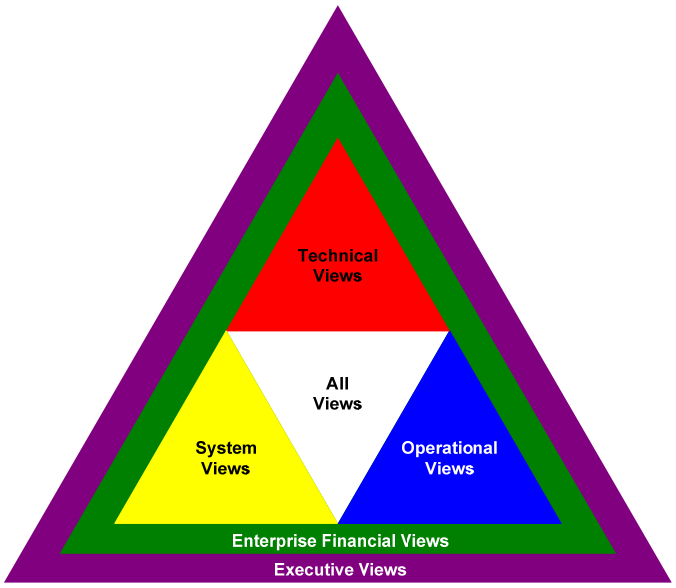
Federal Aviation Administration

National Airspace System Integrated Systems Engineering Framework (NAS ISEF)



Appendix A: Products Development, Integration, and Style Guidance

July 27 , 2012

Version Control

Version	Date	Author	Change Description
3.1	June 30, 2011	SE-2020	Updated initial version to include additional detail on the recommended NAS EA and Requirements products, as well as development and integration guidance.
3.2	July 27, 2012	SE-2020	Incorporates an overview of the recommended development style for NAS EA and Requirements products.

Table of Contents

1 INTRODUCTION AND PURPOSE 1

1.1 Document Structure and Conventions 1
1.2 References 3

2 NAS PROGRAM-LEVEL ARCHITECTURE VIEWS 5

2.1 General Guidance for the Development of Architecture Products 5
2.2 Overview and Summary Information (AV-1) 8
2.3 Integrated Dictionary (AV-2) 9
2.4 High-level Operational Concept Graphic (OV-1) 9
2.5 Operational Node Connectivity Description (OV-2) 11
2.6 Operational Information Exchange Matrix (OV-3) 14
2.7 Operational Activity Model (OV-5) 17
2.8 Operational Event-Trace Description (OV-6c) 21
2.9 Logical Data Model (OV-7) 24
2.10 Systems/Services Interface Description (SV-1) 26
2.11 Systems/Services Communication Description (SV-2) 29
2.12 System/Service Functionality Description (SV-4) 32
2.13 Operational Activity to System/Service Function Traceability Matrix (SV-5) 35
2.14 System/Service Data Exchange Matrix (SV-6) 37
2.15 System/Service Performance Parameter (SV-7) 40
2.16 Systems/Services Event-Trace Description (SV-10c) 42
2.17 Physical Schema (SV-11) 44
2.18 Technical Standards Profile and Forecast (TV-1/2) 46
2.19 Service Roadmap 48
2.20 Infrastructure Roadmap 49

3 NAS PROGRAM-LEVEL REQUIREMENT DOCUMENT 53

3.1 Program Requirement Document (PRD) 53

1 INTRODUCTION AND PURPOSE

The intent of this appendix is to provide Program Offices and the FAA Systems Engineering community with examples and guidance for developing National Airspace System (NAS) Enterprise Architecture (EA) views and Requirement Documents that are meaningful to other architects, requirements analysts, decision makers, and general stakeholders. The products outlined in the following sections provide a starting point for developing Program-level products required throughout the Acquisition Management System (AMS) lifecycle and are based on best practices and industry accepted modeling notations. This guidance documentation is intended to complement, not replace, available and recommended FAA Systems Engineering and Practitioners training courses. The use of consistently applied methodologies, notations, and styles will help the NAS systems engineering community to improve understanding (i.e., reduce interpretation) and to promote integration, standardization, and reuse. This guide also references architecture and requirements authoring tools (IBM Rational System Architect [SA] and IBM Rational Dynamic Object Oriented Repository [DOORS], respectively) and does not infer a "correct" or "incorrect" method for product development, but reflects ANG-B's decision to drive towards uniform architecture and requirement product submissions.

ANG-B will continue its efforts to further refine and implement guidance and communications to increase the standardization of NAS architecture and requirements development and integration. Recommendations for additions or modifications to the content of this guide should be forwarded to the NAS EA and Requirements coordinators for consideration – contact information can be found on the NAS EA Portal.

1.1 Document Structure and Conventions

The remainder of this document is organized as follows:

- Section 2, *Program-level Architecture Products*, provides a description of the Program-level NAS EA products, definitions of the elements contained within each product, as well as product examples, development and integration guidance.
- Section 3, *NAS Requirements*, provides a description of the recommended Program-level NAS requirement document, definitions of the elements contained within the requirement document, as well as a product example, development and integration guidance.

This document uses the following conventions to describe the development and integration guidance:

> *Vertical Integration* - The traceability and alignment of elements at the Program-level, which are either equivalent to or a decomposition of elements at the Enterprise-level. Vertical Integration ensures that the Program satisfies the mission and systems environment dictated for the enterprise (See Figure 1).

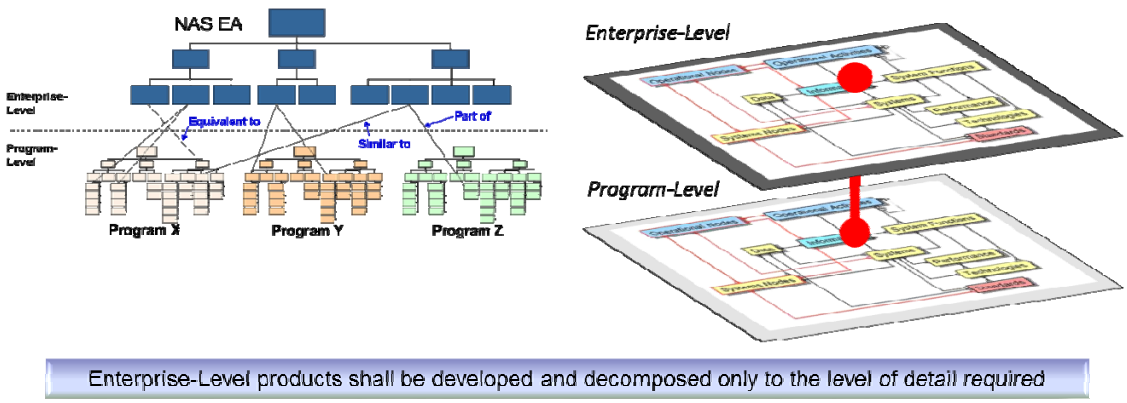


Figure 1: Vertical Integration Concept

Horizontal Integration (Inter) - The traceability and alignment of elements at the Program-level, which are either equivalent or complementary to elements of interrelated Program-level products. Horizontal Integration promotes consistency between Program planning, ensuring that dependencies are accurately depicted.

Horizontal Integration (Intra) - A form of horizontal integration where the elements shared between or referenced by products are consistent across the architecture. Architecture Integration ensures that a holistic picture of the operational and systems environment is depicted (See Figure 2).

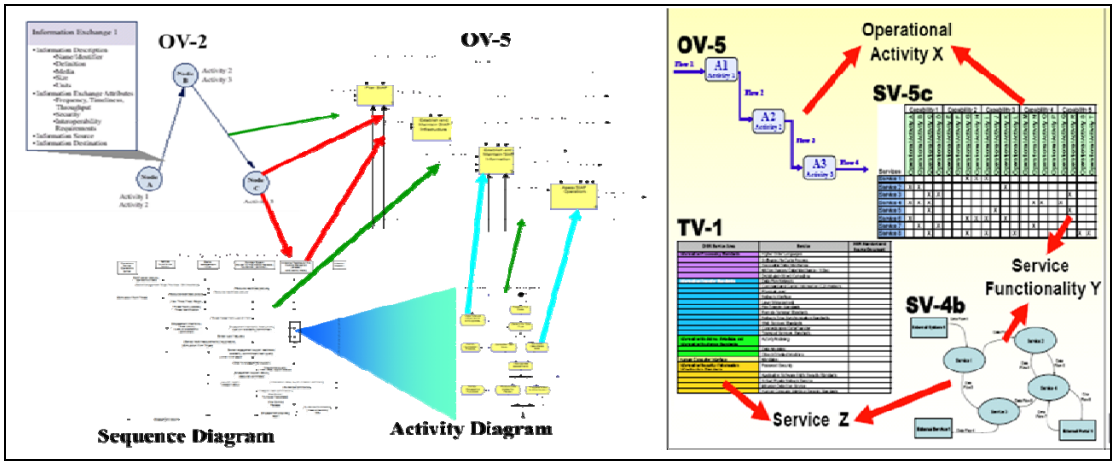


Figure 2: Horizontal Integration (Intra) Concept

The NAS concept of integration (i.e., horizontal and vertical) is implemented through the following common set of principles:

- Vocabularies and taxonomies must be consistent and captured within the NAS EA for visibility, re-use and understandability
- Each architecture element/requirement will be uniquely titled, defined (including required data attributes) and consistently applied

- Interfaces, need lines, information and data exchanges will be loosely coupled, backward compatible, self-describing, and offer a low impact to the enterprise if changed.
- Deployment and use of common IBM Rational System Architect User Properties (including pre-populated templates)
- NAS Enterprise-level EA products and requirements shall be developed and decomposed only to the level of detail required to adequately portray enterprise "To-Be" business capability improvements and transformation priorities, but detailed enough to guide Program-level development and alignment
- Each Program has full authority and responsibility to develop and maintain their portion of the EA/requirements.

These principles are encouraged through the guidance provided throughout the following sections.

1.2 References

This guide lists decisions (products, notations, and recommended styles) for the NAS EA and Requirements community. It supplements other development guides and reference documents, and does not cover step-by-step instructions for modeling architecture products, drafting requirement statements, or administration of either specified tool. It is important that other reference documents are used for lower-level details that may be required. The following documents are referenced throughout the Appendix.

- NAS SE&S Tools Users Guide, v1.0, June 28, 2011 – Describes the procedures for accessing and interacting with the IBM Rational SA and DOORS
- IBM Rational Online References and User Guides (Various Dates based on IBM software releases) – Provides software specific instructions within the IBM Rational tools for performing development tasks.
- NAS Integrated Systems Engineering Framework (ISEF), v3.2, June 2012 - Describe the structure, products and processes that apply to the development of integrated architecture products and requirement documents at the Enterprise- and Program-levels
- FAA Acquisition Management System (AMS) Documentation - Describes the policies and guidance for all aspects of the acquisition lifecycle from the determination of mission needs to the procurement and lifecycle management of products and services that satisfy those needs. In regards to architecture, it specifies the delivery phases for a program development cycle.
- FAA System Engineering Manual (SEM), v3.1, June, 2006 - Describes the proper application of System Engineering elements within the FAA and provides guidance for developing requirements.
- Government Printing Office Style Manual, 2008 – This document is the official guide to the form and style of Federal Government for printed documents.

- NAS SE&S Configuration Management Plan, v3.0, September, 2011 – This document provides information about the process for EA and Requirements approval as well as information on the product naming convention.
- Federal Plain Language Guidelines, Revision 1, May 2011 – This document provides government organizations advice on writing clear communications.
- Guidance for Reading and Understanding the OV-7 – This document describes UML class model diagrams and provides guidance for reading and understanding the OV-7.
- Guidance for Creating and Using Reference Classes in an OV-7 – This document provides Program architects guidance in creating reference classes in the proper manner.
- Federal Information Processing Standards Publication (FIPS Pub) 183 – This standard describes the syntax, semantics, associated rules and techniques for developing structured graphical representations of a system or enterprise using the Integration Definition for Function Modeling (IDEF0) language.

2 NAS PROGRAM-LEVEL ARCHITECTURE VIEWS

The following sections describe the recommended Program-level architecture views identified in the NAS ISEF and additional views not specifically listed in the ISEF. Each section includes a description of the product, the definitions of the elements contained within the product, an example of the product, as well as development and integration guidance including stylistic preferences for developing the product using IBM's Rational System Architect (SA) software. The views may be extended to reflect or relate to other architectural aspects that are not shown on any individual product. Furthermore, the recommended list of views does not preclude the development of other architecture related products that provide additional value (e.g., fit-for-purpose views). The tailoring of the product list or views is at the discretion of the NAS Chief Architect and Program-level architect.

During the architecture development process, all programs should be working with NAS EA Program Coordinators to ensure that the developed architecture is consistent with the precepts outlined in the ISEF. If it is discovered that a deviation from the ISEF is needed, approval will be required from the NAS Chief Architect. For major architectural style deviations required at the Program-level, always ensure that NAS EA coordinators and reviewers are made aware early of the requested change to confirm whether or not the deviation is possible, and to assess the impact. This will ensure that Program-level architecture products required for each development checkpoint are delivered on schedule.

2.1 General Guidance for the Development of Architecture Products

2.1.1 Product Cover Sheets

All architecture products submitted for review must use the approved Program-level architecture cover sheet. A cover sheet template is available on the NAS EA Portal that provides information about what is required to satisfy this requirement. At a high level, the program name, the product name, and the product revision history is required. The cover sheet template can be found at: https://nasea.faa.gov/foryou/archdev/main (Under Tools and Resources).

2.1.2 Product Naming Conventions

Within the SA tool, there is no mechanism that limits how an individual architecture product can be named. While in the explorer mode, the diagram tree displays what diagrams are available in the encyclopedia. In the event that multiple diagrams of the same architecture product type are required, ensure that the names are different enough to distinguish between the diagrams, so as not to confuse a reviewer that will be inspecting the submitted architecture. The recommended product naming convention is as follows:

Format	Format Explanation	Example
Program-level <Program Acronym>-<AMS Phase>-<View>-vm.n-MMDDYY	<Architecture Name> represents a short name for the Enterprise-level encyclopedia, including timeframe	Program-level Example *DATACOMM-IIA-OV-5-v1.0-110111* Refers to the first major version of the DataComm program's architecture OV-5 produced during Initial Investment Analysis and baselined on November 11, 2011
	<Program Acronym> represents the acronym of the program developing the architecture	
	<AMS Phase> represents the phase for which the architecture is being developed (CRD – Concept and Requirements Definition, IIA – Initial Investment Analysis, FIA – Final Investment Analysis)	
	<View> represents the specific architecture view (i.e., OV-2, SV-4, etc.)	
	"m" represents the version number of the document	
	"n" denotes a minor revision	
	"MM" represents the 2-digit month	
	"DD" represents the 2-digit day	
	"YY" represents the 2-digit year	

Figure 3: Architecture Product Naming Convention

Note: All SA encyclopedias will be named prior to access being given to them. This ensures that all encyclopedias follow the approved naming convention as described in the NAS SE&S Configuration Management Plan.

2.1.3 Product Title Blocks

Each individual diagram that is developed in SA must have a title block describing the name of the product and the date it was created. This ensures that the product can be identified as a stand-alone document, even if a cover page has not yet been included with the product for submission. In the event that previous versions of the product are available in the SA database, it will be easily identifiable by its date and revision number. This in effect would reduce the possibility of confusing what the final or most up-to-date product is available. An example of the title can be seen below.

To add a title block while in SA, navigate to the "Draw" function bar along the top of the SA application, and scroll down to select "Doc Block". By selecting this, you will be able to add a title block to the diagram.

