HOPEX NAF

User Guide

HOPEX Aquila 6.2

MEGA
a Bizzdesign company

CONTENTS

Contents

INTRODUCTION TO HOPEX NAF

The NATO Architecture Framework (NAF) provides the rules, guidelines and product descriptions that define a standard way of developing, organizing and presenting enterprise architecture. It also provides guidelines on how to describe communication and information systems. When this framework is applied, architectures are able to effectively contribute to the acquisition and fielding of cost-effective and inter-operable military capabilities. The framework ensures that architectures developed by NATO and its member countries can be compared and related across NATO and National boundaries.

A NAF implementation tool should be able to generate standardized documents that deal with enterprise architectures and that support the search for sponsorship and funding of major mission-oriented systems.

HOPEX NAF is designed to generate NAF deliverables. It is based on the standard features of **HOPEX**, however, dedicated features have been introduced to guide the NAF expert to use the product with the vocabulary with which he/she is accustomed.

The framework details all the views addressing the different aspects of enterprise architectures:

- ✓ (the overall purpose,
- ✓ the operational perspective,
- ✓ the system perspective
- ✓ the technical perspective).

☛ ***HOPEX NAF** is based on the NAF v.3 release.*

HOPEX is designed to facilitate modeling and analyzing enterprise architectures.

Architecture Composition

The issues dealt with in architecture modeling range from business processes to technical infrastructure via systems and applications. Architecture modeling is a very broad subject and modeling a huge architecture which is composed of many business processes can be hard work. Splitting the task into smaller architectures can help to achieve the modeling effort. Adopting this solution means that a set of sub-architectures must be contained in the same repository. These architectures are then reused to compose bigger ones. To facilitate this, the **HOPEX NAF** supplies a referencing service that allows users to integrate architectures into a new one.

Evolution Traceability

NAF has several subviews that address the issue of architecture evolution (NSV-8, NSV-9, NTV- 2). These subviews help to determine the plans for future architecture changes. However, once the plan has been implemented, the architecture itself is in a new state that may be in line with the plan but which may also have differences due to unpredictable situations. In this case, the actual evolution of an architecture can be modeled via several architectures. Each of these architectures matches the actual state of the architecture at a given time and contains its own plans for the future.

Architecture Comparison

One of the main reasons behind having a Standard that specifies the format of enterprise architecture deliverables is that this facilitates the comparison of several architectures. All architectures display their properties and definitions in the same way. This enables readers to compare and analyze the differences in architectures and select the architecture that best corresponds to their needs.

THE HOPEX IMPLEMENTATION OF NAF

This section deals with **HOPEX** implementation of NAF v.3. The aim is to present the main principles that govern this implementation and guide the user in his/her use of **HOPEX** tool to create NAF deliverables.

- ✓ The NAF Views
- ✓ The NAF Subviews
- ✓ Pre-Requisites to using HOPEX NAF
- ✓ Managing Constraints, Contents and Requirements

THE NAF VIEWS

NAF is composed of a set of deliverables, called views, which address different parts of an enterprise architecture. These views are grouped into subviews. Subviews that focus on the same perspective are placed in the same view.

NAF organizes architectures into seven different views.

- All View (NAV)
- Capability View (NCV)
- Operational View (NOV)
- Service-Oriented View (NSOV)
- Systems View (NSV)
- Technical View (NTV)
- Program View (NPV)

Each view portrays certain architecture features. Some features span several views and provide integrity, coherence, and consistency to architecture descriptions.

Here are the definitions of these views as outlined in the NATO Architecture Framework volume 3 document.

NATO All View (NAV)

There are some overarching aspects of architecture that relate to all seven views.

These overarching aspects are captured in All-Views (NAV).

NATO Capability View (NCV)

The NATO Capability View (NCV) supports the process of analyzing and optimizing the delivery of military capabilities in line with NATO's strategic intent. The NCV achieves this by capturing essential elements of NATO's strategic vision and concepts and NATO's capability planning process, and decomposing this data into a capability taxonomy. The taxonomy is augmented with schedule data and measures of effectiveness to enable the analysis of capability gaps and overlaps. The NCV further details the dependencies between military capabilities, enabling capability options to be built in a more coherent manner and effective trade-offs to be conducted across NATO common funded programs.

NATO Operational View (NOV)

The NATO Operational View (NOV) is a description of the tasks and activities, operational elements, and information exchanges required to accomplish NATO

missions. NATO missions include both war-fighting missions and business processes. The NOV contains graphical and textual content that comprise an identification of the operational nodes and elements, assigned tasks and activities, and information flows required between nodes. It defines the types of information exchanged, the frequency of exchange, which tasks and activities are supported by the information exchanges, and the nature of information exchanges.

NATO Service-Oriented View (NSOV)

The NATO Service-Oriented View (NSOV) supports building architectures based on the concept of a Service-Oriented Architecture (SOA), which is fundamental to the NNEC paradigm. The NSOV is a description of services needed to directly support the operational domain as described in the NATO Operational View. A service, within the NSOV, is understood in its broadest sense, as a unit of work through which a provider provides a useful result to a consumer. NSOV focuses strictly on identifying and describing services. The view also supports the description of service taxonomies, service orchestrations, a mapping of services to operational activities, and a description of service behavior.

NATO Systems View (NSV)

The NATO Systems View (NSV) is a set of graphical and textual subviews that describes systems and system interconnections providing for, or supporting, NATO functions. NATO functions include both war-fighting and business functions. The NSV associates system resources to the NOV. These system resources support the operational activities and facilitate the exchange of information among operational nodes. Note that systems providing services can be pure technical systems as documented in the systems view only or a combination of technical and operational elements that is documented with a combination of an operational node and one or several associated system nodes. A logical system providing services is documented in the service-oriented view only.

NATO Technical View (NTV)

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

NATO Programme View (NPV)

Programme Views (NPV) describe the relationships between NATO capability requirements and the various programs and projects being implemented. They provide programmatic details and highlight the dependencies between capability management and the NATO acquisition process.

This information can be further leveraged to show the impact of acquisition decisions on the architecture.

THE NAF SUBVIEWS

Each of the seven views defined above groups subviews. In NAF vocabulary, a subview is either a graphical, textual or tabular deliverable that describes the characteristics that are relevant to the architecture product. Each subview is associated to a short name created from the abbreviation of the view to which it belongs (NAV, NCV, NOV, NSOV, NSV, NTV, NPV) and suffixed by a number.

The following tables list the subview names, coded names and a brief description of each of the different NAF subviews.

