) A SET OF SOFTWARE KNOWN BY A SINGLE NAME, BUT COMPRISED OF ONE OR MORE VERSIONS DEVELOPED OVER TIME.
SYSTEM	(326/1) (A) AN ORGANIZED ASSEMBLY OF INTERACTIVE COMPONENTS AND PROCEDURES FORMING A UNIT.
SYSTEM-DETAIL-NODE-TYPE	(22391/1) (A) A KIND OF REPRESENTATION OR DEPICTION APPLICABLE TO SYSTEMS.
SYSTEM-PROPONENT	(22392/1) (A) AN AGENT RESPONSIBLE FOR RESEARCH, DEVELOPMENT, TEST, OR EVALUATION OF SYSTEMS.
SYSTEM-STATUS-TYPE	(22098/1) (A) THE SPECIFICATION OF A KIND OF DEVELOPMENT OR TRANSITION OF ONE OR MORE SYSTEMS.
SYSTEM-TYPE	(9083/2) (A) A SPECIFIC KIND OF SYSTEM.
SYSTEM-USAGE	(22396/1) (A) THE SPECIFICATION OF EMPLOYMENT FOR WHICH SYSTEMS ARE CREATED.
TASK	(290/2) (A) A DIRECTED ACTIVITY.

TECHNICAL-INTERFACE	(21694/1) (A) A GENERIC CONNECTION BETWEEN TWO ELEMENTS THAT IMPLEMENT INFORMATION TECHNOLOGY IN WHICH INFORMATION IS CAPABLE OF BEING TRANSMITTED FROM THE SOURCE ELEMENT TO THE DESTINATION ELEMENT.
TECHNICAL-INTERFACE-TYPE	(19761/1) (A) A KIND OF GENERIC CONNECTION BETWEEN ELEMENTS THAT IMPLEMENT INFORMATION TECHNOLOGY.
TECHNICAL-SERVICE	(19676/1) (A) A DISTINCT PART OF THE SPECIALIZED FUNCTIONALITY THAT IS PROVIDED A SYSTEM ELEMENT ON ONE SIDE OF AN INTERFACE TO A SYSTEM ELEMENT ON THE OTHER SIDE OF AN INTERFACE.
TECHNICAL-SERVICE-AREA	(19677/2) (A) A FIELD OF SPECIALIZED FUNCTIONALITY, USUALLY SPECIFIED BY A REFERENCE-MODEL TO DEFINE INTERFACES.
TECHNOLOGY	(8936/1) (A) THE APPLICATION OF SCIENCE TO MEET ONE OR MORE OBJECTIVES.
TELEPHONE-ADDRESS	(1938/1) (A) AN ELECTRONIC ADDRESS THAT SUPPORTS COMMUNICATION VIA TELEPHONIC MEDIA.
TRANSITION-PROCESS	(20082/1) (A) THE DESCRIPTION OF A METHOD FOR RELATING A “SOURCE” PROCESS-STATE-VERTEX TO A “TARGET” PROCESS-STATE-VERTEX.
UNIFORMED-SERVICE-ORGANIZATION-COMPONENT-TYPE	(2726/2) (A) A SPECIFIC KIND OF SUBDIVISION OF A UNIFORMED-SERVICE-ORGANIZATION.

Note: 115 entities are listed in this table. Source: DoD CADM Baseline 1.0 (18 June 2003).

ANNEX E
REFERENCES

[All-CADM, 2003a]	All-DoD Core Architecture Data Model (All-CADM) for DoD Architecture Framework Version 1.0, Volume 1, *Overview Description*, Office of the DoD Chief Information Officer, Draft (In Preparation), February 2003, UNCLASSIFIED.
[All-CADM, 2003b]	All-DoD Core Architecture Data Model (All-CADM) for DoD Architecture Framework Version 1.0, Volume 2, *Technical Specification*, Office of the DoD Chief Information Officer, Draft (In Preparation), February 2003, UNCLASSIFIED.
[All-CADM, 2003c]	All-DoD Core Architecture Data Model (All-CADM) for DoD Architecture Framework Version 1.0, Volume 3, *Annexes*, Office of the DoD Chief Information Officer, Draft (In Preparation), February 2003, UNCLASSIFIED.
[ASN(RDA)CHENG, 2002]	Assistant Secretary of the Navy for Research, Development, and Acquisition, Chief Engineer, *Integration and Interoperability Strategy for Capabilities Based Acquisition*, Briefing, March 6, 2002.
[Axelsson, 2002]	Axelsson, J., "Model Based Systems Engineering Using a Continuous-Time Extension of the Unified Modeling Language (UML)," *Systems Engineering*, Vol. 5, No. 3, Fall 2002.
[Bienvenu, 2000]	Bienvenu, M., I. Shin, and A. Levis, "C4ISR Architectures III: An Object-Oriented Approach for Architecture Design," *Systems Engineering*, Vol. 3, No. 4, Fall 2000.
[Booch, 1999]	Booch, G., J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language User Guide*, Addison-Wesley Publishing Company, Reading, Massachusetts, April 1999.
[C4ISR AWG, 1997]	C4ISR Architecture Working Group, *C4ISR Architecture Framework*, Version 2.0, 18 December 1997.
[C4ISR AWG, 1998]	C4ISR Architecture Working Group, *Levels of Information System Interoperability (LISI)*, 30 March 1998.
[C4ISR ITF, 1996]	C4ISR Integration Task Force, *C4ISR Architecture Framework*, Version 1.0, 7 June 1996.
[CJCSI 3170.01C, 2003]	Chairman, Joint Chiefs of Staff, Instruction, CJCSI 3170.01C, *Joint Capabilities Integration and Development System (JCIDS)*, June 24, 2003.
[CJCSM 3170.01, 2003]	Chairman, Joint Chiefs of Staff, Manual, CJCSM 3170.01, *Joint Capabilities Integration and Development System (JCIDS)*, June 24, 2003.
[CJCSM 3500.04C, 2002]	Chairman, Joint Chiefs of Staff, CJCSM 3500.04C, *Universal Joint Task List (UJTL)*, July 1, 2002.
[CJCSI 6212.01B, 2000]	Chairman, Joint Chiefs of Staff, J6, CJCSI 6212.01B, *Interoperability and Supportability of National Security Systems and Information Technology Systems*, 8 May 2000.
[CJCSI 6212.01C, 2003]	Chairman, Joint Chiefs of Staff, J6, CJCSI 6212.01C, *Interoperability and Supportability of National Security Systems and Information Technology Systems*, Draft 2003.