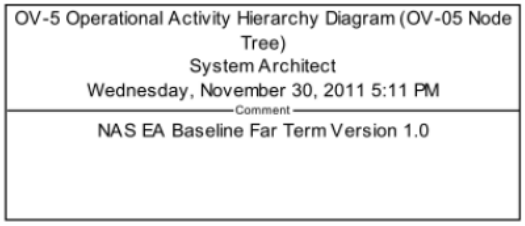


Figure 4: Architecture Product Title Block

2.1.4 Uniform Product Text

Arial 16 point font is used for most objects that will be created in SA for architecture products. In the case of communications between objects, it may be appropriate to use the Arial 10 point font. There are also instances when a diagram with multiple objects requires a smaller font to be used for readability. The individual product sections outlined below specify when a different font is used and provides guidance on when to deviate from the approved font size.

2.1.5 System Architect Style Sheets

At the onset of development, programs will be provided with a pre-populated SA encyclopedia that stores many different objects and associated descriptions. The encyclopedia data will allow programs to create new Program-level architecture products that are more detailed decompositions of the Enterprise-level architecture products. The encyclopedia will include the Enterprise-level USRPROPS file, all enterprise objects and associated definitions, and a base style format created from approved style preferences. More information about the USRPROPS file can be found in the NAS SE&S Tools User Manual. While most of the pre-existing objects will be read-only, a program will be able to reference enterprise objects for alignment.
SA style sheets will be integrated into Program-level encyclopedias prior to any development activities taking place. This action essentially edits the default style of all available SA objects, thereby enforcing the format and style of object symbols.

2.1.6 System Architect Object Definitions

As previously mentioned, all new programs will be provided with a pre-populated SA encyclopedia that stores many different objects and associated descriptions from the Enterprise-level. When developing architecture products, all objects are required to have an associated definition to inform reviewers what the object is and or performs. The definition should be written with the assumption that the reader only has a basic understanding of the content. In addition, abbreviations, acronyms, and/or references that may not be available to all users should be spelled out or avoided.

Recommendations to modify existing definitions at the Enterprise-level should be communicated to the EA coordinator during the product review and working sessions.

2.1.7 General Best Practices and Rules

- Use of Language (Definitions and Descriptions): Use easy-to-understand words and sentences. In general, use the present tense.
- Capitalization: Use sentence-style capitalization for headings. Capitalize and spell element names. To avoid ambiguity, capitalize the first letter of each word in the names of menus, dialog options, commands, fields, and other such elements, regardless of their capitalization on the user interface.
- Acronyms: When using acronyms, ensure that the acronym has a documented full definition available in the final document being provided or in the SA definition.

2.2 Overview and Summary Information (AV-1)

2.2.1 Product Description, Elements, and Example

The Overview and Summary Information (AV-1) is a textual description of the architecture. The objective of the AV-1 is to describe the program's visions, goals, objectives, plans, activities, events, conditions, measures, effects (outcomes), and produced objects. This overview includes assumptions, constraints, and limitations that may affect high-level decisions relating to an architecture-based work program. In the initial phases of architecture development, it serves as a planning guide and later provides a summary of the what, when, why, and how of the plan as well as a navigation aid to the models that have been created.

The AV-1 contains the following elements:

- *Architecture Project Identification*: identifies the architecture name, the architect, and the organization developing the architecture. It also includes assumptions and constraints, identifies the approving authority, and the completion date.
- *Scope: Architecture Views and Products*: identifies the views and products applicable to the development of the architecture.
- *Purpose and Viewpoint*: describes the intent or need of the architecture effort and what the architecture should demonstrate, as well as the perspective in which the product is being developed.
- *Context*: describes the setting in which the program architecture exists, including the mission, relevant goals, objectives, and vision statements, concepts of operation, etc. This section also identifies authoritative sources for the rules, criteria, and conventions followed, as well as the known and/or anticipated linkages to other architectures.
- *Assumptions and Constraints*: describes the assumptions and constraints by which the architecture was developed under and within.
- *Tools and File Formats Used*: identifies the tool suite used to develop the Program-level architecture and related products, and associated file formats.
- *Findings*: presents the findings and recommendations that have been developed based on the architecture effort.

An example/template for the AV-1 can be found online at http://nasea.faa.gov/foryou/archdev/main under the “Tools and Resources” section.

2.2.2 Product Integration

2.2.2.1 Vertical Integration

- Referenced Operational Improvements and Concepts should be consistent with the Enterprise-level operational capabilities/improvements.

2.2.2.2 Horizontal Integration (Inter)

- Interrelated Program-level architectures should be identified to enable integration.

2.2.2.3 Horizontal Integration (Intra)

- Not applicable as this is a summary of the entire Program-level architecture.

2.3 Integrated Dictionary (AV-2)

2.3.1 Product Description, Elements, and Example

The Integrated Dictionary (AV-2) is an architectural data repository with definitions of all terms used throughout the architectural data and presentations. The AV-2 facilitates architectural development, validation, maintenance, and re-use by providing consistency across populated views and across architectural descriptions. It can also help trace architectural data to authoritative sources.

The AV-2 contains the following elements:

- *Term*: name of the architecture element. The full text name of the architecture element consistent with the name entered in SA.
- *Description*: description of the architecture element and the applicable source/reference (authoritative documentation). The definition used to create an AV-2 entry should be the same definition entered into SA.
- *Element Type*: represents a particular classification of architecture entities (i.e., Operational node, System, Service).
- *View*: denotes the architectural view(s) that contain the term.

Term	Definition	Acronym	Type	View(s)
3rd Party Provider	A NextGen commercial partner who provides commercial outlets for NAS information.	3PP	System Node	SV-1, SV-2, SV-6
A/C Weather Receiver	A weather receiver located on an aircraft that receives broadcasted weather information.		System	SV-1, SV-2, SV-6
Aircraft	The Aircraft operational node is a device that is used, or intended to be used, for flight.	A/C	Operational Node	OV-2

Figure 5: Example Integrated Dictionary

2.3.2 Product Integration

2.3.2.1 Vertical Integration

- The element names and descriptions contained in the Program-level AV-2 should not conflict with names and descriptions in the Enterprise-level AV-2.

2.3.2.2 Horizontal Integration (Inter)

- The Program-level Integrated Dictionary contains the same glossary and component architecture relationships between interrelated Program-level architecture.

2.3.2.3 Horizontal Integration (Intra)

- Each labeled item in the Program-level product set should have a corresponding entry in the Integrated Dictionary.

2.4 High-level Operational Concept Graphic (OV-1)

2.4.1 Product Description, Elements, and Example

The High-level Operational Concept Graphic (OV-1) consists of a high-level diagram with accompanying text that describes the main aspects of operations. The content of an OV-1

depends on the scope and intent of the architectural description, but in general describes the business activities or missions, high-level operations, organizations, and geographical distribution of assets. The model frames the operational concept (what happens, who does what, in what order, to accomplish what goal) and highlights interactions to the environment and other external systems. However, the content is at an executive summary-level as other models allow for more detailed definition of interactions and sequencing. The OV-1 conveys, in simple terms using text and pictures, what the architecture is about and how available or planned resources will be employed to support and execute NAS operations. Instructions and template for creating an OV-1 can be found at https://nasea.faa.gov/foryou/archdev/main (under Tools and Resources).

The OV-1 typically contains the following elements:

- *Operational Concept*: Scenario centric view, which may be as abstract as a mission or scenario or as specific as a network or software application.
- *Operational Node*: A logical function or grouping, organization, facility, or human role where information is produced, consumed, or transformed
- *Actor*: A type of Operational Node that performs activities within the architecture
- *System Node*: A physical location or logical grouping of systems, that consumes, produces, or processes information.

Figure 6: Example High-level Concept Graphic

2.4.2 Product Integration

2.4.2.1 Vertical Integration

- The Program-level operational concepts should support the operational concepts identified in the Enterprise-level architecture products.
- The graphics utilized (may include Operational Nodes, Actors, System Nodes) should align to corresponding Enterprise-level elements.

2.4.2.2 Horizontal Integration (Inter)

- The graphics utilized (may include Operational Nodes, Actors, System Nodes) should be consistent across interrelated Program-level architecture products if the same graphics or architecture elements are applicable.

2.4.2.3 Horizontal Integration (Intra)

- Organizations, organization types, and/or human roles depicted in the OV-1 will be traceable to operational nodes in OV-2 and relationships in OV-1 also trace to needlines in OV-2.
- Operational concepts represented in the OV-1 should be decomposed into activities in the OV-5 activity decomposition tree.
- The overarching mission or scenario in the OV-1 is represented by detailed scenarios and processes in the OV-6c.
- Systems and technologies supporting operational improvements (OIs) align to systems and communications in the SV-1 and SV-2, respectively.

2.5 Operational Node Connectivity Description (OV-2)

2.5.1 Product Description, Elements, and Example

The Operational Node Connectivity Description (OV-2) graphically depicts the operational nodes (or organizations) with needlines between those nodes that indicate a need to exchange information. The OV-2 is intended to track the information needs of operational nodes that play a key role in the architecture and what operational nodes are exchanging that information. An OV-2 diagram does not represent a communications link or network, and does not identify the systems (or other means) required to execute the information transfer. Each needline only indicates that there is a relationship and a need for information transfer between the two connected nodes.

The OV-2 contains the following elements:

- *Operational Node:* A logical function or grouping, organization, facility, or human role that produces, consumes, or transforms information. What constitutes an operational node may vary among architectures, including, but not limited to, representing a human role (e.g., Pilot), an organization (e.g., Air Traffic Organization), or a logical or functional grouping (e.g., Air Traffic Control Operations).

- *Needline:* Documents the requirements to exchange information between operational nodes. The needline does not indicate how the transfer is implemented. A needline is represented by a unidirectional arrow indicating the direction of information flow between the operational nodes, and is annotated with a diagram-unique identifier (see §2.5.3 for naming conventions). There is a one-to-many relationship from needlines to information exchanges (e.g., a single needline can represent multiple individual information exchanges). The mapping of the information exchanges to the needlines occurs in the OV-3.

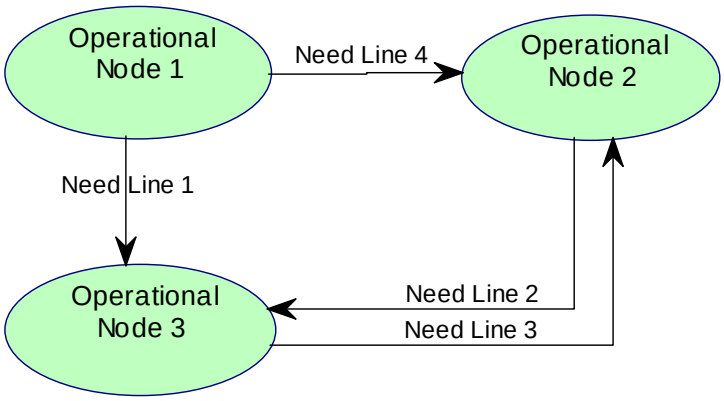


Figure 7: Example Operational Node Connectivity Diagram

2.5.2 Product Integration

2.5.2.1 Vertical Integration

- The operational nodes found in Program-level operational activity models should align to the Enterprise-level OV-2. Depending on the scope of the program, Program-level operational nodes may be a decomposition of an Enterprise-level node to reveal greater detail.

2.5.2.2 Horizontal Integration (Inter)

- The operational nodes and needlines in the Program-level architecture products should integrate with operational nodes and needlines in interrelated Program-level architecture products at the point(s) of interface.

2.5.2.3 Horizontal Integration (Intra)

- The operational nodes in the OV-2 formally describe organizations, facilities, humans, or other functional groupings depicted in the OV-1.
- The needlines in the OV-2 are detailed in the OV-3, which lists out the information exchanges found on each OV-2 needline. A needline in the OV-2 can map to one or more information exchanges in the OV-3.
- The operational activities annotating an operational node in an OV-2 map to operational activities described in the OV-5H. Similarly, the OV-5 should document the operational nodes that participate in each operational activity.
- The operational nodes in the OV-2 correlate to the system/service nodes in the (SV-1) diagram.

2.5.3 Modeling Best Practices and Rules

- A maximum of two needlines can exist between any two operational nodes, one in each direction.
- Operational nodes and needlines should be arranged to minimize overlapping needlines to increase readability of the diagram.
- When naming needlines, use a consistent convention such as "NL_001" where 001 represents a number and the needlines are numbered sequentially as they are created.
- The recommended needline definition format is as follows: "(Sending Op Node) provides (needline data) to (receiving Op Node)".
- Operational views 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. Use of operational nodes in this manner supports analysis and design by separating business process modeling and information requirements from the materiel solutions that support them. (However, operational views often have 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 the operational node. These are often taken as recommendations or boundaries for further system view details).

2.5.4 Product Style Guidance

Object	Object Operator	Representation (shape)	Object Color	Text/ Labeling
Operational Node	NAS Enterprise	Oval	Light Green RGB Values: 204, 255, 204	Arial 16 pt
Operational Node	Internal (FAA) Partners	Oval	Peach RGB Values: 255, 204, 153	Arial 16 pt
Operational Node	Flight Operators	Oval	Light Blue RGB Values: 204, 255, 255	Arial 16 pt
Operational Node	External Partners	Oval	Grey RGB Values: 191, 191, 191	Arial 16 pt
Needline	All	Unidirectional Arrow	Black RGB Values: 0, 0, 0	Arial 16 pt

The object color corresponds to the four operational nodes derived from the NAS Enterprise-level OV-2. Only introduce different colors for operational nodes if a highly complex model is being created, and a different coloring scheme would help to simplify the model's complexity.

- NAS Service Provider: The NAS Service Provider node is responsible for providing air navigation service to consumers, conducting operational analysis, and managing NAS resources to support operations.
- Internal Partners and Consumers: The Internal Partners and Consumers node is responsible for conducting aviation safety inspections and certifications, conducting accident investigations, providing regulations and policies that govern how service providers and consumers operate in the NAS, and managing improvements to the NAS.
- Flight Operators: The Flight Operators node is comprised of individuals, airlines, and organizations responsible for planning and operating a flight within the NAS, including flight crews (on the aircraft or controlling it remotely), FOC personnel, Flight Schedulers, and Flight Planners. This node includes personal, business, commercial aviation, and commercial organizations, as well as government and military organizations.
- External Partners and Consumers: The External Partners and Consumers node provides information and services to support the safety and efficiency of NAS operations. This node also consumes NAS Service Provider services when operating in the NAS. This node includes security and emergency operations, search and rescue, external accident investigation, military operations, airport operations, weather operations, and international air traffic management operations.

2.6 Operational Information Exchange Matrix (OV-3)

2.6.1 Product Description, Elements, and Example

The Operational Information Exchange Matrix (OV-3) Details the information exchanges and identifies who exchanges what information with whom, why the information is necessary, and how the information exchange must occur. The OV-3 identifies information elements and relevant attributes of the information exchanges and associates the exchange to the producing and consuming operational nodes and activities, and to the needline that the flow satisfies. The emphasis of this view is on the logical and operational characteristics of the information. It is important to note that the 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. Rather, this view is intended to capture the most important aspects of selected information exchanges.