List of Subviews

NATO All View

Code	Subview Name	Comment
NAV-1	Overview and Summary Information	Architecture project identification, scope, purpose, viewpoint, context tools and file formats used, analytical findings

NATO Capability View Subviews

Code	Subview Name	Comment
NCV-1	Capability Vision	High-level graphical/textual description of operational concept
NCV-2	Capability Taxonomy	Structured list of capabilities required during a particular time-frame

Code	Subview Name	Comment
NCV-3	Capability Phasing	Representation of available capabilities at different points in time or during specific timeframes
NCV-4	Capability Dependencies	Capabilities, capability functions, describes the dependencies between capabilities, defines logical groupings of capabilities
NCV-6	Capability to Operational Activities Mapping	Describes the mapping between capability elements and the operational activities that those capabilities support

NATO Operational View Subviews

Code	Subview Name	Comment
NOV-1	High Level Operational Concept Descriptions	High-level graphical description of the operational environment of the architecture, in terms of operational elements involved, geographic regions, nodal connectivity, types of forces employed, etc., and its functionality.
NOV-2	Operational Node Connectivity Description	Operational nodes, connectivity, and information exchange needlines between nodes
NOV-3	Operational Information Requirements	Information exchanged between nodes and the relevant attributes of the exchange
NOV-4	Organizational Relationships Chart	Organizational context, role or other relationships among organizations
NOV-5	Operational Activity Model	Capabilities, operational activities/operational tasks, relationships among activities, inputs and outputs; additional data can show cost, performing nodes or other pertinent information

Code	Subview Name	Comment
NOV-6a	Operational Rule Model	One of three subviews used to describe operational activity. it identifies business rules that constrain operations
NOV-6b	Operational State Transition Description	One of three subviews used to describe operational activity - identifies how an operational node or activity responds to events
NOV-6c	Operational Event-Trace Description	One of three subviews used to describe operational activity - traces actions in a scenario or sequence of events
NOV-7	Information Model	Operational object, information, business rule, used to analyze the information aspect of the operational domain and guide the design of information systems

NATO System-Oriented View Subviews

Code	Subview Name	Comment
NSOV-1	Service Taxonomy	Organization and classification of services according to different criterion
NSOV-2	Service Definitions	Definitions of services: definitions could refer to service outcome, identification, properties, interfaces and policies
NSOV-4a	Service Constraints	Specification of constraints that apply to implementations of services
NSOV-4b	Service State Model	Specification of the possible states a service may have, and the possible transitions between those states
NSOV-4c	Service Interaction Specification	Specification of how a service interacts with external agents, and the sequence and dependencies of those interactions

NATO System View Subviews

Code	Subview Name	Comment
NSV-1	Systems Interface Description	Identification of systems and system connections and the information exchanges between them
NSV-2c	System Connectivity Clusters	Shows how individual connections between system ports are grouped into logical connections between nodes
NSV-4	System Functionality Description	Functions performed by systems and the system data flows among system functions
NSV-5	Systems Function to Operational Activities Traceability Matrix	Mapping of systems back to capabilities or operational activities
NSV-6	Systems Data Exchange Matrix	Provides details of system data elements being exchanged between systems and the attributes of this exchange
NSV-7	Systems Quality Requirement Description	Specifies the quality characteristics of systems, system hardware/software items, their interfaces and their functions as well as the current quality requirements and the expected or required quality requirements at specified times in the future.
NSV-8	Systems Configuration Management	Planned incremental step toward migrating a suite of systems to a more efficient suite, or toward evolving a current system to a future development of the architecture
NSV-9	Technology and Skills Forecast	Emerging technologies and software/hardware products that are expected to be available in a given set of time frames and that will affect future development of the architecture

Code	Subview Name	Comment
NSV-10a	Resources Constraints Specification	One of three subviews used to describe system functionality - identifies constraints that are imposed on the architecture or its systems under specified conditions.
NSV-10b	Resources State Transition Description	One of three subviews used to describe system functionality - identifies responses of a system to states, events and actions
NSV-10c	Resources Event-Trace Description	One of three subviews used to describe system functionality - identifies system-specific refinements of critical sequences of events described in the NATO Operational View
NSV-11a	Logical Data Model	Allows analysis of a system‘s data definition aspect, without consideration of implementation specific or product specific issues and provides a common dictionary of data definitions to consistently express subviews wherever logical-level data elements are included in the descriptions.
NSV-11b	Physical Data Model	Physical implementation of the Logical Data Model entities, for example, message formats, file structures, physical Schema

NATO Technical View Subviews

Code	Subview Name	Comment
NTV-1	Standards Profile	Listing of standards that apply to Systems and Services View elements in a given architecture and how they need to be, or have been, implemented.
NTV-2	Standards Forecast	Description of emerging, obsolete and fragile standards and potential impact on the architecture and its constituent elements
NTV-3	Standards Configurations	Description of standard configurations that apply to or emerge from the architecture effort or that are used or encountered in any of the subviews developed in the architecture effort.

NATO Programme View Subviews

Code	Subview Name	Comment
NPV-1	Programme Portfolio Relationships	Details the relationships among projects within a programme to show how these projects are grouped organizationally to form a coherent acquisition programme. It summarizes the interdependencies among projects and the linkages between project phases. It is also used to identify the level of maturity to be achieved at each stage of a programme‘s life-cycle
NPV-2	Programme to Capability Mapping	Mapping of programmes and projects to capabilities to show how the specific projects and programme elements help to achieve a NATO capability, as defined in a NATO capability package

NAF Subview Generation

According to the NAF standard, there are three kinds of deliverables: graphical, textual, tabular.

Take for example the NOV-4 subview: the NOV-4 subview graphically describes the hierarchy and relationships in the organization. This is graphically described in a **HOPEX** organizational chart.

PRE-REQUISITES TO USING HOPEX NAF

The **HOPEX NAF** contains a feature that allows users to create and manage alternative versions of objects and models. This is the variation feature.

Variants are used to describe objects and models differently from the reference model or object or to make updates to objects and models over time.

Variants inherit the elements of the object from which they derive. Inherited elements are displayed in the navigation tree with an inheritance arrow next to them.

To activate variations:

- Click the main menu and select **Settings > Options > HOPEX Solutions > Common Features > Activate variations**.

For more information on variations, see **HOPEX Common Features**, "Handling Repository Objects", "Object Variations".

☛ *In the **HOPEX NAF**, variants can be used for processes, more particularly, in NOV-5, NOV-6c and NSV-4. See:*

- *NOV-5 Operational Activity Model*,
- *NOV-6c Operational Event-Trace Description*
- *NSV-4 Systems Functionality Description*

MANAGING CONSTRAINTS, CONTENTS AND REQUIREMENTS

In the **HOPEX NAF** contents, requirements and constraints can either be operational or system items. When these objects are created from the corresponding subview folder, for example, the Operational Constraints folder in the NOV-6a Operational Rule Model subview, the constraint is automatically tagged as operational.