[DEB, 2000]	Defense Electronic Business, *Joint Electronic Commerce Architecture*, 2000.
[DeMarco, 1979]	DeMarco, Tom, *Structured Analysis and Systems Specification*, Prentice-Hall, Englewood Cliffs, New Jersey, 1979.
[Department of the Air Force, 2000]	Department of the Air Force, Air Force Instruction 33-124, Enterprise Information Technology Architectures, 1 May 2000.
[Department of the Navy, undated]	Department of the Navy Chief Information Officer, *Architecture Development Process Model*, Online, Available: www.doncio.navy.mil, undated.
[DISA, 2002]	Defense Information Systems Agency, *Joint Technical Architecture*, Version 4.0, 17 July 2002.
[DISC4, 1998]	Director for Information Systems, Command, Control, Communications and Computers (DSC4), *Army Enterprise Architecture Guidance Document*, Version 01.1, 23 December 1998.
[DoD JP 1-02, 2001]	Department of Defense Joint Publication, JP 1-02, "Dictionary of Military and Associated Terms," *Joint Publication 1-02*, August 2002.
[DoD TRM, 2001]	Department of Defense, *Technical Reference Model*, Version 2.0, 9 April 2001.
[DODD 4630.5, 2002]	Department of Defense Directive, DoDD 4630.5, *Interoperability and Supportability of Information Technology (IT) and National Security Systems (NSS)*, January 11, 2002.
[DoDD 5000.1, 2003]	Department of Defense Directive, DODD 5000.1, *The Defense Acquisition System*, May 12, 2003.
[DoDD 8000.1, 2003]	Department of Defense Directive, DODD 8000.1, *Management of DoD Information Resources and Information Technology*, February 27, 2002; Administrative Reissuance March 20, 2002.
[DoDD 8100.01, 2003]	Department of Defense Directive, DODD 8100.01, *Global Information Grid Overarching Policy*, September 19, 2002.
[DoDI 4630.8, 2002]	Department of Defense Instruction, DODI 4630.8, *Procedures for Interoperability and Supportability of Information Technology (IT) and National Security Systems (NSS)*, May 2, 2002.
[DoDI 5000.2, 2003]	DoD Instruction 5000.2, *Operation of the Defense Acquisition System*, May 12, 2003.
[FIPS 183, 1993]	Integration Definition for Function Modeling (IDEF0), *Federal Information Processing Standards (FIPS) Publication 183*, 21 December 1993.
[FIPS 184, 1993]	Integration Definition for Data Modeling (IDEF1X), *Federal Information Processing Standards (FIPS) Publication 184*, 21 December 1993.
[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.

[IDEF3, 1995]	Information Integration For Concurrent Engineering (IICE) IDEF3, *Process Description Capture Method Report*, KBSI-IICE-90-STR-01-0592-02, Knowledge Based Systems, Incorporated, 1995.
[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.
[ITMRA, 1996]	Information Technology Management Report Act (Clinger-Cohen Act of 1996).
[JS/J8, 2003]	Joint Staff, J8, *Introduction to the Joint Capabilities and Integration Development System*, Briefing, 2003, Online, Available: http://dod5000.dau.mil/.
[Levis, 2000]	Levis, A., and L. Wagenhals, "C4ISR Architectures I: Developing a Process for C4ISR Architecture Design," *Systems Engineering, Vol. 3*, No. 4, Fall 2000.
[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 and 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.
[NATO, 2000]	Allied Data Publication 34, *NATO C3 Technical Architecture, Volume 2: Architectural Descriptions and Models*, Version 4.0, 7 March 2003, pp. 35-38.
[NRO, 2001]	National Reconnaissance Office, *National Reconnaissance Office Architecture Framework (DRAFT)*, Version 0.9, May 2001.
[OASD (C3I), 2000]	Office of the Assistant Secretary of Defense for (C3I), *Global Information Grid (GIG) Architecture (v1.0) DRAFT*, 31 December 2000.
[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: Internet: http://www.omg.org.
[OMG, 2001]	Object Management Group, *Unified Modeling Language Specification,* Version 1.4, Framingham, Massachusetts, Available: http://www.omg.org, September 2001.

[OMG, 2000]	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.
[OMG, 2001]	Object Management Group, *UML 2001, Unified Modeling Language Specification*, Version 1.4, Framingham, Massachusetts, Available: http://www.omg.org, September 2001.
[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.
[USD(A&T), ASD(C3I), J6, 1997]	Under Secretary of Defense (Acquisition & Technology), Assistant Secretary of Defense (Command, Control, Communications, and Intelligence [C3I], Joint Staff/J6 Memorandum, Subject: DoD Architecture Coordination Council, 14 January 1997.
[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, http://www.zifa.com/.