The OV-3 contains the following elements:

- *Needline:* Documents the requirements to exchange information between operational nodes. The needline does not indicate how the transfer is implemented. A needline is represented by a unidirectional arrow indicating the direction of information flow between the operational nodes, and is annotated with a diagram-unique identifier (see §2.5.3 for naming conventions). There is a one-to-many relationship from needlines to information exchanges (e.g., a single needline can represent multiple individual information exchanges). The mapping of the information exchanges to the needlines occurs in the OV-3.
- *Information Exchange:* The act of exchanging information between two distinct operational nodes and the characteristics of that act. A needline represents one or more information exchanges.

- *Information Element:* The information content that is required to be exchanged between operational nodes. An information element may be used in one or more information exchanges.
- *Operational Node:* A logical function or grouping, organization, facility, or human role that produces, consumes, or transforms information. What constitutes an operational node may vary among architectures, including, but not limited to, representing a human role (e.g., Pilot), an organization (e.g., Air Traffic Organization), or a logical or functional grouping (e.g., Air Traffic Control Operations).
- *Operational Activity:* An action performed in conducting the business of an enterprise that either produces or consumes an information exchange. It is used to portray operational actions not hardware/software system functions.

The OV-3 Information Exchange Matrix also contains the following attributes:

<u>Transaction Description</u>

- *Transaction Type (Transaction Description Category):* Descriptive field that summarizes the intended method of transmission for a specific information exchange.
- *Triggering Event:* Brief textual description of events(s) that triggers the need for the information exchange.
- *Interoperability Level:* The level achieved or achievable through the exchange (Levels 0-4).
    - Level 0: Isolated (Manual)
    - Level 1: Connected (Peer-to-Peer)
    - Level 2: Functional (Distributed)
    - Level 3: Domain (Integrated)
    - Level 4: Enterprise (Universal)
- *Criticality:* The criticality assessment of the information being exchanged in relationship to the mission being performed.
    - Category 1: Mission critical – critical and high-level information
    - Category 2: Mission operations – required in support of operations
    - Category 3: Core functions – ongoing information exchanges
    - Category 4: Mission critical – unspecified
    - Category 5: Mission support – logistics, transportation, medical, etc.
    - Category 6: Administrative – personnel, pay, training, etc.

<u>Performance Attributes</u>

- *Periodicity:* Frequency of information exchange transmission – may be expressed in terms of worst case or average frequency.
- *Timeliness:* Required maximum time from operational node to operational node expressed in seconds.

<u>Information Assurance</u>

- *Access Control:* The class of mechanisms used to ensure only those authorized can access a specific information element.
- *Availability:* The relative level of effort required to be expended to ensure that the information element can be accessed. Represents how widely a specific information exchange has been implemented.

- *Confidentiality:* The kind of protection required for information element to prevent unintended disclosure.
- *Dissemination Control:* The kind of restrictions on receivers of information element based on sensitivity of information element.
- *Integrity:* The kind of requirements for checks that the content of the information element has not been altered.

Security

- *Accountability:* The organization responsible for the security of the information exchange.
- *Protection:* The code that represents how long the information element must be safeguarded.
- *Classification:* Classification of the information.
- *Classification Caveat:* A set of restrictions on information exchange of a specific classification. Supplements a security classification with information exchange on access, dissemination, and other types of restrictions.

Needline ID	Information Exchange ID	Information Element Name	Sending Operational Node Name	Sending Operational Activity	Receiving Operational Node Name	Receiving Operational Activity
NL_001	IER_001	Flight Procedures	Traffic Management Coordinator	A3.1.1 Design Airspace	Air Traffic Controller	A1.3.3 Assign Procedure

Figure 8: Example Operational Information Exchange Matrix

Each information exchange in the figure above is associated with the needline it helps satisfy. There may be many individual exchanges that collectively satisfy a needline. Note also that each information element exchanged is related to the leaf operational activity (from 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 operational activities in the OV-5.

2.6.2 Product Integration

2.6.2.1 Vertical Integration

- Information exchanges detailed in the Enterprise-level OV-3 correspond to the information being produced and consumed by operational activities in the Program-level OV-5. The information exchanges (captured as inputs, outputs, and controls in the OV-5) may be decomposed at the Program-level depending on the level of detail appropriate for the Program-level operational activities. This vertical alignment ensures program activity models are capturing the correct scope of operational information exchange.
- Information exchanges detailed in the Enterprise-level OV-3 inform the development of Program-level data flow diagrams (SV-4) by providing the high-level information exchange requirements to which data exchanges at the Program-level may vertically align.

2.6.2.2 Horizontal Integration (Inter)

- The Information exchanges in the Program-level architecture products should be consistent with Information Exchanges in interrelated Program-level architecture products at the point(s) of interface.

2.6.2.3 Horizontal Integration (Intra)

- One or more information exchanges in the OV-3 map to a needline in the OV-2.
- Information exchanges detailed in the OV-3 correspond to one or more information flows in the OV-5, or none, if the information flow does not cross node boundaries.
- Information exchanges and their characteristics relate to information types in the OV-7.
- Information exchanges correlate to one or more data exchanges between systems and services detailed in the SV-6, if any part of the information element originates from or flows to an operational activity that is to be automated.

2.6.3 Modeling Best Practices and Rules

- Include an identifier for each information exchange to aid navigation of the OV-3 matrix.
- Information inputs and outputs between operational activities performed at the same operational node (i.e., not associated with a needline on the OV-2), will not show in the OV-3.

2.6.4 Product Style Guidance

- Not applicable, as this product is typically tool generated.

2.7 Operational Activity Model (OV-5)

2.7.1 Product Description, Elements, and Example

The Operational Activity Model (OV-5) describes the operations or tasks that are required to support a defined mission within an organization and the input and output flows between those activities. An important feature of the OV-5 is that it gradually introduces greater levels of detail through the incremental decomposition of higher-level activities. The OV-5 can be used to uncover unnecessary operational activity redundancy, make decisions about streamlining, combining or omitting activities, define or flag issues, opportunities or operational activities and their interactions that need to be scrutinized further. A complete OV-5 consists of an activity hierarchy diagram and activity models built using the Integrated Definition for Functional Modeling (IDEF0) notation. Refer to FIPS Publication 183 for more information on IDEF0.

The OV-5 contains the following elements:

- *Operational Activity*: An action performed in conducting business that either generates or consumes the information exchange. It is used to portray operational actions and not hardware/software system functions.
- *Input, Control, Output, or Mechanism (ICOM) Arrow*:
    - Input: Information or resource flow that is transformed by the operational activity to produce outputs. Input arrows are associated with the left side of an operational activity.

- Control: Conditions required by the operational activity to produce the correct output (e.g., specifications, standards, laws, budgets). Control arrows are associated with the top side of an operational activity.
- Output: Information or objects produced by the operational activity. Output arrows are associated with the right side of an operational activity.
- Mechanism: The means used to perform the operational activity (e.g., people, facilities, equipment). Mechanism arrows are associated with the bottom side of the operational activity.

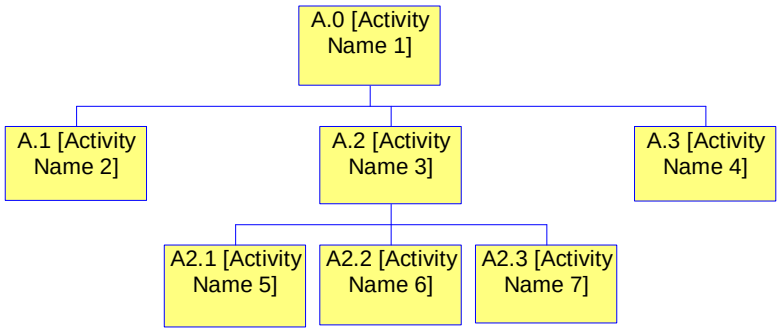


Figure 9: Operational Activity Hierarchy

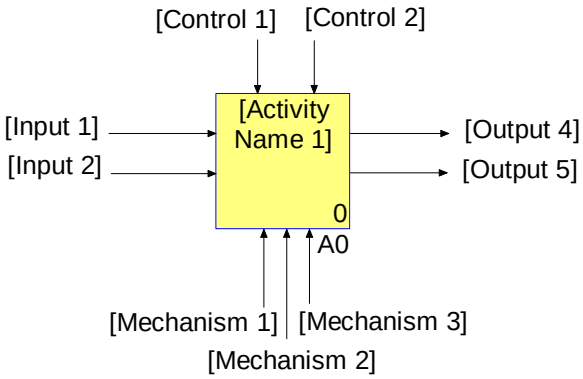


Figure 10: IDEF0 Context Diagram

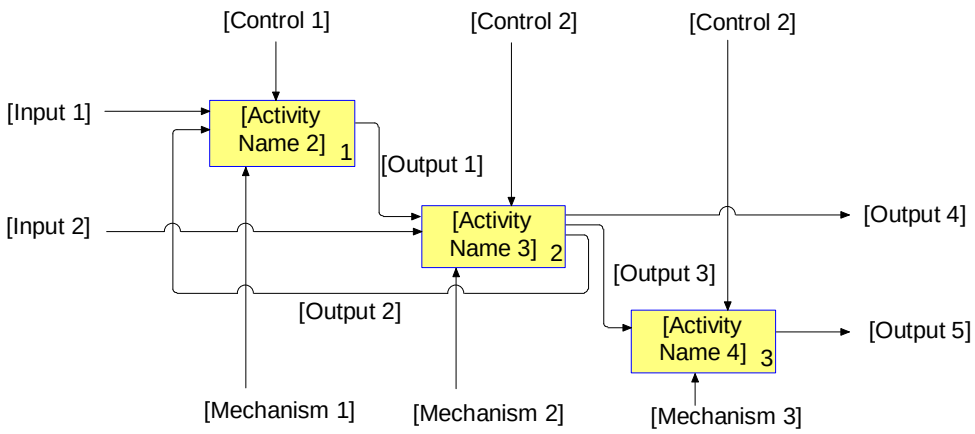


Figure 11: IDEF0 Diagram

2.7.2 Product Integration

2.7.2.1 Vertical Integration

- Provide the appropriate connection points (e.g. operational activities, IERs) to align Program-level activity models with Enterprise-level activity models.
- Aligning Program-level actors and operational nodes with elements in the Enterprise - level OV-2.

2.7.2.2 Horizontal Integration (Inter)

- The operational activities performed by actors at operational nodes are consistent across interrelated Program-level architecture products if the same activities are applicable.
- Inputs and outputs should correspond to interrelated Program-level inputs and outputs.

2.7.2.3 Horizontal Integration (Intra)

- The operational activities annotating an operational node in an OV-2 map to operational activities described in the OV-5H. Similarly, the OV-5 should document the operational nodes that participate in each operational activity.
- Information exchanges detailed in the OV-3 correspond to one or more information flows in the OV-5, or none, if the information flow does not cross node boundaries.
- Each leaf level activity in the OV-5 becomes an activity in the OV-6c.
- Each input and output may be mapped to one or more data entities (or data classes) in the OV-7 data model.
- Each leaf level activity in the OV-5 matches the operational activities in the SV-5 matrix.
- Each activity in the OV-5 should be mapped to one or more NAS operational requirements.

2.7.3 Modeling Best Practices and Rules

- The OV-5 should represent unique Operational activities describing what is performed, not how the activity is performed. The decomposition of a parent activity into child activities is

intended to provide additional clarity while staying within the scope of the parent. Furthermore, if there are multiple instances of a mechanism or more than one mechanism for the same activity and those mechanisms or instances need to communicate/coordinate as part of the accomplishment of that activity, then the activity needs to be decomposed.

- Operational activities should be decomposed to the level necessary to delineate the components performed by a system/service vice human.
- Ensure that all activities are correctly captured and defined, and that all of the boxes are the same size across all IDEF0 diagrams in the encyclopedia.
- Activity titles should be created using verb-noun descriptions. This would be an action verb followed by a noun or noun phrase.
- Information exchanges/output ICOM arrows should have one and only one source operational activity.
- An operational activity can send an information exchange/output to multiple destination activities; however, if the content/attributes are different (and important enough to capture at the Enterprise-level) a separate, uniquely titled/defined information exchange/output should be created.
- When creating IDEF0 ICOM arrows, it is best to ensure that the longest input enters at the top of the activity. The shortest input line should be at the bottom of all inputs.
- When looping ICOM arrows across multiple controls or mechanisms, the developer should attempt to loop the shortest output arrow at the bottom of outputs back first, with each following arrow slightly longer in length looping back, until the final arrow is completed.
- Overlapping lines best practice: Always make an effort to ensure that overlapping ICOM arrows are evenly spaced.
- When creating the OV-5 hierarchy diagram, ensure that there are at least two children activities if a parent activity will be decomposed. Traditionally, more than five child activities indicate an activity can be further decomposed at the parent level.
- Recommended operational activity definition format: *"The (Activity Name) activity (describe what the activity does, not how). This activity begins with (inputs). This activity requires/adheres to/ is conducted within/is constrained by (controls). This activity uses (mechanisms). This activity produces (outputs)."*

2.7.4 Product Style Guidance

Object	SA Representation (shape)	Object Color	Text/ Labeling
Operational Activity	Standard square	Yellow (SA Default) RGB Values: 255, 255, 102	Arial 16 pt
Input	Arrow (one direction)	Black RGB Values: 0, 0, 0	Arial 12 pt

Object	SA Representation (shape)	Object Color	Text/ Labeling
Control	Arrow (one direction)	Black RGB Values: 0, 0, 0	Arial 12 pt
Output	Arrow (one direction)	Black RGB Values: 0, 0, 0	Arial 12 pt
Mechanism	Arrow (one direction)	Black RGB Values: 0, 0, 0	Arial 12 pt

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

2.8.1 Product Description, Elements, and Example

The Operational Event-Trace Description (OV-6c) provides a time-ordered examination of the information exchanges between participating operational nodes as a result of conducting operational activities within a particular scenario. Each event-trace diagram should have an accompanying description that defines the particular scenario or situation. The OV-6c is valuable for moving to the next level of detail from the initial operational concepts. The product helps further define the node interactions from the Operational Node Connectivity Description (OV-2) and operational activities from the Operational Activity Model (OV-5). An OV-6c 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. Two recommended methods are Business Progress Modeling Notation (BPMN) and a Unified Modeling Language (UML) Sequence Diagram.

The OV-6c contains the following elements:

- *Pool*: Represents a grouping of operational entities involved in a process or set of activities.
- *Lane*: A sub-partition within a pool, used to organize and categorize activities within a pool; typically an operational node or actor.
- *Event*: A triggering event that initiates activities.
- *Task*: Operational activity (tasks or processes that transforms information).
- *Sequence Flow*: An arrow used to show the order that activities will be performed.
- *Message Flow*: An arrow used to show the flow of messages between two entities that are prepared to send and receive them.
- *Gateway*: A controlling mechanism that dictates the flow of activities and events.
- *Data Object*: A mechanism to show what data is required or produced by activities.
- *Associations*: Arrow used to associate information consumed or produced by activities with data objects.