If constraints, contents or requirements are created in diagrams they are not automatically tagged as system or operational items. They may, however, have been linked to other objects in the repository. It is important to tag these objects to avoid any form of confusion and to enable their identification.

Tagging each of these objects one by one can be a tedious task. You can, however, simplify this task by tagging several objects simultaneously. This can be done by using the Query feature.

You can also use the Query feature when you want to change the Architecture View type of an object.

The best way to proceed is to find all the operational or system contents, requirements and constraints retrieved by linked objects.

The available Queries are:

- All Operational Contents retrieved by linked objects (EAF)
- All System Contents retrieved by linked objects (EAF)
- All Operational Constraints retrieved by linked objects (EAF)
- All System Constraints retrieved by linked objects (EAF)
- All Operational Requirements retrieved by linked objects (EAF)
- All System Requirements retrieved by linked objects (EAF)

For instance you want to find the contents linked to operational objects.

To tag objects as operational items:

1. Open the **Query** dialog box and find the **All Operational Contents retrieved by linked objects (EAF)** query.
2. Right-click the query and in the dialog box that appears select **Process**.
3. Specify the architecture on which the query is to be applied and click **OK**.
4. In the dialog box with the list of objects retrieved and from the **Show** menu select **Customize columns**.
5. Add the "NAF Architecture View Type" to the list of columns to be displayed and click **OK**.
   The "NAF Architecture View Type" column is displayed for the list of retrieved objects.
6. Select the objects you wish to tag and in the pop-up menu click **Modify**.
7. From the list proposed, select the desired option, for example "Operational Architecture View" if you want to tag the content objects as operational items and press **<Enter>** on your keyboard.
   The architecture View type for the selected objects has been modified.

OVERVIEW OF HOPEX NAF

HOPEX NAF comes with two different ergonomic tools to help the NAF expert use the **HOPEX** Modeling tool.

- ✓ Connecting to HOPEX NAF
- ✓ Discovering the NAF Navigation Tree
- ✓ Designing NAF Architectures

CONNECTING TO HOPEX NAF

To connect to HOPEX NAF, see Accessing HOPEX.

1. Use one of the following profiles:
   - NAF Architect
   - NAF Functional Administrator

DISCOVERING THE NAF NAVIGATION TREE

The NAF navigation tree contains all the architectures contained in the **HOPEX** repository.

Accessing the NAF navigation tree

To launch the **NAF** navigation tree:

- In the navigation bar, select **Architectures**.

The tree contains:

- an "Architectures" folder in which all the architectures of the current repository are located.

  ☛ *see Description of the Architecture Tree*

- a "NAF Dictionaries" folder which contains the official NAF terms and acronyms, if imported, approximately 50% of which are approved by NATO.

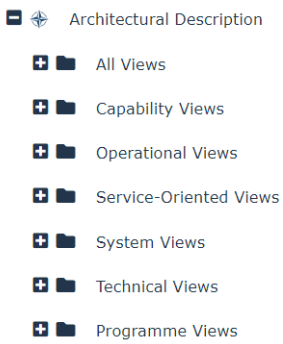
  ☛ *see Description of the Dictionary Tree*

Description of the Architecture Tree

☛ *See Accessing the NAF navigation tree.*

NAF architectures are found in the "Architectures" folder.

Each architecture displays folders which correspond to one of the seven NAF views:



These folders are automatically created with architectures. When expanded, each View folder reveals other folders which contain items that correspond to the subviews associated to the View described.

Description of the Dictionary Tree

While creating architectures, it is important that consensus be made about the terms used for any item defining the architecture.

To view the NAF terms and acronyms contained in the **HOPEX** dictionary:

- From the navigation bar, select **Architectures** and expand the **NAF Dictionaries** folder.

The terms and acronyms are arranged in alphabetical order. Any of these definitions can be attached to the items relating to a specific architecture.

DESIGNING NAF ARCHITECTURES

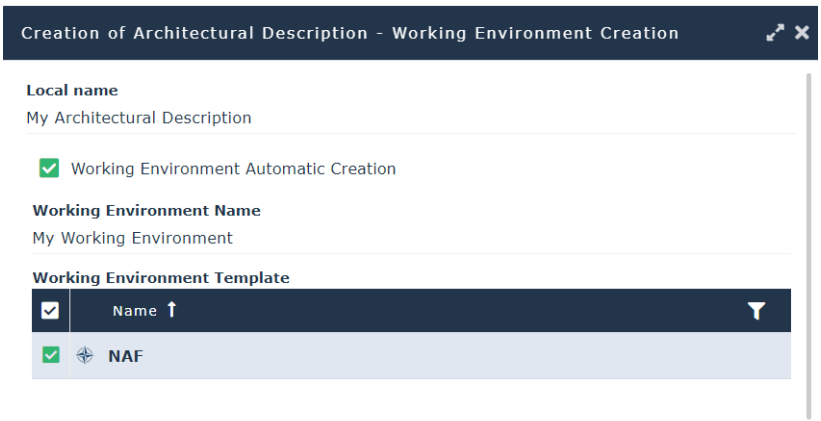
Creating a NAF Architecture

To create a NAF architecture:

1. From the NAF navigation tree, right-click the "Architectures" folder and select **New > Architectural description**.

   ☛ *See Accessing the NAF navigation tree.*

2. Click **Next**.
3. In the wizard that appears select the "NAF" working environment template.
4. Enter the **Working Environment Name.**



5. Click **OK**.

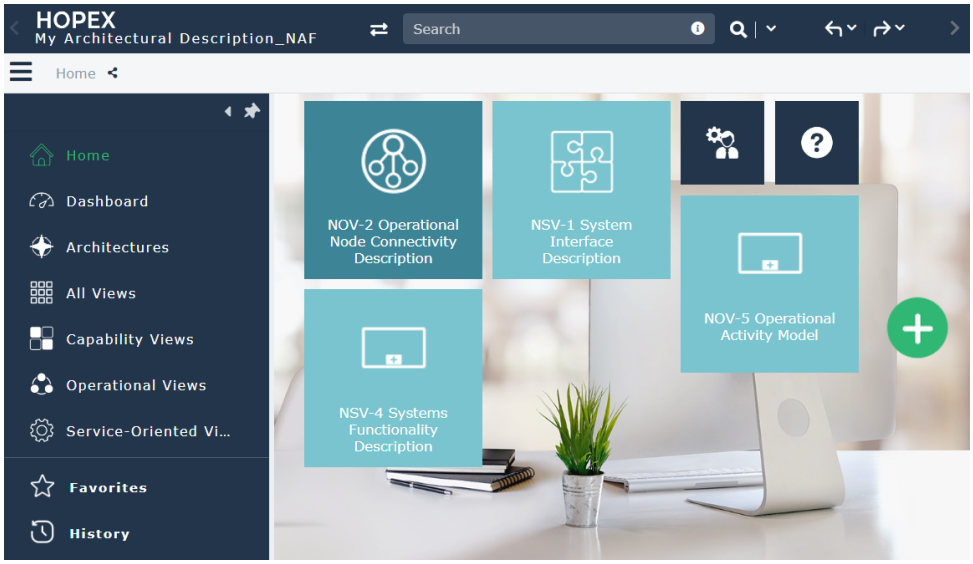
A working environment for the architecture is automatically created.