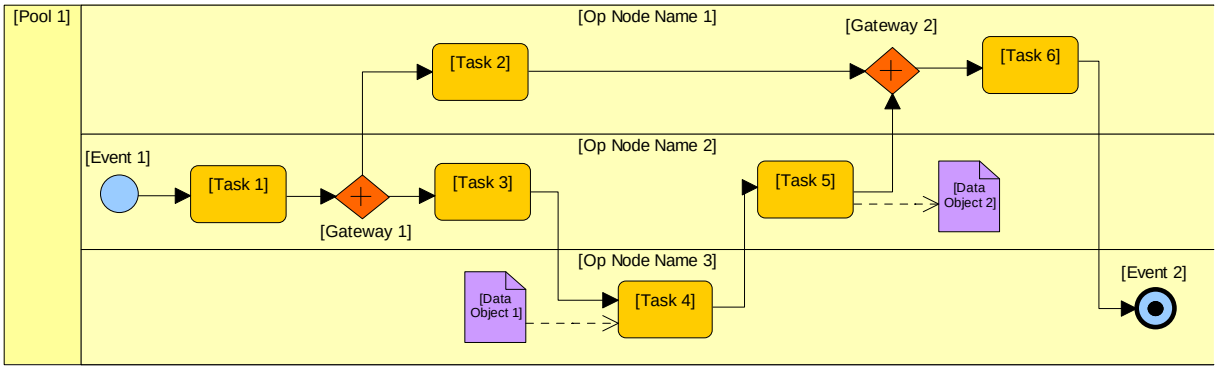


Figure 12: Example Business Process Model

2.8.2 Product Integration

2.8.2.1 Vertical Integration

- Tasks (operational activities) performed are subcomponents of the Enterprise-level operational activities.
- Pools and lanes (actors and operational nodes) are consistent with those in the Enterprise-level OV-2 and OV-6c.

2.8.2.2 Horizontal Integration

- Operational activities or events should be consistent across interrelated Program-level architecture products if the same activities are applicable.
- Operational activity sequencing should be consistent with the sequencing in interrelated Program-level architecture products if the same scenarios or processes are utilized.
- Elements in the process or scenario should be consistent with elements in interrelated Program-level architecture products (Data Objects, Tasks, Pools/Lanes).

2.8.2.3 Horizontal Integration (Intra)

- Tasks (operational activities) must be consistent with the activities depicted in the OV-5.
- Pools and lanes (actors and operational nodes) must be consistent with the operational nodes depicted in the OV-2.
- Data objects must be consistent with the information exchanges depicted in the OV-3.

2.8.3 Modeling Best Practices and Rules

- The recommended notation for this product is BPMN.
- Model the processes from left to right, and top to bottom. This means that the pool and swimlanes are first added to a diagram with associated activities and gateways being added sequentially from left to right across the swimlanes.
- Align objects in a grid style format. This entails creating events, tasks, and gateways in an aligned configuration across the swimlanes and pool. The reader should notice uniform object placement throughout the process.

- Gateway connections do not have to depict outputs only exporting out of the right corner of the gateway diamond in scenarios when a gateway has multiple outputs. However, for gateways that only have one output, the sequence flow line should be drawn from the right corner.
- The OV-6c must not have decision point to decision point gateways. This is always considered an inaccurate notation.
- When modeling, there should always be an activity prior to a gateway.
- A gateway is only considered a decision router. The actual work should always be done in the activity following the gateway.
- Processes may require additional decomposition into sub-processes if the diagram becomes too complex.

2.8.4 Product Style Guidance

Object	Representation (shape)	Object Color	Text/ Labeling
Pool	Pool	Grey RGB Values: 242, 242, 242	Arial 16 pt
Lane	Swimlane	Grey RGB Values: 242, 242, 242	Arial 16 pt
Event	Single Circle: Start Double Circle: Intermediate Bold Circle: End	Start: Green RGB Values: 146, 208, 80 Intermediate: Blue RGB Values: 142, 180, 227 End: Red RGB Values: 255, 0, 0	Arial 14 pt
Task	Rounded rectangle	Gold RGB Values: 255, 204, 0	
Sequence Flow	Solid Straight Line; Round Corners; 0.05	Black RGB Values: 0, 0, 0	Arial 14 pt
Message Flow	Arrow (Dashed line)	Black RGB Values: 0, 0, 0	Arial 14 pt
Gateway	Black fill with white outline (Use default style for different gateway types in SA)	Black RGB Values: 0, 0, 0	Arial 14 pt
Data Object	Rectangle	Purple RGB Values: 179, 162, 199	Arial 14 pt
Associations	Dotted Straight Orthogonal Line	Black RGB Values: 0, 0, 0	Arial 14 pt
Procedure (Sub-process)	Rounded rectangle (+/- sign within rectangle to represent additional processes.)	Gold RGB Values: 255, 204, 0	Arial 14 pt

2.9 Logical Data Model (OV-7)

2.9.1 Product Description, Elements, and Example

The OV-7 describes the structure of the architecture domain's system data and provides a definition of data types, their attributes or characteristics, and their interrelationships. The OV-7 allows analysis of an architecture's data definition aspect, without consideration of implementation-specific or product-specific issues.

The OV-7 contains the following elements:

- *Attributes*: Attributes are properties or characteristics of the Class that describe or portray information about the class' Instances. Attributes are the containers for data values and apply to all Instances of the Class.
- *Class*: Classes describe real world entities (e.g., people, places, things, events, concepts, etc.) and the fundamental information we need to know to support business functions. A class may represent things that are concrete and tangible, or abstract and conceptual, and whose Instances may change over time. Entities are described using the "Class" structure in a UML class diagram.
- *Class Association*: Associations represent relationships or connections between two Classes and may carry a special set of Attributes. Each Association is described by the cardinality (aka multiplicity) that depicts the number of instances of Class A that may/must be related to an instance of Class B. The inverse is also depicted; how many instances of Class B may/must exist for each instance of Class A. Additionally, an Association also contains a text phrase that allows the modeler to describe the nature/purpose of the relationship.
- *Class Instance*: An occurrence or member of the set of items that are identified as constituents in a Class. Instances are not depicted on the model diagram, but are often used as a means for understanding the nature of the Class. For example, if the Class is Airport, then two instances would be Chicago O'Hare International Airport and Ronald Reagan Washington National Airport.
- *Inheritance*: The mechanism by which more specific elements incorporate structure and behavior defined by more general elements.

Examples of an OV-7, and guidance for reading and understanding the OV-7 and UML Class Model diagrams can be found online at http://nasea.faa.gov/foryou/archdev/main under the "Tools and Resources" section.

2.9.2 Product Integration

2.9.2.1 Vertical Integration

- Classes and associations should be consistent with those represented in the Enterprise-level OV-7.

2.9.2.2 Horizontal Integration (Inter)

- Classes and associations should be consistently used across interrelated Program-level architecture products.

2.9.2.3 Horizontal Integration (Intra)

- Information Exchanges should be constructed of entities of the OV-7.
- The OV-7 should provide the logical structure necessary to construct the physical schema described in the SV-11.
- Technical standards identified in the TV-1/2 apply to modeling techniques in the OV-7.

2.9.3 Modeling Best Practices and Rules

- Use the UML Class Diagram.
- Use a minimum set of class modeling constructs to enhance understandability by non-technical personnel.
- Ensure that all classes are correctly captured, named, and defined. The definition should describe what the class is, not how it is manipulated or processed. Class titles should be created using nouns with any necessary modifiers. The titles should be singular and not plural. The title should be presented in all capital letters.
- Ensure that all associations are captured and named, and that all associations have the correct multiplicities assigned. The association name should point from the parent class to the child. Associations should be drawn with the "Arrange Line(s) Orthogonally" property set to yes. Minimize crossing lines.
- Ensure that all class attributes are correctly captured, named, and defined. The definition should describe what the attribute is, not how it is manipulated or processed. The attribute title should begin with the class name and end with an appropriate class domain term (e.g., Description, Name, Code, Text, Date, etc.). The title should be presented in initial capital letter for each term in the title.
- A class attribute should represent only one data concept. Do not aggregate multiple concepts into one class attribute.
- Each class attribute should have its data type set to the appropriate value. If the model implies a specific implementation target (e.g., Oracle or Sybase database, XML schema, etc.), then ensure that the data types are appropriate for the selected implementation target. This is set at the diagram level and will 'flow through' to each class attribute.
- Each class attribute that has an enumerated list (i.e., set of valid values) should use an associated reference table to explicitly state the valid values. For assistance in creating reference classes in the proper manner, please see the associated document titled "Guidance for Creating and Using Reference Classes in an OV-7" found online at http://nasea.faa.gov/foryou/archdev/main under the "Tools and Resources" section.
- When a measure-type attribute is needed to characterize a class, the modeler should create the measure attribute (e.g., Aircraft Maximum Speed, Runway Length) and an associated Unit of Measure attribute (e.g., Aircraft Maximum Speed Unit of Measure, Runway Length

Unit of Measure). The data type for the Unit of Measure attribute should be one of the approved set of Unit of Measure classes in the Enterprise OV-7 (e.g., UOM SPEED, UOM DISTANCE).

2.9.4 Product Style Guidance

Object	SA Representation (shape)	Object Color	Text/ Labeling
Data Class	Standard rectangle	Dependent on the native UML Package that contains the class. **Domain** / **Red** / **Green** / **Blue** Air Transport Infrastructure 128 255 255 Aircraft 128 255 128 ATM Operations 128 128 255 Flight Data 255 128 64 NAS Base Infrastructure 255 255 128 References 255 204 153 Surveillance 192 192 192 Weather 255 128 255	Arial 16 pt
Association	Line	Black RGB Values: 0, 0, 0	Arial 12 pt
Generalization	Line	Black RGB Values: 0, 0, 0	Arial 12 pt
Aggregation	Line	Black RGB Values: 0, 0, 0	Arial 12 pt

2.10 Systems/Services Interface Description (SV-1)

2.10.1 Product Description, Elements, and Example

The Systems/Services Interface Description (SV-1) links the operational and systems architecture models by depicting how resources are structured and interact to realize the logical architecture specified in the OV-2. In specific, the SV-1 is used to describe the interactions between resources (systems, organizations, software) in the architecture and to describe a solution, or solution option, for components of capability and their physical integration on platforms and other facilities. Documenting NAS systems and their corresponding interactions enables stakeholders to gauge the impact of system acquisition, system retention and system termination on other systems as well as the enterprise as a whole.

An SV-1 may represent the realization of a requirement specified in an OV-2 (i.e., in a “Far-Term” architecture), and so there may be many alternative SV models that could realize the operational requirement. Alternatively, in an “As-Is” architecture, the OV-2 may simply be a simplified, logical representation of the SV-1 to allow communication of key resources flows to non-technical stakeholders.

The SV-1 depicts the interaction of system/service nodes and systems/services. It also depicts the interfaces between the identified systems and services. These interfaces are a simplified, abstract representation of one or more communications paths between system/service nodes or

systems/services. The SV-1 does not display the means of communication or how the connection is physically implemented; details of the communications infrastructure (e.g., physical links, communications networks, satellites, etc.) are documented in the Systems/Services Communications Description (SV-2).

The SV-1 contains the following elements:

- *System/Service Node*: A node with the identification and allocation of resources (e.g., platforms, units, facilities, and locations) required to implement specific roles and missions.
- *System/Service*: Any organized assembly of resources and procedures united and regulated by interaction or interdependence to accomplish a set of specific functions.
- *Service Family*: A grouping of independent services that can be arranged or interconnected in various ways to provide different capabilities.
- *Application Service*: A concrete supporting service that is typically identified and defined by application developers, is specific to the application scope they are defined under, and is generally used to perform fine-grained application-specific functions such as data collection, validation, and transfer.
- *Infrastructure Service*: A concrete service that supports non-business related functions of the enterprise and is generally shared and used by Enterprise Services (and sometimes Application Services).
- *System Interface*: An abstract representation of one or more communication paths between system nodes or between systems.
- *Service Interface*: An abstract representation of one or more communication paths between service nodes or between services.

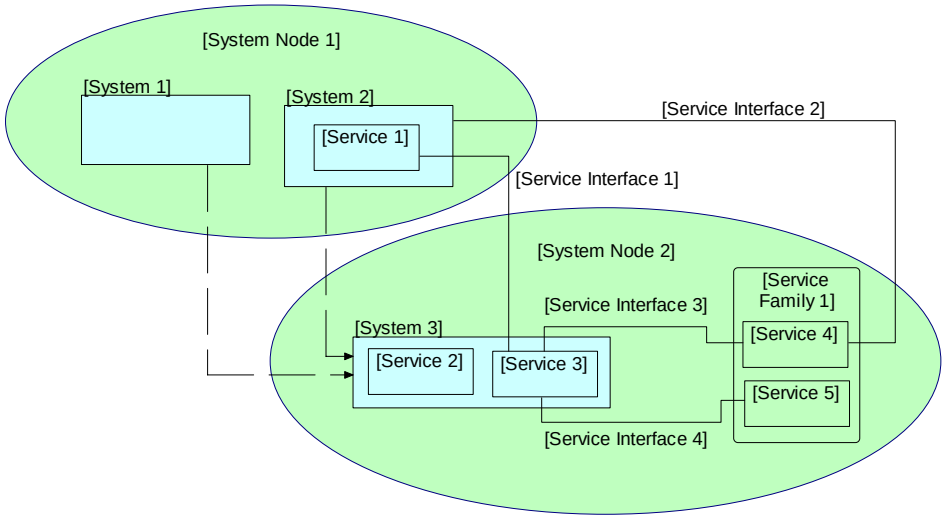


Figure 13: Example Systems/Services Interface Description

2.10.2 Product Integration

2.10.2.1 Vertical Integration

- Program-level system/service nodes should be consistent with or a decomposition of system/service nodes in the Enterprise-level SV-1.
- Systems/services in the Program-level SV-1 also show further detail/decomposition of the Enterprise-level SV-1, including system/service components, software, etc.
- The systems/services utilized in the Program-level architecture products should be consistent with the systems/services identified on the NAS EA Infrastructure Roadmaps.

2.10.2.2 Horizontal Integration (Inter)

- The systems and services utilized in the Program-level architecture products must be consistent across interrelated Program-level architecture products and have a common interface reference.

2.10.2.3 Horizontal Integration (Intra)

- Operational nodes in the OV-2 may be supported by one or more system/service nodes in the SV-1 (indicating that the operational node is responsible for the system/service). A needline in the OV-2 may map to one or more interfaces in an SV-1, and an interface in the SV-1 maps to one or more needlines in the OV-2.
- An interface in the SV-1 is implemented by communications link(s) or communications network(s) in the SV-2.
- Systems/services defined in the SV-1 are executed by system/service functions defined in the SV-4.
- Systems/services in the SV-1 match systems/services in the SV-5.
- Each system/service data element appearing in a data exchange is graphically depicted by one of the interfaces in the SV-1; an interface supports one or more data exchanges.
- Technical standards in the TV-1 apply to and sometimes constrain systems, subsystems, and system hardware/software items in SV-1.
- Timed standard forecasts in TV-2 impact systems, subsystems, and system hardware/software items in SV-1.
- The system interfaces should correspond to the requirements specified in Section 5.1 (Interface Requirements) of the program requirement document.

2.10.3 Modeling Best Practices and Rules

- Only systems, services, subsystems, or hardware/software items and their associated standards are documented in the SV-1, where applicable.
- Keep all systems/service elements the same size if possible.
- Do not use bi-directional arrows.
- System/service nodes and interfaces should be arranged to minimize overlapping lines to increase readability of the diagram.
- Several versions of an SV-1 may be developed (i.e., fit-for-purpose) to highlight different perspectives of the system/service interfaces. For example an intermodal version describes system/service nodes and the interfaces between them or the systems/services resident at the

system/service node. An intra-system version describes subsystems of a single system and the interfaces among them.