You can set an architecture as the default architecture. In so doing, all the objects created are created within the context of this architecture and are consequently contained in the working environment of the architecture. See Setting Up a Default Architecture.

Setting Up a Default Architecture

To set an architecture as the default architecture:

- In the main menu, select **Switch work environment** then the name of the architecture you created.
  The workspace now displays the different views of your architecture.



HOPEX NAF CONCEPT SUMMARY

Concepts Used in SubViews

Code	Main Concepts Used
All Views	
	Enterprise Phases Timelines
Capability Views	
NCV-1 Capability Vision	Master Plans Objectives
NCV-2 Capability Taxonomy2	Capability Maps Business Capabilities
NCV-3 Capability Phasing	Resource Architectures Master Plans
NCV-4 Capability Dependencies	Capability dependencies Capabilities
NCV-6 Capability to Operational Activities Mapping	Business Capabilities Functional activities/processes
Operational Views	
NOV-1 High-Level Operational Concept Description	Library Diagrams External References Notes

Code	Main Concepts Used
NOV-2 Operational Node Connectivity Description	Operational Nodes Interactions
NOV-3 Operational Information Requirements	(Operational) Contents
NOV-4 Organizational Relationships Chart	Org-Units Competences
NOV-5 Operational Activity Model	Functional Processes
NOV-6a Operational Rules Model	(Operational) Constraints (Operational) Requirements
NOV-6b Operational State Transition Description	(Operational) State Machines
NOV-6c Operational Event-Trace Description	Organizational Processes
NOV-7 Information Model	Data Models
Programme Views	
	Projects
Service-Oriented Views	
NSOV-1 Service Taxonomy	(Operational) Services
NSOV-2 Service Definitions	Service Definitions
NSOV-4a Service Constraints	(Service) Constraints (Service) Requirements
NSOV-4b Service State Model	(Service) State Machines
NSOV-4c Service Interaction Specification	(Service) Interaction scenarios
System Views	
NSV-1 System Interface Description	Applications Artifacts Resource Architectures
NSV-2c System Connectivity Clusters	Communication channel clusters
NSV-4 Systems Functionality Description	System Processes Systems Functions
NSV-6 Systems Data Exchange Matrix	(System) Contents

Code	Main Concepts Used
NSV-7 System Quality Requirements Description7	(Quality) Requirements
NSV-8 Systems Configuration Management8	(System) Master Plans
NSV-9 Technology and Skills Forecast	(Infrastructure) Master Plans
NSV-10a Resources Constraints Specificationsa	(System) Constraints (System) Requirements
NSV-10b Resources State Transition Descriptionb	(System) State Machines
NSV-10c Resources Event-Trace Description	(System) Interaction Scenarios
NSV-11a Logical data Model	Data Models
NSV-11b Physical Data Model	Databases
Technical Views	
NTV-1 - Standards Profile	Standards
NTV-2 Standards Forecast	Master Plans
NTV-3 - Standard Configurations	Standards
Programme Views	
	Projects

NAF Terminology

To bridge the gap between both worlds (**HOPEX** and NAF) and simplify the appropriation of the NAF application by a NAF expert, the **HOPEX** metamodel has been translated and the notions used renamed.

This section therefore details the mapping made between the **HOPEX** and NAF concepts used.

The table below lists the concepts renamed with their standard definition (from the **HOPEX** perspective) and the NAF definition. The aim of this renaming is to make

the mapping between the NAF concepts and the **HOPEX** concepts as invisible as possible for the user.

HOPEX Concept	NAF Concept	NAF Definition
Business Function	Operational Node	An operational node is a node that performs a role or a mission. A node is a representation of an element of architecture that produces, consumes or processes data.
Folder of Business Functions	Folder of Operational Nodes	Non-methodological grouping of operational nodes. This concept enables assembly under the same root of operational nodes around a common theme when this theme cannot be explained in the method.
Exchange contract (protocol)	Service Definition	

Definitions of the Main HOPEX Concepts

Artifact

An artifact is any element in the physical domain that is not a system or an organizational element (where organizational includes people).

An Artifact can represent a physical system, sub-system, platform, component or simply a physical item that has specific attributes.

Business Capability

A Business Capability is a set of features that can be made available by an enterprise.

Capability Configuration

A capability configuration consists of a Resource Architecture attached to a capability of the NAV architecture.

Constraint

A constraint is represented by a control or a business rule that must be applied during processing.

Content

Content designates the content of a message or a message flow, independently of its structure. Content may be used by several messages or message flows, since it is not associated with an sender and a destination.

Database

A database stores data physically or logically.

Functional Activity

A functional activity is an identifiable phase or step within a functional process.

Functional Process

A functional process is an end-to-end collection of functional activities that creates an outcome for a customer, who may be the ultimate customer or an internal end-user of the functional process.

Interaction

An Interaction represents a contract between entities in a specific context inside or outside a company. These entities can be organizations, activities, or processes. The content of this contract is described by an exchange contract.

Interaction Scenario

An interaction scenario lists all the communications that takes place within an operational node for a particular scenario (context).

Library

Libraries are collections of objects used to split HOPEX repository content into several independent parts. They allows virtual partition of the repository. In particular, objects owned by different libraries can have the same name.

Master Plan

A master plan is a long term outline of a project. It groups a set of consistent milestones and planned items temporally constrained against these milestones.

Needline

Needlines group together internal operational flows.

Objective

An objective is a goal that a company/ organization wants to achieve, or the target set for a business process or an operation. An objective allows you to highlight the features in a business process or operation that require improvement.

Operational Node

An operational node is a node that performs a role or a mission. It is a representation of an element of architecture that produces, consumes or processes data.