- When naming interfaces, use a consistent convention such as “SI_001”

2.10.4 Product Style Guidance

Object	Representation (shape)	Object Color	Text/ Labeling
System Node	Rectangle	Light Green RGB Values: 204, 255, 204	Arial 16 pt
System	Rectangle	Light blue RGB Values: 219, 238, 244	Arial 16 pt
Service Family	Round Corner Square	White RGB Values: 255, 255, 255	Arial 16 pt
Application Service	Rectangle	Yellow RGB Values: 255, 255, 102	Arial 16 pt
Infrastructure Service	Rectangle	Purple RGB Values: 179, 162, 199	Arial 16 pt
System Interface	Arrow (one direction)	Black RGB Values: 0, 0, 0	Arial 10 pt
Service Interface	Arrow (one direction)	Black RGB Values: 0, 0, 0	Arial 10 pt

Note: Only introduce different colors for system nodes if a highly complex model is being created, and a different coloring scheme would help to simplify the model’s complexity.

2.11 Systems/Services Communication Description (SV-2)

2.11.1 Product Description, Elements, and Example

The Systems/Services Communication Description (SV-2) depicts the communication infrastructure that support NAS systems/services and the means for information to be communicated throughout the network. It depicts the communication infrastructure that supports systems/services and implements their interfaces. The SV-2 contains the following elements:

- *System/Service Node*: A node with the identification and allocation of resources (e.g., platforms, units, facilities, and locations) required to implement specific roles and missions.

- *System/Service*: Any organized assembly of resources and procedures united and regulated by interaction or interdependence to accomplish a set of specific functions.
- *Communication System*: System whose primary function is to control the transfer and movement of system/service data as opposed to performing the application processing.
- *Communication Link*: A single physical connection from one system/service (or node) to another.

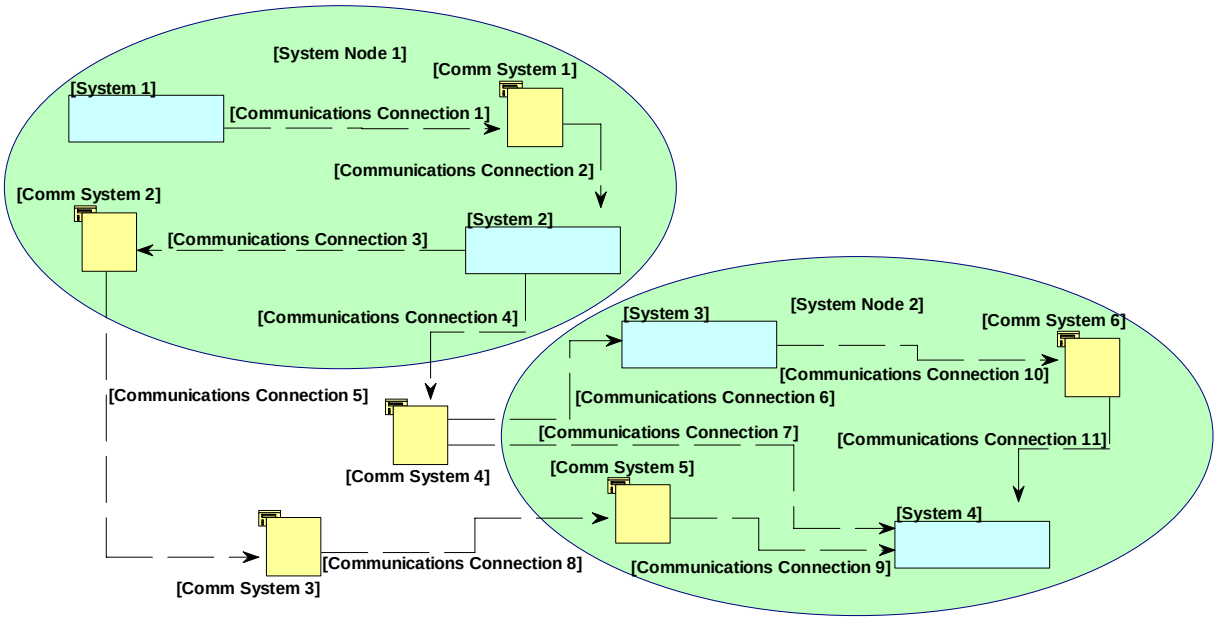


Figure 14: Example Systems/Services Communication Diagram

2.11.2 Product Integration

2.11.2.1 Vertical Integration

- System and service nodes in the Enterprise-Level SV-2 correlate to the nodes in Program-level SV-2. This alignment ensures that the Enterprise-level system views are an accurate reflection of systems being built and maintained as captured in Program-level system views. This provides a linking mechanism for Program-level system architectures to provide further levels decomposition and detail.
- The FAA Telecommunications Infrastructure (FTI) program's systems are depicted as a Telecommunications Infrastructure Backbone. This backbone supports the NAS Ops IP Network (also provided by FTI). The IP network in turn allows for a NAS enterprise messaging service. However, NAS enterprise messaging services are accessible to any external system that can reach the NAS IP backbone (directly, or through an appropriate boundary gateway). This implies that the messaging service will be accessible to external stakeholders via boundary protection, as well as aircraft via Air-to-Ground Data Communications. This allows development programs to depict their communication architectures with alignment to the NAS technical communications infrastructure.

2.11.2.2 Horizontal Integration (Inter)

- The communications services and communications systems should be consistent across interrelated Program-level architecture products.
- The interrelated Program-level architecture products should have an instantiated communication path (links and networks) that links the architectures products for communications.

2.11.2.3 Horizontal Integration (Intra)

- An interface in the SV-1 is implemented by communications link(s) or communications network(s) in the SV-2.
- Technical standards in TV-1 apply to and sometimes constrain communications systems, communications links, and communications networks in SV-2.
- Timed standard forecasts in TV-2 impact communications systems, communications links, and communications networks in SV-2.

2.11.3 Modeling Best Practices and Rules

- The SV-2 can present either an intermodal or intra-nodal diagram, since the SV-2 depicts implementation details for the SV-1 interfaces by decomposing them into communication systems/service, links, and networks.
- Keep all systems/service elements the same size if possible.
- Nodes and communication links should be arranged to minimize overlapping lines to improve readability of the diagram.

2.11.4 Product Style Guidance

Object	Representation (shape)	Object Color	Text/ Labeling
System Node	Rectangle	Light Green RGB Values: 204, 255, 204	Arial 16 pt
System	Rectangle	Light blue RGB Values: 219, 238, 244	Arial 16 pt
Service Family	Rounded Corner Square	White RGB Values: 255, 255, 255	Arial 16 pt
Application Service	Rectangle	Yellow RGB Values: 255, 255, 102	Arial 16 pt
Infrastructure Service	Rectangle	Purple RGB Values: 179, 162, 199	Arial 16 pt
Communication System	Dashed line	Black RGB Values: 0, 0, 0	Arial 10 pt

Object	Representation (shape)	Object Color	Text/ Labeling
Communication Link	Arrow (one direction)	Black RGB Values: 0, 0, 0	Arial 10 pt

2.12 System/Service Functionality Description (SV-4)

2.12.1 Product Description, Elements, and Example

The Systems Functionality Description (SV-4) describes the system/service functions and flows of data that are required to support the mission of the NAS and is the SV counterpart to the OV-5. The primary purpose of SV-4 is to develop a clear description of the necessary system functions and associated data flows that are input (consumed) by and output (produced) by each resource. A complete SV-4 includes both a Systems/Services Functional Hierarchy Diagram and a Systems/Services Data Flow Diagram (DFD). At the Enterprise-level, the scope of this product may not include which systems/services perform which functions, and at the Program-level, it may be system/service specific. Variations may focus on intra-nodal system/service data flow, system/service data flow without node considerations, function to system/service allocations, and function to node allocations. The system/service functions documented in the SV-4 may be identified using a functional taxonomy (i.e., FEA Application Reference Model, JCSFL) and correlated to SV-1 and SV-2 systems/services. System/service functions can include Human Computer Interface (HCI) and Graphical User Interface (GUI) functions or functions that consume or produce data from/to functions that belong to external systems/services.

The SV-4 contains the following elements:

- *Service:* Capabilities that are implemented using design principles that support interoperability, sharing and the reuse of functions across the enterprise. They exist as operationally oriented processes, applications, infrastructure, or any combination.
- *Function*: A data transform that supports the automation of activities or information element exchange.
- *External Source*: An object (can be a person, thing, or another system) interacting with the system, but not encompassed by the system.
- *Data Flow*: Information/resources transferred from one function to another.
- *Data Store*: A database or related means of storing data.

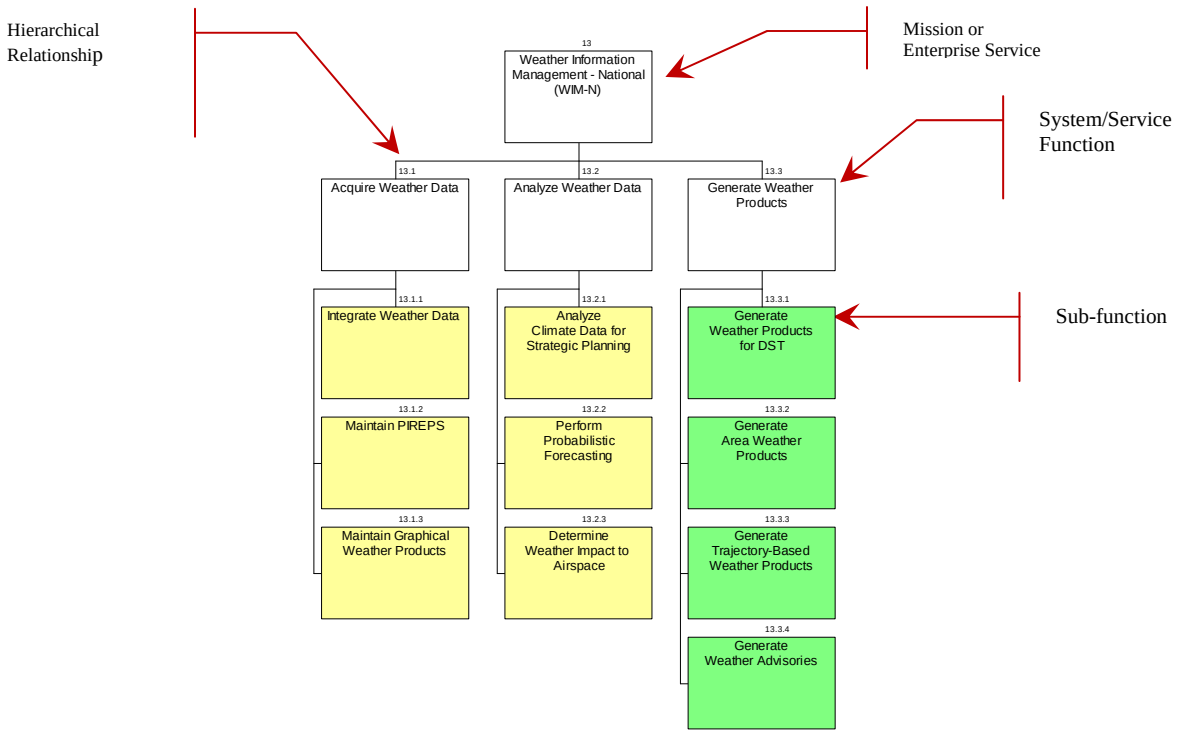


Figure 15: Example SV-4 Hierarchy

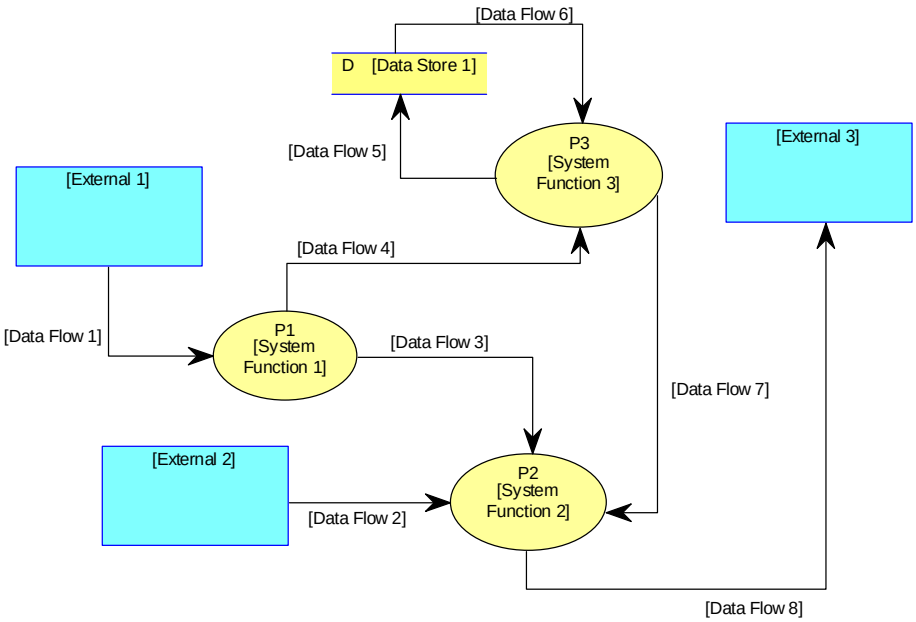


Figure 16: Example Data Flow Diagram

2.12.2 Product Integration

2.12.2.1 Vertical Integration

- Top-level system functions within Program-level architectures align to Enterprise-level system functions identified in the SV-4. Programs can then decompose system functions to the appropriate level.

2.12.2.2 Horizontal Integration (Inter)

- The systems utilized in the Program-level architecture products should be consistent across interrelated Program-level architecture products and have a common interface if applicable.

2.12.2.3 Horizontal Integration (Intra)

- Each lowest level system function in the SV-4 is aligned to the lowest level operational activity that it supports in the form of the SV-5. Multiple system functions may support a single operational activity.
- Data flows in the Data Flow Diagrams should satisfy an Information Exchange Requirement (IER) in the OV-3. This data exchange should be reflected via the SV-6.
- The system functions should correspond to the requirements specified in Section 5 (Functional Requirements) of the program requirement document.

2.12.3 Modeling Best Practices and Rules

- The SV-4 should represent unique functions describing 'what' a system/service has to perform, not 'how' the function is to be performed. The decomposition of a parent function into child functions is intended to provide additional clarity while staying within the scope of the parent. At the Program-level, by FID, functions should be decomposed to the level necessary to fully describe the Functions performed by the system/service.
- All symbols representing functions in the diagram should be the same size.
- Data exchanges/flows should have one and only one source function.
- Functions and data flows should be arranged to minimize overlapping lines to increase readability of the diagram.
- A function can send a data exchange/flow to multiple destination functions; however, if the content/attributes are different (and important enough to capture at the Enterprise-level) a separate, uniquely titled/defined data exchange/flow should be created.
- Data Stores can be created in the DFD to indicate known and anticipated means for storing data. They can be used to denote transactional data, net-centric data, and master data. Transactional data stores show the data that is created and manipulated in support of specific business processes and is highly dynamic. Net-centric data stores are the location for pervasive and highly accessed/shared data that is made available for constant push and pull operations. This might include weather, NOTAMS, and flight plan data. Master data stores depict the most stable shared data that is controlled by the enterprise. This might include geospatial/terrain, obstacle, defined routes, and aerodrome data.

- Data flows should be unidirectional, instead of bi-directional with the direction of the arrow pointing in the direction of the system/service data flow.
- SV-4 DFD symbol colors: If non-standard colors or lines are used for externals, functions, or data flows, a legend should be attached. If a data flow is colored, the body color and line color should match, so the head does not look odd.
- The SV-4 DFD should try to limit the number of functions and complexity in any one model. It's more appropriate to decompose or abstract functions to keep a comprehendible level of detail.