Project	Projects are implemented to produce required deliverables that enable reaching different states.
Request Point	Request points are used to interact with the node that is considered to be the consumer in the interaction.
Requirement	A requirement is a need or expectation explicitly expressed, imposed as a constraint to be respected within the context of a project. This project can be a certification project or an organizational project or an information system project.
Resource Architecture	A resource architecture is the combination of physical and organizational assets configured to supply a capability.
Sequence Flow	A sequence flow is used to show the order in which the steps of a function will be performed. It has only one source and one target.
Service	A service is a type of delivered functionality, specified independently of the capabilities that provide it. The different services are classified as operations, application services, information services
Service Definition	A Service definition transfers content between operational nodes.
Service Point	A service point is used to interact with the node that is considered to be the provider of the information exchanged.
Stage	An enterprise stage is a past, current or future stage of an enterprise plan.
Standard	A standard is a definition or format that has been approved by a recognized standards organization or is accepted as a de facto standard by the industry.
State Machine	A state machine is a set of states and transitions governing the state changes that can match any time-dependent object.

System Process

A system process is the executable representation of a process. The elements that formalizes a system process are the events of the workflow, the tasks to be carried out during the processing, the algorithmic elements used to specify the way in which the tasks follow on each other, the information flows exchanged with the participants.

In the BPMN notation, the system process represents a sub-process from the system point of view.

System Function

System functions are tasks performed by the IT system and are described in a system process.

Timeline

A timeline is used to create one or more calendars for the overall architecture, with dates and events that are common to everyone in the organization. These calendars can then be linked to Master Plans of the architecture to benefit from the timespots defined in the timelines.

Vision

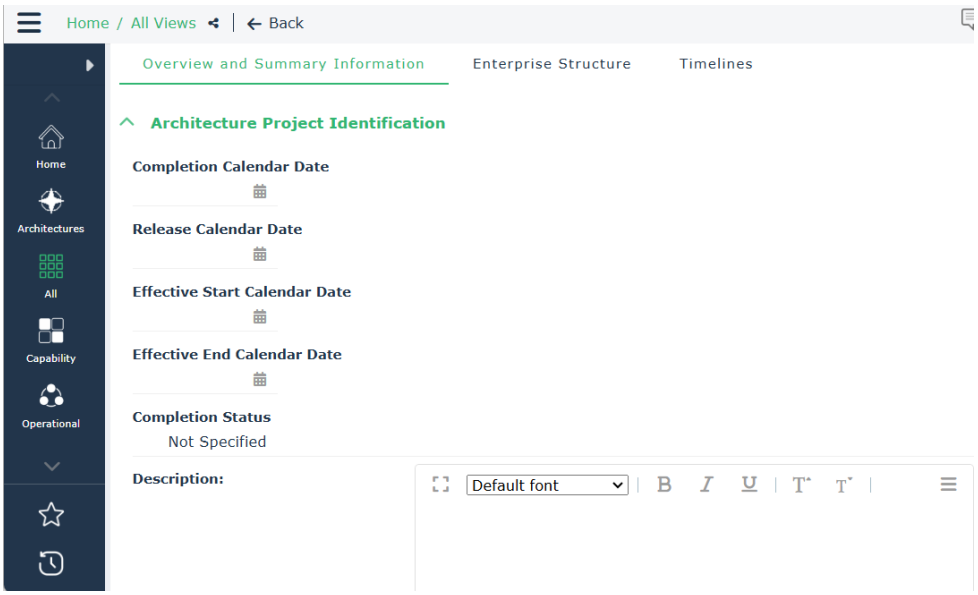
A Vision is the ultimate, possibly unattainable, state the enterprise would like to achieve. A Vision is often compound, rather than focused toward one particular aspect of the business problem. A Vision is supported or made operative by Missions. It is amplified by Goals.

NAF ALL VIEWS

NAV-1 Overview and Summary Information provides executive-level summary information in a consistent form that allows quick reference and comparison among architectures.

This subview includes an **Architecture Project Identification** card with:

- Calendar dates (completion, release, effective start, effective end)
- Completion status
- Description
- Architecture Constraint Summary



It also gives information on:

- Scope
- Purpose and viewpoint
- Context
- **Enterprise Structure** (enterprise phases attached to the architecture as enterprise items).
- **Timelines**.

> 🕮 *A timeline is used to create one or more calendars for the overall architecture, with dates and events that are common to everyone in the organization. These calendars can then be linked to Master Plans of the architecture to benefit from the timespots defined in the timelines.*
>
> ☛ *For more information on timelines, see the "Using Timelines" section of the **HOPEX IT Portfolio Management** documentation.*

NAF Capability Views Subviews

The NAF Capability Views subviews that can be generated in the **HOPEX NAF** are as follows:

- ✓ NCV-1 Capability Vision
- ✓ NCV-2 Capability Taxonomy
- ✓ NCV-3 Capability Phasing
- ✓ NCV-4 Capability Dependencies
- ✓ NCV-6 Capability to Operational Activities Mapping

NCV-1 Capability Vision

The purpose of an NCV-1 is to provide a strategic context for the capabilities described in the architecture. It also provides a high-level scope for the architecture which is more general than the scenario-based scope defined in NOV-1.

An NCV-1 Capability Vision subview begins with a description of the high-level concept. This concept will then be further augmented by describing the high-level operational goals and strategy in military capability terms.

☛ *As a general rule, the NCV-1 subview is not intended to specify:*

- *system requirements or user requirements. Instead, its role is to set the scope for the architecture in terms of future or current military capability vision.*
- *the success criteria for an architecture. NCV-2 provides metrics against each capability which may be used to successfully measure fielded capabilities.*

The main items dealt with in this subview are:

- the enterprise vision through master plans
- goals through objectives

☛ *These are located in the corresponding sub-folders of the NCV-1.*

Enterprise Vision

The enterprise vision describes the overall aims of an enterprise over a given period of time.

This is supported by a ***master plan*** in which the evolutions that help to achieve the overall objectives of the enterprise are described.

🕮 *A master plan is a long term outline of a project. It groups a set of consistent milestones and planned items temporally constrained against these milestones.*

The selected master plan is used to describe capability availability and show the associations between enterprise phases and capabilities.

Goals

Goals can be described through the ***objective*** notion. The objective items are attached to the enterprise vision (strategic master plan) as achieved objectives.

🕮 *An objective is a goal that a company/organization wants to achieve, or the target set for a business process or an operation. An objective allows you to highlight the features in a business process or operation that require improvement.*

NCV-2 Capability Taxonomy

The NCV-2 subview provides a structured list of the capabilities and sub-capabilities required for the vision established in the NCV-1 subview.

> 🕮 *A Business Capability is a set of features that can be made available by a system (an enterprise or an automated system).*

This list of capabilities is to be delivered for a particular timeframe. It has a hierarchical structure where capabilities are sometimes subdivided into sub capabilities and/or functions in order to provide clarity and the appropriate level of granularity required by subsequent processes in the capability management process.