2.12.4 Product Style Guidance

Object	Representation (shape)	Object Color	Text/ Labeling
System Function	Rectangle	Yellow RGB Values: 255, 255, 102	Arial 16 pt
External Source	Rectangle	Blue RGB Values: 142, 180, 227	Arial 16 pt
Data Flow	Arrow (one direction)	Black RGB Values: 0, 0, 0	Arial 12 pt

Note: SV-4 Hierarchies only depict System Functions. Color coding may be applied depending on the diagram complexity, but should be reflected with a legend if needed.

2.13 Operational Activity to System/Service Function Traceability Matrix (SV-5)

2.13.1 Product Description, Elements, and Example

The Operational Activity to System/Service Function Traceability Matrix (SV-5) specifies the relationships between the set of operational activities applicable to an architecture, and the set of related system/service functions defined in that architecture. The SV-5 depicts the mapping of operational activities to system/service functions, therefore identifying the transformation of an operational need into a purposeful action performed by a system or service. The SV-5 can be extended to depict the mapping of capabilities (i.e., operational improvements) to operational activities, operational activities to systems/services, and relate the capabilities to the systems/services that support them. Such a matrix allows decision-makers and planners to quickly identify stovepipe systems/services, redundant/duplicative systems/services, gaps in capability, and possible future investment strategies.

The SV-5 contains the following elements:

- *Operational Activity:* An action performed in conducting the business of an enterprise that either generates or consumes the information exchange. It is used to portray operational actions and not hardware/software system functions.
- *Function*: A data transform that supports the automation of activities or information element exchange.

Operational Activity / *System/Service Function*	*Perform Separation*	*Synchronize Aircraft*	*Transfer Control Responsibility*	*Support Emergency Response*	…
Satellite Navigation Service	X	X			
Aircraft Navigation Processing	X			X	
Separation Management					
…..					

Figure 17: Example Operational Activity to System/Service Function Matrix

2.13.2 Product Integration

2.13.2.1 Vertical Integration

- Programs are able to review the enterprise SV-5 to determine if system/service function redundancies exist, and ensure that they are not creating additional redundancies in the new development efforts.

2.13.2.2 Horizontal Integration (Inter)

- Mappings identified in the SV-5 should be consistent across all interrelated programs.

2.13.2.3 Horizontal Integration (Intra)

- Each leaf level activity in the OV-5 appears as a column in the SV-5 matrix.
- Functions in the SV-4 should map one-to-one with functions in the SV-5.
- Systems in the SV-1 match systems in the SV-5.

2.13.3 Modeling Best Practices and Rules

- The objects along the X-axis (across the top of the matrix) represent the leaf level operational activities.
- The objects along the Y-axis (down the left side of the matrix) represent the leaf level system/service functions.
- An “X” indicates an alignment mapping.
- The relationship between operational activities and system/service functions can be many-to-many. This essentially means that one operational activity can be supported by multiple system/service functions, and one system/service function can support multiple operational activities.

2.13.4 Product Style Guidance

- An “X” should be used to indicate a mapping between the matrix axes.

2.14System/Service Data Exchange Matrix (SV-6)

2.14.1 Product Description, Elements, and Example

The System/Service Data Exchange Matrix (SV-6) specifies the characteristics of the data exchanged between functions and their corresponding/associated systems or services. This product focuses on the automated information exchanges (from the OV-3) that are implemented in systems or services. Non-automated information exchanges, such as verbal orders or directives, are captured in the operational views. The SV-6 provides the architect or analyst with a view into the data, to include where it originates and its associated attributes. Specifically, the focus of the SV-6 is on how the data exchange is implemented, in specific details covering periodicity, timeliness, throughput, size, information assurance and security characteristics needed for the exchange. Additionally, the data elements, format, and media type, accuracy, and standards are also described.

The SV-6 relates to and grows out of the OV-3. The operational characteristics for the OV-3 information exchanges are replaced with the corresponding system/service data characteristics. Performance attributes for the operational information exchanges are replaced by the actual system/service data exchange performance attributes for the automated portions of the information exchange. On the SV-6, each operational needline is decomposed into the interfaces that are the system equivalents of the needline. The SV-1 graphically depicts system/service data exchanges as interfaces that represent the automated portions of the needlines; the implementation of the SV-1 interfaces is described in the SV-2; the system/service data exchanges documented in the SV-6 trace to the information exchanges detailed in the OV-3 and constitutes the automated portion(s) of the OV-3 information elements.

The SV-6 contains the following elements:

- *System/Service Data Exchange*: The collection of system/service data elements and their performance attributes.
- *System/Service Interface*: An abstract representation of one or more communication paths between system nodes or between systems.
- *System/Service*: Any organized assembly of resources and procedures united and regulated by interaction or interdependence to accomplish a set of specific functions.
- *Function*: A data transform that supports the automation of activities or information element exchange.
- *Content Description*: Describes the data exchange and nature of transaction.

The SV-6 Data Exchange Matrix also contains the following attributes:

<u>Data Description</u>

- *Format Type:* Application level format (e.g. XML/DTD, ASCII Text) with parameters and options used, or other relevant protocols.
- *Media Type:* Type of media used.
- *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.

Producer Information

- *Data Standard :* e.g., AIXM (see TV-1/2)

Transaction Description

- *Transaction Type:* Descriptive field that identifies the type of exchange.
- *Triggering Event:* Brief textual description of events(s) that triggers the data exchange.
- *Criticality:* The criticality assessment of the information being exchanged in relationship to the mission being performed.
- *Connection Initiation Responsibility:* Identifies the source of the connection (e.g., an external system, an FAA system external to the DMZ, an FAA system inside the DMZ, etc.).
- *Data Flow Direction:* Identifies the direction of the data flow (e.g., Internal NAS system to Internal NAS Enterprise Security Gateway DMZ).
- *Total Port Range Used:* Identifies the ports used (e.g., 20, 21).
- *Transport Protocol:* Identifies the communications protocol that establishes a connection and ensures the data is transported (e.g., TCP).
- *Application Protocol:* Identifies the protocol that governs how processes are performed, providing the bytes that carry the messages and the responses for various processes (e.g., FTP, WAP, etc.).

Performance Attributes

- *Periodicity:* Frequency of system data exchange transmission – may be expressed in terms of worst case or average frequency.
- *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:* The size of the data exchanged.

Information Assurance

- *Access Control:* The class of mechanisms used to ensure only those authorized can access a specific system data element.
- *Availability:* The relative level of effort required to be expended to ensure that the system data can be accessed.
- *Confidentiality:* The kind of protection required for system data to prevent unintended disclosure.
- *Dissemination Control:* The kind of restrictions on receivers of system data based on sensitivity of system data.
- *Integrity:* The kind of requirements for checks that the content of the system data element has not been altered.
- *Non-Repudiation Producer:* The requirements for unassailable knowledge that the system data received was produced by the stated source.
- *Non-Repudiation Consumer:* The requirements for unassailable knowledge that the system data sent was consumed by the intended recipient.

Security

- *Protection:* The code that represents how long the system data must be safeguarded.

- *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:* e.g., ISO 27002, NIST SP 800-53 (see TV-1/2)

The unique identifier for each Service interface from the SV-1

The name of the Service sending the Data Exchange

The name of the Sub-function sending the Data Exchange

The name of the Service receiving the Data Exchange

Service Interface ID	Data Exchange ID	Service Data Exchange Name	Producer			Consumer		
			Sending Service Node Name	Sending Service Name	Sending Sub-function	Receiving Service Node Name	Receiving Service Name	Receiving Sub-function
SI_001_004	SDX_0001	Airport Status	Enterprise Wide Operations (National)	Aeronautical Information Management	Maintain Airport Status	Enterprise Wide Operations (National)	Flow Contingency Management	Assess Airport Status
	SDX_0002	Airport Structures Definition	Enterprise Wide Operations (National)	Aeronautical Information Management	Maintain Airport Definitions	Enterprise Wide Operations (National)	Flow Contingency Management	Assess Airport Status

The name of each data exchange

The name of the Service Node originating the Data Exchange

The name of the Service Node receiving the Data Exchange

The name of the Sub-function receiving the Data Exchange

Figure 18: Example System and Service Data Exchange Matrix

Note: The system/service interface identifiers have the following naming convention: SI_XXX_YYY, where "SI" represents "System/Service Interface," "XXX" represents a numerical code of the sending service, and "YYY" represents a numerical code of the receiving service.

2.14.2 Product Integration

2.14.2.1 Vertical Integration

- Data elements at the Program-level architecture align to system/service data elements in the Enterprise-level architecture.

2.14.2.2 Horizontal Integration (Inter)

- The system/service data exchanged between interrelated Program-level architecture products system/service elements must be consistent.

2.14.2.3 Horizontal Integration (Intra)

- If any part of an information exchange in the OV-3 originates from or flows to an operational activity that is to be automated, then that information exchange should map to one or more service data elements in the SV-6.
- Each system/service data element appearing in a system/service data exchange is graphically depicted by one of the interfaces in the SV-1; an interface supports one or more system/service data exchanges.
- System/service data flows in the SV-4 should map to system/service data elements appearing in system/service data exchanges of SV-6.

- Technical standards in the TV-1 apply to and sometimes constrain system/service data elements in SV-6.
- Timed standard forecasts in TV-2 impact system/service data elements in SV-6.
- The data exchanges should correspond to the requirements specified in Section 5.2 (Information Requirements) of the program requirement document.

2.14.3 Modeling Best Practices and Rules

- All source and destination types are System Entity as this is the default that SA provides for the symbol chosen in the other system view products. This column is typically deleted after the generation of the SV-6. The name of the data exchange is sufficient to provide meaning.
- An identifier for each data exchange should be used to aid navigation of the SV-6 matrix.

2.14.4 Product Style Guidance

The SV-6 matrix is typically automatically generated from data stored within SA. This automatic generation is accomplished through the SV-4 DFDs and Systems/Services Interface Description (SV-1). The SV-6 matrix can be created in Microsoft Excel or Word, using data capture from SA or by simply using the Report Generator tool in SA. How the final matrix is created, is a decision for the program. However, the data that is used to populate the matrix must come from SA. For more information on SA reports, please reference the following website: http://publib.boulder.ibm.com/infocenter/rsysarch/v11/nav/22.

2.15 System/Service Performance Parameter (SV-7)

2.15.1 Product Description, Elements, and Example

The Systems/Services Performance Matrix (SV-7) 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. 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.

The SV-7 builds on the SV-1, SV-2, SV-4, and SV-6 by specifying performance parameters for systems/services and system/service hardware/software items and their interfaces (SV-1), communication details (SV-2), their functions (SV-4), and their data exchanges (SV-6). Performance parameters include all technical performance characteristics of systems/services when requirements can be developed and defined.

The SV-7 contains the following elements:

- *Service Family*: A grouping of independent services that can be arranged or interconnected in various ways to provide different capabilities.
- *Application Service*: A concrete supporting service that is typically identified and defined by application developers, is specific to the application scope they are defined under, and

is generally used to perform fine-grained application-specific functions such as data collection, validation, and transfer.

- *Infrastructure Service*: A concrete service that supports non-business related functions of the enterprise and is generally shared and used by Enterprise Services (and sometimes Application Services).
- *System/Service*: Any organized assembly of resources and procedures united and regulated by interaction or interdependence to accomplish a set of specific functions.
- *Function*: A data transform that supports the automation of activities or information element exchange.
- *System/Service Data Exchange*: The collection of system/service data elements and their performance attributes.
- *Performance Parameters*: Describes how system performance will be measured.

Parameter ID	System/Element	Performance Requirement	Threshold	Objective
Hardware				
H 1.1	SPECS 2 Transmitter	Transmission Rate	1.2 mbps	3 mbps
H 2.1	SPECS 2 Receiver	Gain	40 dB	60 dB
H 2.2		Signal to Noise Ratio	15 dB	20 dB
H 3.1	SPECS 2 Signal Processor	Comms Channel Bandwidth Support	1.2 GB	2 GB
H 4.1	SPECS 2 Video Recorder	Resolution	1024x768	1440x600
H 4.2		Storage Capacity	14 hours	20 hours
Software				
S 1.1	Video Analysis	Accuracy of target location	9 mi	10 mi
		Accuracy of target speed	1 mph	2 mph
S 1.2	Target Status Alerting	Alert Response Time	45 seconds	30 seconds

Figure 19: Example System and Service Performance Parameters Matrix

2.15.2 Product Integration

2.15.2.1 Vertical Integration

- The systems and/or service referenced should correspond to systems and/or services in the Enterprise-level SV-1.

2.15.2.2 Horizontal Integration (Inter)

- The systems functional characteristics for processing and interface characteristics for system data exchanges in the Program-level architecture products should meet the specified criteria (required performance) from interrelated Program-level products.
- The performance requirements for interrelated Program-level architecture products should be sufficient to meet overall performance requirements.

2.15.2.3 Horizontal Integration (Intra)

- Performance parameters of the SV-7 apply to systems/services, subsystems, and system/service hardware/software items in the SV-1.
- Performance parameters of the SV-7 relate to communications systems/services, links, and networks and should map to the corresponding elements in the SV-2.

- Performance parameters in the SV-7 relate to interface performance and data exchange capacity requirements and should trace to system/service data exchanges in the SV-6.
- The required system/service performance should correspond to the requirements specified in Section 3.2 (Performance Requirements) of the Program Requirement Document.

2.15.3 Modeling Best Practices and Rules

- Additional columns may be inserted to express expected or required performance characteristics at specified times in the future between T_0 and T_n. If the future performance expectations are based on expected technology improvements, then the performance parameters and their time periods should be coordinated with the TV-1/2.
- 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 the NAS Infrastructure Roadmaps.
- In order to auto generate the report using SA's default SV-7 matrix, all fields must be populated from the SV-1 and SV-6.

2.15.4 Product Style Guidance

Not applicable.

2.16 Systems/Services Event-Trace Description (SV-10c)

2.16.1 Product Description, Elements, and Example

The Systems/Services Event-Trace Description (SV-10c) provides a time-ordered examination of the system/service data elements exchanged between participating systems/services (internal/external), system/service functions, or human roles as a result of a particular scenario. The SV-10c may reflect system/service-specific aspects or refinements of critical sequences of events described in the operational views. The SV-10c is valuable for moving to the next level of detail from the initial solution design, to help define a sequence of functions and system data interfaces, and to ensure that each participating resource or system role has the necessary information it needs, at the right time, to perform its assigned functionality.

The SV-10c contains the following elements:

- *Lifeline:* A role in an interaction that represents a participant over a period of time. Lifelines may be composed of systems/services, system/service functions, or human roles. It is shown as a vertical line, parallel to the time axis, with a head symbol showing its name and type
- *Event*: An event in a sequence diagram implies the action that produced it.
- *Activation Box (Optional):* A rectangle box drawn on the lifeline to represent that processes are being performed in response to an event.
- *Interaction Fragment (Optional)*: A structural piece of an interaction that shows more complex flow of control. A fragment has an operand keyword and one or more interaction operands such as alternative, loop, optional, parallel, etc.

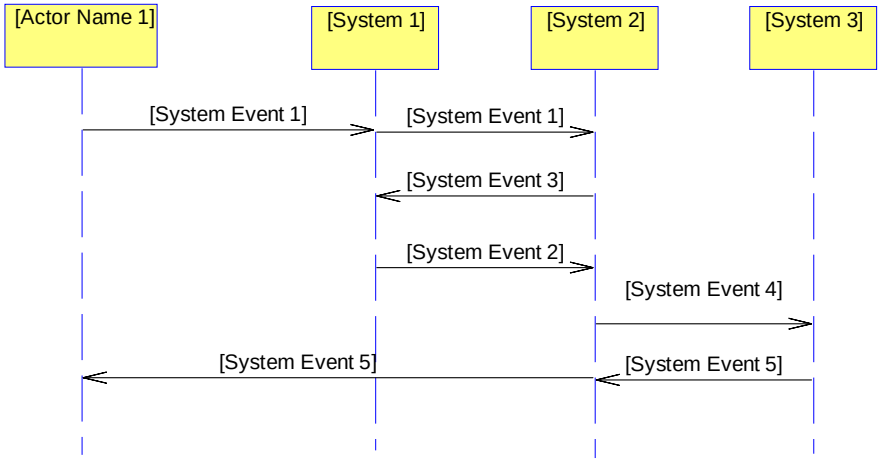


Figure 20: Example System/Service Event-Trace Diagram

2.16.2 Product Integration

2.16.2.1 Vertical Integration

- The systems/services utilized in the Program-level architecture products should be consistent with the Enterprise-level systems/services.
- The system/service interfaces established between systems/services in the Program-level architecture products should be consistent with system/service interfaces identified at the Enterprise-level.
- The systems/services utilized in the Program-level architecture products should be consistent with the NAS EA Infrastructure Roadmaps.

2.16.2.2 Horizontal Integration (Inter)

- The system/service threads, scenarios sequences, and data exchanges of interrelated Program-level architecture products should be consistent with other products, including the SV-1, SV-4, or SV-10c from interrelated Program-level architectures.

2.16.2.3 Horizontal Integration (Intra)

- The events trigger should be consistent with the corresponding set of data flows captured in the SV-4 DFD.
- The systems/services should be consistent with the systems/services on the SV-1, and functions consistent with those in the SV-4.
- Operational nodes should be consistent with the OV-2.
- The scenarios and solutions detailed in the SV-10c should satisfy the operational needs (information exchanges/operational events) in the OV-6c.

2.16.3 Modeling Best Practices and Rules

- The recommended notation for this product is UML.

- The SV-10c provides a time-ordered examination of the data elements exchanged between participating systems or system nodes, and the required time interval between exchanges may be shown as a vertical measure between the arrows.
- Different scenarios should be depicted by separate diagrams.
- Each event-trace diagram will have an accompanying description that defines the particular scenario or situation.
- The diagram should be constructed to read from top to bottom, left to right. The items across the top of the diagram represent systems, system functions, or human roles that take action based on the types of events.
- Each diagram may represent systems/services (internal/external) or system/service functions, but not both in the same diagram. Human roles may also be used in the diagram along with either systems/services or system/service functions in order to describe the human's interfaces to the systems/services or system/service functions.
- Each system, function, or human role has a lifeline associated with it that runs vertically.
- Labels indicating timing constraint 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 lifelines represent events, and the points at which they intersect the lifelines represent the times at which the system/service/function/role becomes aware of the events.
- Use unidirectional arrows. The direction of the arrows (events) represents the flow of control from one system/service/function/role to another based on the event.
- The content of the exchanges that connect lifelines in an SV-10c may be related with interfaces from the SV-1, data flows from the SV-4 and SV-6, and data schema entities from the SV-11.

2.16.4 Product Style Guidance

Symbol/Object	Representation (shape)	Object Color	Text/ Labeling
System (Node) Event Timeline	Vertical dashed lines	Yellow/blue (SA default) RGB Values: 255, 255, 102	Arial 16 pt
System Event	Horizontal arrow	Black (SA default) RGB Values: 0, 0, 0	Arial 12 pt

2.17Physical Schema (SV-11)

2.17.1 Product Description, Elements, and Example

The SV-11 is among 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. The product serves several purposes, including providing as much

detail as possible on the system data elements exchanged between systems, thus reducing the risk of interoperability errors; and providing system data structures for use in the system design process, if necessary. The SV-11 contains the following elements:

- *System Data Element*: a piece of data, data flow, or system event.
- *Attributes: characteristics that describe the system data element to which they are associated.*
- *System Data Relationships: Connections that describe the relationships and cardinality between data elements.*

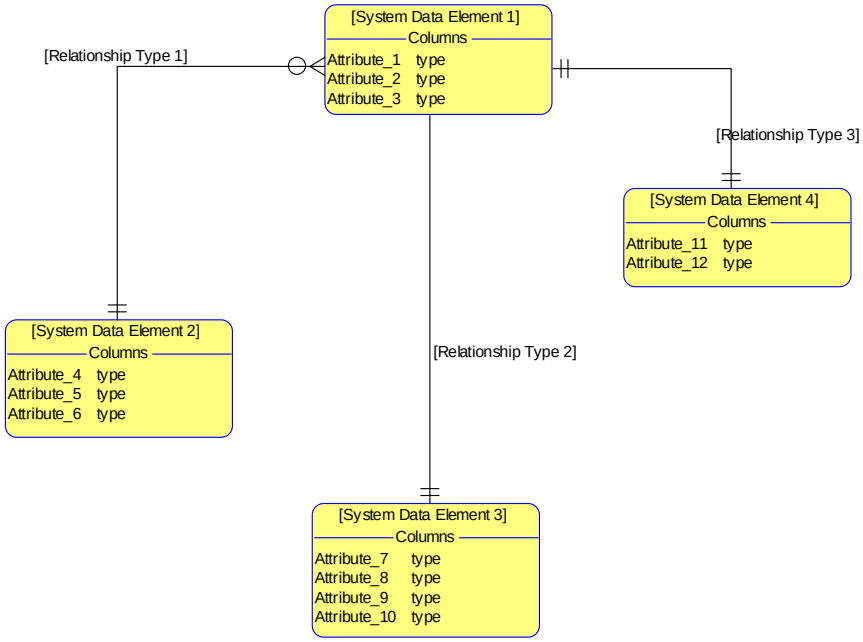


Figure 21: Physical Schema

2.17.2 Product Integration

2.17.2.1 Vertical Integration

- The data elements in the Program-level architecture products should align with Enterprise-level OV-7.

2.17.2.2 Horizontal Integration (Inter)

- The system data elements should be consistently represented across Program-level SV-4 and SV-11.

2.17.2.3 Horizontal Integration (Intra)

- The System Data Elements should correspond to the system functions input and output in the SV-4.
- The System Data Elements should correspond to the data elements that are produced by system events in the SV-10c.

2.17.3 Modeling Best Practices and Rules

- For a relational SV-11, the IDEF1X modeling notation should be employed.

2.17.4 Product Style Guidance

Object	Representation (shape)	Object Color	Text/ Labeling
Table	Rectangle	Grey RGB Values: 242, 242, 242	Arial 16
Column	Text	Black RGB Values: 0, 0, 0	Arial 14
Relationship	Line with cardinality icons	Black RGB Values: 0, 0, 0	Arial 14

2.18 Technical Standards Profile and Forecast (TV-1/2)

2.18.1 Product Description, Elements, and Example

A Technical Standards Profile (TV-1) consists of the systems standards rules that govern and sometimes constrain the choices that can be made in the design, implementation, and operation of an 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).

A Technical Standards Forecast (TV-2) contains expected changes in technology-related standards and conventions that appear in the TV-1. It contains predictions about the availability of emerging standards, and delineates the standards that will potentially impact the relevant system elements (from SV-1, SV-3, SV-4, SV-6 and OV-7).

Standards that need to be submitted for consideration and use, that are not present in the pre-defined encyclopedia, should be sent to the NAS EA development team for review. After the standard review has been performed, either the standard will be integrated into the encyclopedia and made available to reference, or will be sent back the submitter for further details. In order to submit a standards update request, submit a request through the "Contact Us" link on the NAS EA Portal home page (https://nasea.faa.gov/comment/main/contact_us).

The TV-1/2 contains the following elements:

- *Standard Designation*: The designation or acronym of the standard referenced.
- *Standard Title*: The title of the standard referenced.
- *Standard Date*: The date the standard referenced was activated.
- *Status:* The status of the standard referenced (e.g., draft, active, legacy, etc.)
- *Timeframe:* Identifies the timeframe in which the standard referenced applies.

Standard Designation	Standard Title	Standard Date	Status	Timeframe
1.0 NASEA Policies and Guidance				
ITMRA of 1996	Information Technology Management Reform Act of 1996	February 10, 1996	active	
OMB Circular A-130	Management of Federal Information Resources	November 28, 2000	active	
FISMA-II	Federal Information Security Management Act Version II		draft	Near-Term(1-4 yrs
2.0 Information Processing Standards				
2.1 System and Software Engineering Services				
ANSI/EIA 632-1998 also known as ANSI/EIA 632-1999	Processes for Engineering a System	January 7, 1999 (reaffirmed Sept 2003)	active	
ANSI/ISO 9899-1999	ANSI Standard for the C Programming Language	1999	active	
FAA SEM v3.1	FAA System Engineering Manual Version 3.1	October 1, 2006	active	
FAA-STD-018A	Computer Program Quality Program Requirements	September 30, 1987	active	
IEEE 1220-2005	Standard for Application and Management of the Systems Engineering Process	September 9, 2005	active	
IEEE Std 12207-2008	Systems and Software Engineering-Software Life Cycle Processes	February 1, 2008	active	
ISO/IEC Std 14882:2003	Programming Language C++	October 1, 2003	active	
ISO/IEC Std 15288-2008	System and Software Engineering - System Life Cycle Processes	2008	active	
MIL-STD-498	Software Development and Documentation	December 5, 1994	legacy	
2.2 User Interface Services				
FED STD-795	Uniform Federal Accessibility Standards (UFAS)	April 1, 1988	active	
IEEE Std 1003.1 2004	Portable Operating System Interface Part 1 - Library Routines, Systems Calls and Header Files	2004	active	
ISO/IEC 9945-1:2003	Information technology -- Portable Operating System Interface (POSIX) -- Part 1: Base Definitions	2003	active	

Figure 22: Example Technical Standards Profile and Forecast

2.18.2 Product Integration

2.18.2.1 Vertical Integration

- Through inheritance, technical standards that are aligned to Enterprise-level product elements and subsequently allocated to Programs and/or systems will also apply to Program-level architectural elements.

2.18.2.2 Horizontal Integration (Inter)

- The standards utilized in the Program-level architecture products must be consistent with standards referenced by interrelated Program-level architectures to ensure interoperability.

2.18.2.3 Horizontal Integration (Intra)

- Technical standards apply to modeling techniques in OV-7 and SV-11.
- Technical standards apply to and sometimes constrain systems, subsystems, and system hardware/software items in SV-1. Timed standard forecasts impact systems, subsystems and system hardware/software items in SV-1.
- Technical standards apply to and sometimes constrain communications systems, communications links, and communications networks in SV-2. Timed standard forecasts impact communications systems, communications links, and communications networks in SV-2.
- Technical standards apply to system functions in SV-4. Timed standard forecasts impact system functions in SV-4.
- Technical standards apply to and sometimes constrain system data elements in SV-6. Timed standard forecasts impact system data elements in SV-6.
- Technical standards constrain evolving systems, subsystems, and system hardware/software items of the NAS Infrastructure Roadmaps.

- Timed technology forecasts may force a standard to move to its next version. Timed standard forecasts may depend on the timed technology forecast becoming available.

2.18.3 Modeling Best Practices and Rules

- Not applicable.

2.18.4 Product Style Guidance

- Not applicable.

2.19 Service Roadmap

2.19.1 Product Description, Elements, and Example

The "NAS Service Roadmaps" are a rolling 15-year strategic roadmap that depict the expected evolution and delivery of NAS services, capabilities, and benefits over time. More specifically, it outlines the strategic activities for service and capability delivery to sustain and improve NAS operations towards the target state vision. The Operational Improvements and sustainment initiatives identified on the roadmaps are used to guide, inform, and focus deliberations on NAS capabilities. The Service Roadmaps are updated annually as research and analyses more clearly define the evolution of NAS services. The XV-1 contains the following elements:

- *Operational Improvement*: a discrete strategic activity for service and/or capability delivery to improve NAS operations. They are expressed as cross-domain statements comprising sets of anticipated benefits to be realized at some future date.
- *Roadmap Swim Lanes*: A section of a roadmap diagram that logically groups other roadmap elements by NAS Service.
- *NAS Service*: The services and capabilities that the NAS requires to provide safe and efficient Air Traffic Control (e.g. ATC-Separation Assurance, Trajectory Management Synchronization) and Congressionally mandated in the Federal Aviation Act.

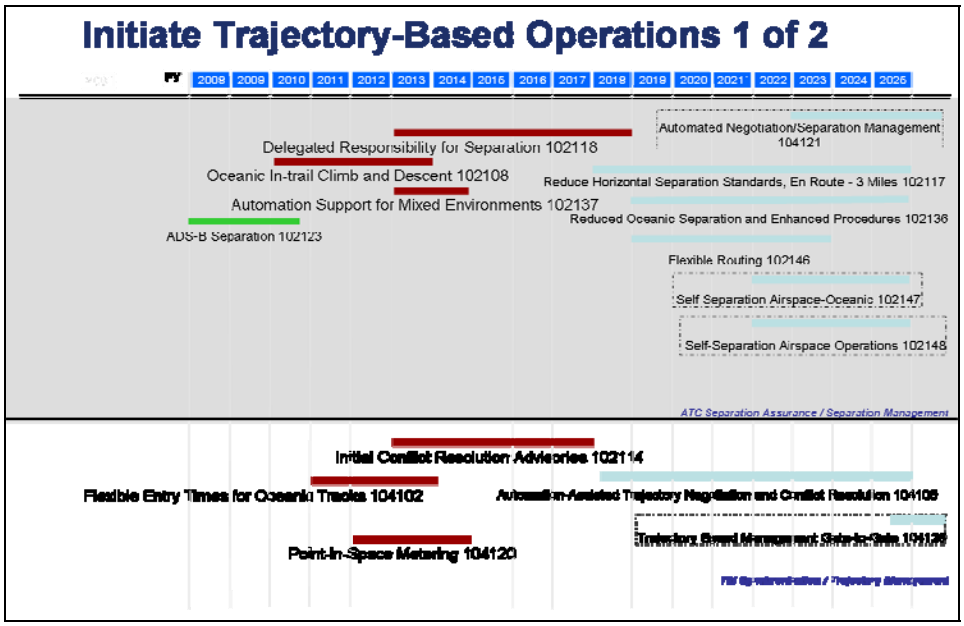


Figure 23: Example Service Roadmap

2.19.2 Product Integration

2.19.2.1 Vertical Integration

- A Program-level AV-1 should identify which Operational Improvements it aligns to and how its part in the implementation of the capability.

2.19.2.2 Horizontal Integration (Inter)

- It is possible that more than one Program could support the implementation of an Operational Improvement.

2.19.2.3 Horizontal Integration (Intra)

- An Operational Improvement should be further described by one or more Operational Requirements and modeled within the appropriate operational architecture views.