NCV-3 Capability Phasing

The NCV-3 subview indicates the planned or available capabilities at different points in time or during specific time periods (states). This is capability phasing.

The different capability time periods (states) are indicated and are associated with milestones. This subview is created through an analysis of programmatic and deployment data in order to determine when system elements from NATO capabilities are to be deployed, upgraded and/or retired. This data may be provided in part by the Programme to Capability Mapping (NPV-2) subview.

The systems identified are structured according to the required capabilities determined in the Capability Taxonomy (NCV-2) subview and the associated timeframes. The projects that can deliver the capabilities within the slated time periods and states are also indicated.

The availability of capabilities at specific times and the deliverables expected can be demonstrated in Master plans, which in the context of information system planning, can be used to define the evolutions of the system that best respond to business function demands. The resource architectures/systems put in place to support the capabilities (capability configuration) are also demonstrated in Master plans. Capability configurations are created in the NSV-1 subview.

Master plans are expected to not only show planned elements at different successive levels and states, but also to provide information on functional suitability of evolutions related to business function expectations, and the impact of the of this Master plan implementation.

☛ *The Master Plans used for the NCV-3 Capability Phasing report are found in the NSV-8 subview. For more information, see NSV-8 Systems Configuration Management.*

NCV-4 Capability Dependencies

The NCV-4 subview shows the dependencies between capabilities (or capability functions) which are of interest to the architecture and groups these capabilities into logical groupings based on the need for these elements to be integrated. These groupings are used to inform the acquisition process and the Capability Phasing (NCV-3) subview.

Accessing the diagram that contains a dependency

To access the Capability Structure Diagram that contain a dependency:

1. From the navigation bar, select **Capability > NCV-4**.
2. Right-click the dependency whose diagram you want to access.
3. In the pop-up menu that appears, select **Diagrams Containing Object**.
   An miniature of a Capability structure diagram containing the dependency appears.
   A flashing square indicates the location of the dependency in the diagram.
4. Open the diagram by clicking the **Open Diagram** icon at the bottom of the image.
   The diagram opens in the desktop.

Creating dependencies

Dependencies are defined within the context of an upper capability. They are created in Capability Structure Diagrams and linked to capability compositions.

To create a dependency:

- Create or open an already existing capability structure diagram.

  ☛ *In the navigation tree, capabilities are found under the NCV-2 capability taxonomy folder.*

- In the object toolbar, select the dependency icon and draw the dependency from the source capability composition to the target capability composition.

  ☛ *The target capability composition is dependent on the source capability composition.*

Example of a capability structure diagram with a dependency

The name of the dependency is a combination of the name of the upper capability and those of the source and target capabilities of the dependency.

```
For example, in the diagram above, the name of the
dependency is "Monitoring [Detection -> Observation]".
```

It is, however, possible to have more than one source or target capability. This could be a case where a capability composition is included in several dependencies in the same structure diagram or where a composition is part of dependencies in different structure diagrams. Whatever the case, the naming rule is the same.

☛ *The generation of dependency names is automatic and these names cannot be changed by the user.*

NCV-6 Capability to Operational Activities Mapping

The NCV-6 Capability to Operational Activities Mapping subview describes the mappings that exist between capabilities and the operational activities that they support.

> 🕮 *A Business Capability is a set of features that can be made available by a system (an enterprise or an automated system).*

This subview is used to show which capabilities support which operational activities. It can also be used to show which capabilities are supported (or not), to reveal gaps in service provisions as well as to see if there are any redundancies, where a capability is supported by more than one operational activity.

NCV-6 Matrices

Here you can create a matrix to ***map Business capabilities to Operational Activities***.

The matrix displays:

- in columns, a subset of ***Business Capabilities***
- in rows, a subset of ***Operational Activities***

Select or unselect a cell to make the mapping between two objects.

NCV-6 Reports

Here you can create a report for a synthesis of the mapping of the ***Operational activities*** to the ***Business Capabilities***.

> ☛ *The report is in read-only mode (contrary to the matrix which is used to make the mapping).*

NAF Operational Views Subviews

The NAF Operational Views subviews that can be generated in the **HOPEX NAF** are as follows:

- ✓ NOV-1 High-Level Operational Concept Description
- ✓ NOV-2 Operational Node Connectivity Description
- ✓ NOV-3 Operational Information Requirements
- ✓ NOV-4 Organizational Relationships Chart
- ✓ NOV-5 Operational Activity Model
- ✓ NOV-6a Operational Rules Model
- ✓ NOV-6b Operational State Transition Description
- ✓ NOV-6c Operational Event-Trace Description
- ✓ NOV-7 Information Model

NOV-1 High-Level Operational Concept Description

NOV-1 is used to depict the "big picture" of the operational context of the architecture. It describes capability and highlights the main operational nodes (see NOV-2 definition) and interesting or unique aspects of operations. It provides a description of the interactions between the subject architecture and its environment, and between the architecture and external systems. A textual description accompanying the graphic is crucial. Graphics alone are not sufficient for capturing the necessary architecture data.

This subview addresses people wanting to have a general understanding of the architecture but who have no experience in modeling or architecture methods. The aim is therefore to be able to transfer the main purpose of the architecture without the need for a formal description. This is why this first level of architecture explanation uses external documents and notes and not formal models.

Another source of general information is the repository items cited in the texts of the architecture. Dropping any item in an object text automatically creates a link between the referencing object and the cited item. The list of all cited items can then be used as starting points to a deeper study of the architecture. This list only contains the cited objects and users cannot directly insert new items. It is therefore read-only.

NOV-2 OPERATIONAL NODE CONNECTIVITY DESCRIPTION

NOV-2 graphically depicts the operational nodes (or organizations) with needlines between the nodes that indicate a need to exchange information.

The graphic includes:

- internal operational nodes (internal to the architecture)
- external nodes.

NOV-2 is intended to track the need to exchange information from specific operational nodes (that play a key role in the architecture) to others. It does not depict the connectivity between nodes.

> 🕮 *An operational node is a node that performs a role or a mission. A node is a representation of an element of architecture that produces, consumes or processes data.*

Using Operational Nodes

To create an operational node:

1. In the NAF navigation tree, click **Operational Views > NOV-2**.
   The operational nodes available for the operational view appear in the **Root Operational Nodes** folder.
2. Right-click the Root Operational Node folder and select **Operational node**.
3. In the dialog box that appears enter the name of the operational node.
4. Click **OK**.

The new operational node appears.

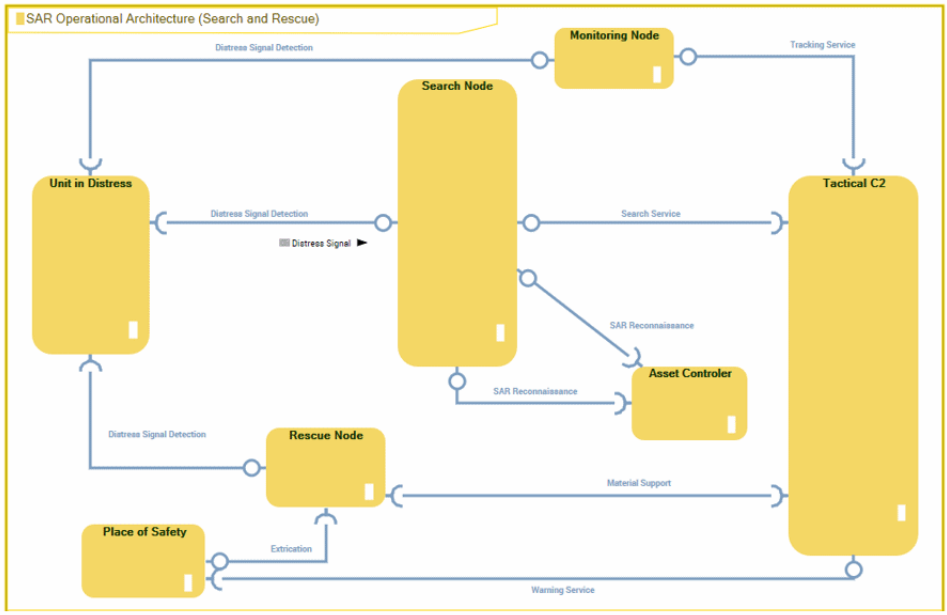
> ☛ *Operational nodes can also be initially created in the operational node structure diagram. See Creating Operational Node Structure Diagrams.*

Creating Operational Node Structure Diagrams

After creating your operational nodes the need for information exchange between operational nodes can be illustrated in the ***Operational Node Structure Diagram***. This information exchange is represented by interactions created between the nodes. The diagram shows how operational nodes and interactions interact with each other.

An Operational Node Structure Diagram details the structure of an operational node. The node is composed of sub-nodes that are connected to each other through interactions.

Below is an example of an operational node structure diagram. The operational nodes are represented by yellow rectangles.



Example of an Operational Node Structure Diagram (no content displayed)

The described node establishes a context for the interactions. For this reason sub-nodes are not directly connected in the diagram.

As sub-nodes can be reused in other contexts the interactions link the intermediate objects that reference the sub-nodes and that are defined locally within the context of the node. This way you can differentiate the interactions performed in one node context from those performed in another node context. In the case of operational nodes, the intermediate objects are called Operational Components. If no name is set for an operational component, a name is automatically created from the referenced operational node.

To create an operational node structure diagram:

1. In the NAF navigation tree click **Operational Views > NOV-2**.
2. Right-click the operational node concerned and select **New > Operational Node Structure Diagram.**
   The new diagram opens with a Root Operational Node positioned in it.

Adding operational nodes to the diagram

Operational nodes in the Structure diagram are referred to as operational components.

To add an operational component to the diagram:

1. Click the **Operational Component** icon in the object bar and click in the diagram.

2. In the **Add Operational node** dialog box that appears, select the operational node concerned from the drop-down list.

   ☛ *The name of the operational component is the name of the operational node that is used for the component.*

3. Repeat this step to create as many operational nodes as necessary.

Adding interactions to operational nodes

After creating your operational nodes you can display the interactions between them. Interactions describe the information that can be exchanged between two nodes.

📖 *An Interaction represents a contract between entities in a specific context inside or outside a company. These entities can be organizations, activities, or processes. The content of this contract is described in a protocol.*

☛ *All the structure diagrams (operational node, resource architecture or artifact assembly in NSV-1) are based on the same interaction principle. For more information, see NSV-1 System Interface Description.*

To add an interaction to operational nodes:

1. In the objects toolbar for a diagram, click **Interaction**
2. Click the entity requesting the service and draw a link to the entity providing the service.
3. In the add interaction dialog box, specify the exchange contract you wish to use.
4. Click **Add**.

Adding content to interactions

It is possible to add and display the content of interactions in the different structure diagrams.

Content of an interaction is described by an *exchange contract*.

📖 *An exchange contract is a model of a contract between organizational entities. This contract is described by exchanges between an initiator role and one or several contributor roles.*

In a service-oriented architecture, communication is based on *service points* and *request points*.

📖 *A service point is a point of exchange by which an agent offers a service to potential customers.*

📖 *A request point is a point of exchange by which an agent requests a service from potential suppliers.*

For more details, see Adding Service and Request Points to Operational Nodes.

Adding Service and Request Points to Operational Nodes

☛ *See Using Operational Nodes.*

Interaction with an operational node can be made through interaction points. Interaction points are ports for information exchanges. When a node is used as a sub-node in an operational structure, it can interact with another node through these interaction points.

In relation to the information in his possession, the designer of the structure can describe the interactions between two sub-nodes and specify the points of these sub-nodes that the interaction uses to exchange the information.

There are two kinds of interaction points:

- service point
- request point.

> ☛ *Please note that service and request points and interactions deal with the information that is exchanged and not the means by which the information is exchanged. The technical means by which information is exchanged is addressed in the NSV-2 subviews by way of communication ports.*

Service points

The service point is used to interact with the node that is considered to be the provider of the information exchanged. In this case, the interacting item at the other end of the interaction is the requester of the information that the node is able to supply.

Request points

Request points on the other hand are used for interacting with the node that is considered to be the consumer in the interaction.

Creating Realization of Operational Nodes

An operational node may implement a capability. To describe the capabilities implemented, you must define a *Realization* on the operation node in question.

To describe that a operational node is implementing a capability:

1. In the NAF navigation tree, click **Operational Views > NOV-2**.
2. Open the property page of the operational node that interests you.
3. Select the **Characteristics > Realizations** page.
4. In the **Composite Realization** section, click **New**.
   The selection window opens.
5. Select "Business Capability fulfillment" and click **OK**.
6. Chose the capability implemented.
7. Click **Add.**
   The capability realization appears in the properties page.

NOV-3 Operational Information Requirements

NOV-3:

- details information exchanges
- identifies who exchanges information,
- what information is exchanged,
- why the information is necessary,
- how the information exchange must occur.

There is no one-to-one mapping of NOV-3 information exchanges to NOV-2 needlines/interactions; rather, many individual information exchanges may be associated with one needline.

Information exchanges express the relationship across the three basic architecture data elements of an NOV subview (operational activities, operational nodes, and information flow) with a focus on the specific aspects of the information flow and the information content.