2.20 Infrastructure Roadmap

2.20.1 Product Description, Elements, and Example

The "Infrastructure Roadmap" is a 15-year strategic roadmap that depicts the planned infrastructure improvements and sustainment initiatives, effectively showing the evolution of major FAA programs/systems in today's NAS infrastructure to meet the target state vision. The Infrastructure Roadmaps contain programmatic and schedule information that define the enabling infrastructure (i.e., people, systems, facilities, and support activities) for service delivery; identify system replacements, convergence and modernization; and the relationships among various infrastructure elements. The Infrastructure Roadmaps also identify key decision points that represent acquisition, strategy, and policy decisions associated with a particular program/system. The decision points indicate the FAA's approval of a particular improvement/sustainment initiative; an investment decision that must precede implementation of an improvement initiative; or the research and/or analysis that must be conducted before an

investment decision or solution implementation. The Roadmaps, combined with funding data, facilitate analysis of cost and schedule tradeoffs, and are used to guide, inform, and focus deliberations on the NAS infrastructure. The XV-3 contains the following elements:

- *Diagram*: an individual power point slide assigned to a specific domain, which includes all or some of the product elements described below.
- *Roadmap Swim Lanes*: A section of a roadmap diagram that logically groups other roadmap elements. The swim lane sections are also differentiated by color. For example, all research activities are organized in a swim lane called "support activities" which has a green background color.
- *Decision Point* (DP): A decision that is made by a governing body that further shapes the direction of NextGen strategic activities. There are five categories of DPs (described below) and are uniquely identified by a DP number. It is visually depicted by a diamond shape, filled with a category specific color (described below).
- *Program*: A funded initiative that usually includes the development of one or more systems. A program defines objectives to be carried out or goals to be accomplished.
- *Project:* A temporary endeavor undertaken to create a unique product or service.
- *System*: Automation that contains functionality required to support an operational activity or procedure for a specific mission or domain. It may receive or transmit data with one or more external systems and services.
- *Support Activity*: A NAS initiative that supports the development of systems and system functions. They are depicted by shades of blue rectangle shapes, and are always placed in a green swim lane.
- *Operational Node*: A logical function or grouping, organization, facility, or human role where information is produced, consumed, or transformed.
- *Actor*: A type of operational node that performs activities within the architecture. They are depicted by an oval shape. An example of an actor is a "Sector Controller".
- *Facility*: A type of operational node that accommodates systems and actors. They are depicted by an oval shape. An example of an FAA Facility is the Mike Monroney Aeronautical Center.

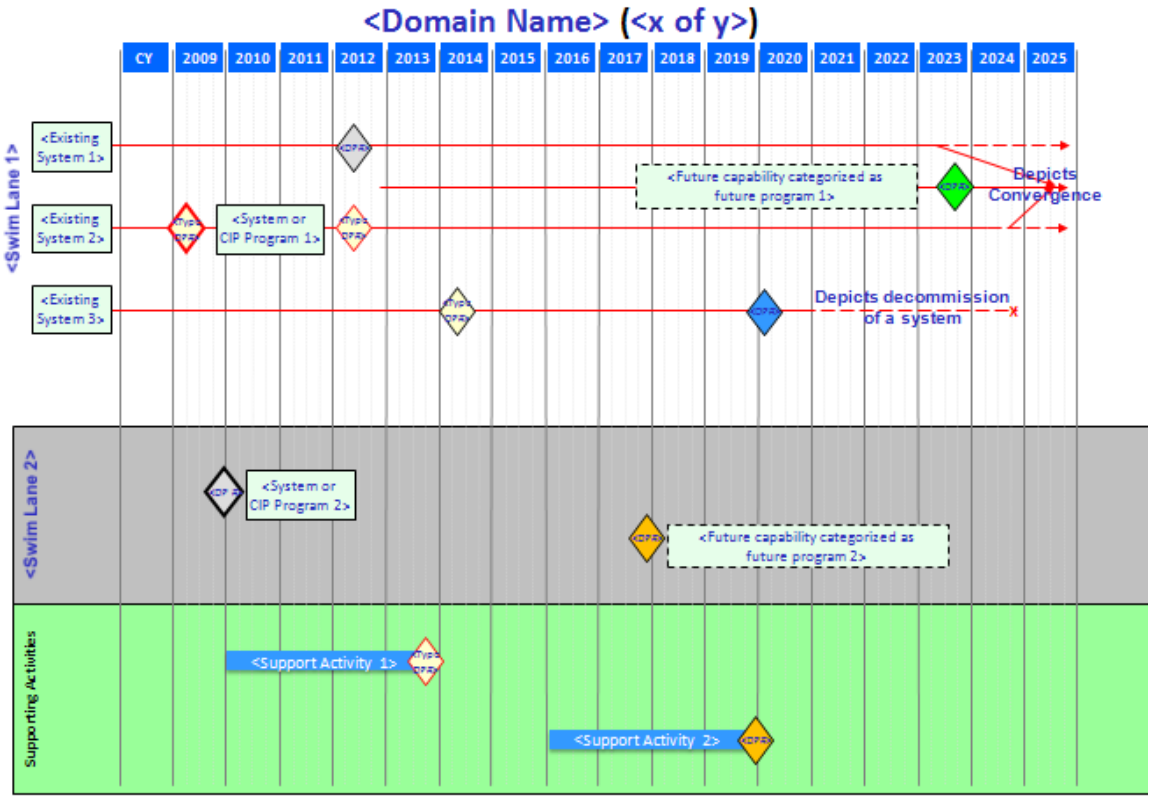


Figure 24: Example NAS Infrastructure Roadmap

2.20.2 Product Integration

2.20.2.1 Vertical Integration

- Programs, projects, systems, and support activities will be aligned to the most recently published NAS Capital Investment Plan.
- Programs, projects, and systems will be aligned to the most recently published NAS Service Roadmaps (including Operational Improvements).

2.20.2.2 Horizontal Integration (Inter)

- Existing systems will be placed on the left hand side of the roadmap diagrams, and will be aligned to the Enterprise-level As-Is SV-1.
- Future systems will be aligned to mid-term and far-term SV-1.
- Operational activities, actors and facilities will be aligned to the OV-2.
- Support activities will be aligned to the FAA NAS R&D portfolio managed by ATD&P.

2.20.2.3 Horizontal Integration (Intra)

- Grey diamonds on a roadmap diagram denote a decision point owned by another roadmap domain. Hence, the related domain roadmap should also show the same DP of a specific color other than grey.

- All roadmap elements that are shared across roadmap domains will be consistent in nomenclature and attribute descriptions.

2.20.3 Modeling Best Practices and Rules

- If quarter is unknown or cannot be planned, place DP in the middle of the year.
- For AMS DPs, all pre-implementation DPs must be identified. DPs may be left off of the roadmap only with approved tailoring.
- For non-AMS DPs, only DPs with impact to NAS Enterprise should be shown.
- Decision Points are decisions and should be written as an event requiring a decision.
- DPs are not activities. Support activities are depicted on the roadmaps as boxes with start and stop dates.
- Projected/estimated benefits to the NAS, flying public, airlines, controllers, technicians, etc.
- Number of other roadmaps that have your DP on them (implies degree of dependence and size of impact if the DP slips).
- Size/complexity of the program/project/work project associated with the DP.
- Amount of projected cost savings and reductions in FAA future budget (lifecycle cost savings).
- Complexity of the DP entrance and exit criteria - i.e., # of stakeholders that need to be bought in, political sensitivity, who has to make the decision, etc.

2.20.4 Product Style Guidance

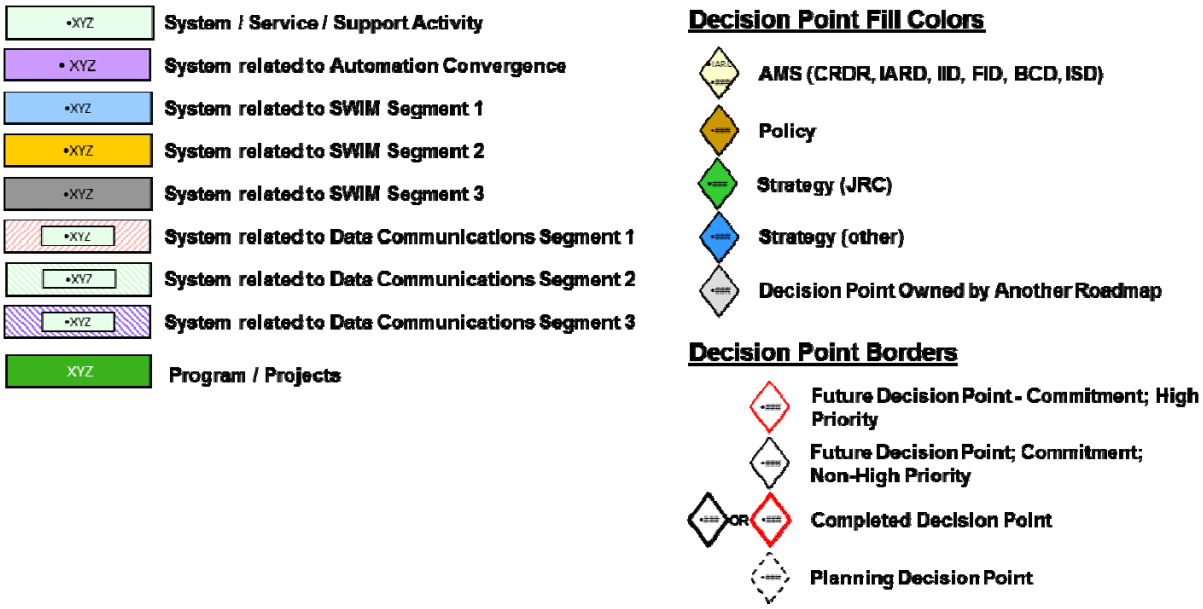


Figure 25: NAS Infrastructure Roadmap Legend

3 NAS PROGRAM-LEVEL REQUIREMENT DOCUMENT

The following sub-section describes the recommended Program-level NAS Requirements product identified in the NAS ISEF. The section includes a description of the product, the definitions of the elements contained within the product, an example of the product, as well as development and integration guidance.

3.1 Program Requirement Document (PRD)

3.1.1 Product Description, Elements, and Example

The Program Requirement Document (PRD) drives the search for a realistic and affordable solution to mission need during investment analysis. The sponsoring line of business develops a preliminary program requirement document (pPRD) during concept and requirements definition phase of the AMS, which translates the "need" in an NAS EA Roadmap into preliminary top-level functional and performance requirements. These preliminary requirements do not describe a specific solution to mission need, and should not preclude leasing, commercial, or non-developmental alternatives. During the initial investment analysis phase of the AMS, preliminary requirements are evaluated against the cost, benefits, schedule, and risk of various alternatives and brought into balance with an affordable solution to mission need. The investment analysis team develops a final program requirement document (fPRD) during final investment analysis phase of the AMS. This document defines exactly the concept of use and performance requirements the investment program is intended to achieve, and is the basis for evaluating the readiness of resultant products and services to be fielded for operational use within the FAA. Any requirements not in the fPRD are returned to the sponsoring line of business for disposition.

Note that the PRD defines the functional and performance requirements the investment program is intended to satisfy. It is NOT a system specification and should not contain the detailed level of specification necessary for that document. The system specification is derived from the PRD, and is included with a Request for Offers to prospective providers of the capability to satisfy mission need.

To ensure that requirements are derived, analyzed, and updated in the proper methodology, the Program-level requirements are developed using the FAA's SEM, Section 4.3 – Requirements Management. Table 1 contains a checklist derived from the SEM of quality metrics used in developing the Program-level requirements.

Table 1: Criteria for Evaluating Requirement Statements

Criteria	Criteria Evaluation
Attainable	- Is the requirement achievable given cost and schedule constraints? - Is the requirement achievable given technical and programmatic constraints?
Necessary	- Does the requirement satisfy an existing and documented need?
Verifiable and Measurable	- Can the requirement be verified either by test, analysis, demonstration or inspection? - Note: Words such as "optimize", "maximize", "sufficient", "satisfactory", "etcetera" should not be used as they render the requirement unverifiable. It is preferred to have a quantifiable metric against which to verify the requirement
Level of Detail	- Does the requirement only state "what" needs to happen, avoiding any prescription

Criteria	Criteria Evaluation
	regarding "how" it might be implemented?
Complete	- Are quantitative values and units of measure specified in the requirement? - Does the requirement contain all the information necessary to fully address the need?
Consistent	- Is the requirement de-coupled from all other requirements as much as possible? - Is the requirement dependent on another requirement? If so, is the dependency documented? Note: All interdependent requirements should be documented, designed, and tested together - Is the requirement consistent with all others, not contradicting any other requirements? - Are the parent, child, and peer requirements related to this requirement identified?
Format Compliance	- Does the requirement have a unique identifier? - Is the requirement written in the standard "shall" format?

The PRD contains the following elements:

- *Background*: identifies the Service-Level Mission Need Statement addressed by the Program-level requirement document and summarizes the need, briefly describes the deficiency in capability or technology opportunity, and how the proposed capability will satisfy the need.
- *Operational Concept*: describes the activities, nodes, information exchanges and intended service life for the required capability.
- *Technical Performance*: defines the operational and functional requirements, in the form of metrics or targets, the new capability must provide to satisfy the mission need.
- *Physical Integration*: defines the physical integration requirements associated with integrating the new capability into the physical environment.
- *Functional Integration*: defines the functional integration requirements associated with integrating the new capability into the operational environment.
- *Human Integration*: defines the human integration requirements associated with integrating the new capability into the operational environment.
- *Security*: defines the physical, information systems, and personnel security requirements associated with integrating the new capability into the operational environment.
- *In-Service Support*: defines the maintenance and support requirements for the new capability once in-service.
- *Test and Evaluation*: defines the test and evaluation requirements, including mandatory evaluations of safety, environmental, and energy conservation issues prior to joint acceptance and inspection. It also specifies whether independent operational test and evaluation is required.
- *Implementation and Transition*: defines requirements related to transitioning from the current capability to the new capability so as to not disrupt services, including requirements that encompass implementation planning, pre-installation checkout,

installation and checkout, site integration, system shakedown, dual operations, and the removal/disposal of replaced systems, equipment, land, facilities, and other items.

- *Quality Assurance*: defines quality assurance requirements, including contractor status reporting, an in-plant Quality Reliability Officer, independent verification and validation, vendor quality assurance plans, or a documented process for software development.
- *Configuration Management*: defines configuration management requirements for hardware, software, data, documentation, interfaces, and tools.
- *In-Service Management*: defines requirements for monitoring, assessing, and optimizing the performance of this capability during the in-service management phase of the acquisition management lifecycle.
- *System Safety Management*: defines requirements for an appropriate system safety program over the lifecycle of the investment program including system safety analyses and risk-resolution and tracking processes.

An example/product guide for the Program-level Requirement Document can be found online at http://nasea.faa.gov/foryou/archdev/main under the "Tools and Resources" section or on the FAA's Acquisition Management System (AMS) site http://fast.faa.gov/RequirementsManagement.cfm?CFID=2257372&CFTOKEN=14306477&p_title=AcquisitionPractices.

3.1.2 Product Integration

3.1.2.1 Vertical Integration

- Program-level requirements must align to the lowest level of the Enterprise-level NAS RD. data elements at the Enterprise-level, and vice versa.

3.1.2.2 Horizontal Integration (Inter)

- The requirements defined in the Program-level requirement document must be consistent with requirements referenced by interrelated Program-level requirement documents to eliminate unnecessary duplication. If a Program imposes a requirement upon another Program in their requirement document, that particular requirement must be managed via a Service Level Agreement (SLA) between the two program offices. Through the SLA, the program office imposing the requirement will include the requirement in their documentation, while the program office receiving the requirement will have a process in place for managing the assigned requirement.

3.1.2.3 Horizontal Integration (Intra)

- Program-level requirements must be unique across the PRD.
- Program-level requirements must be aligned to the Program's Functional Analysis results.
- Operational requirements should correspond to the operational activities identified in the OV-5.
- Interface requirements should correspond to the system interfaces specified in SV-1.

- Functional requirements should correspond to the system functions specified in the SV-4.
- Information requirements should correspond to the data exchanges specified in the SV-6.
- Performance requirements should correspond to the systems required performance specified in the SV-7.