Certain aspects of the information exchange can be crucial to the operational mission and should be tracked as attributes in NOV-3.

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

The NOV-3 is about contents.

> 🕮 *A content designates the content of a message flow or message, independently of its structure. A content may be used by several message flows or messages, since it is not associated with a sender or with a recipient.*

You may create contents in the NOV-3 folder before creating the structure diagram.

> ☛ *For more information about the usage of contents in a structure diagram, see Creating Operational Node Structure Diagrams.*

NOV-4 Organizational Relationships Chart

NOV-4 illustrates the command structure or relationships (as opposed to relationships with respect to a business process flow) among human roles, organizations, or organization types that are the key players in architectures.

This subview clarifies the various relationships that can exist between organizations and sub-organizations within the architecture as well as between internal and external organizations.

The root org-units of the organization are created from the NAF navigation tree.

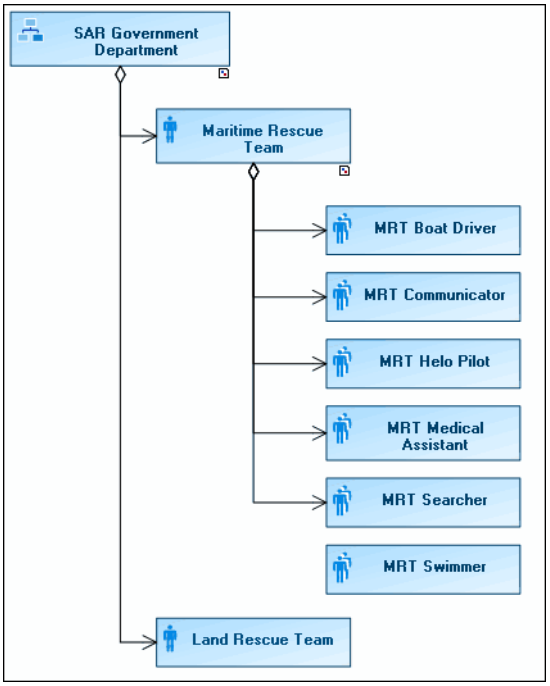
A global organization chart can be created to graphically show the hierarchy of org-units. If the hierarchy is too complex to be shown in a simple diagram, sub organizational charts can be added from any sub org-unit. Org-units are linked by dependency links that can be tuned to express a hierarchical or functional dependency.

To create an org-unit from the NAF navigation tree:

1. From the architecture concerned expand **Operational Views > NOV-4 - Organizational Relationship Chart**.
2. Right-click the **Root Org-Units** folder and select **New > Org-Unit**.
3. In the dialog-box that appears, enter the name of the Org-unit.
4. Select the **Org-Unit type**:
   - Company
   - Institute
   - Vendor
   - Agency
   - Organization
   - Role
   - Generic
   - Post

   The new Org-unit appears.

The figure below is an example of an organizational chart.



Example of Organizational-Unit Chart

NOV-5 Operational Activity Model

NOV-5 describes the operations that are normally conducted in the course of achieving a mission or an operational activity. It describes:

- capabilities,
- operational activities (or tasks),
- input and output (I/O) flows between activities,
- I/O flows to/from activities that are outside the scope of the architecture.

The NOV- 5 analysis enables the generation of different reports that assist in rapidly getting a synthesis of modeled information to aid in decision making. The displayed information can particularly help to:

- Clearly define and identify the responsible entities for the different operational activities, when coupled with the operational nodes of NOV-2
- Reveal unnecessary and redundant operational activities
- Make decisions about adding, merging and omitting operational activities
- Define and identify problems, opportunities and operational activities and their interactions that require close examination
- Provide a necessary foundation for depicting activity sequencing and timing in NOV-6 views
- Provide a clear picture of how operations are performed and thereby support the analysis and design of services and systems.

The entry point for the operational activity models is the ***functional process*** concept. This describes a sequence of activities.

A functional process is an end-to-end collection of functional activities that creates an outcome for a customer, who may be the ultimate customer or an internal end-user of the functional process.

To describe a functional process, you can use the Functional Process diagram. This functional process can then be broken down to reveal the different activities of the process.

The Functional Process diagram contains the activities that can be assigned to operational nodes. The details of an activity can then be further described in another Functional Process diagram. This is how a hierarchy of activities is to be generated.

To create the "Functional Process Diagram":

1. In the NAF navigation tree, expand **NOV-5 Operational Activity Model > Root Functional Processes**.
2. Right-click the functional process for which you wish to create the diagram and select **New > Diagram > Functional Process Diagram**. A diagram opens with a frame representing the functional process to be described.

Participants

You can place your participants inside as well as outside of the process on which the Functional Process diagram is based.

The participants placed on the outside of the process should not have activities placed on them.

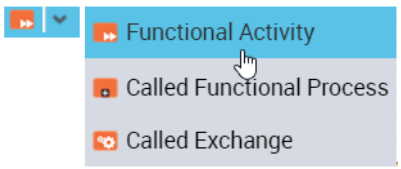
> ☛ *The purpose of the participant is not to describe activities but to help detail the external interactions of the process.*

Functional Activities

Activities are added to the participants in the diagram to indicate the steps in the process.

To add an activity to the functional process diagram:

1. In the objects bar select the **Functional Activity** icon and click on the participant responsible for the activity.



2. Enter the name of the functional activity and click **OK**.
   The new activity appears in the diagram.

> ☛ *If the activity represents another functional process, you can choose the "Called Functional Process" sub-menu from the Functional Activity button. Note the shape difference between activities with and without called processes.*

Events

When you add events to your diagram (Start End, etc.), you may choose to place them:

- inside the process
- on the frame of the Process described by the diagram.

> ☛ *If you place the event on the frame, a red rectangle appears in the event.*

This can be done through a drag and drop motion. Note that the event cannot be returned to the inside of the process with a drag and drop motion.

To remove the event from the frame and place it inside the process:

1. Right-click the event and select **Detach**.
2. Drag and drop the event into the frame.

Message flows

Messages flows are used to pass on information from one element to another.

Sequence flows

About sequence flows

The activities in a process are not all carried out simultaneously. Often the execution or completion of one activity triggers or leads to another activity.

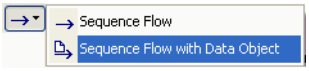
Sequence flows are used to indicate the steps to be followed (sequence of activities) in a process.

Creating sequence flows with content

If you so desire, you can combine the sequence of activities with the exchange of information between these activities.

To create a sequence flow with an information content between two activities:

1. In the objects menu bar click the arrow of the **Sequence Flow** icon and select **Sequence Flow with Data Object**.



2. Click on the activity where the flow begins and holding down the mouse button, draw a line to the activity to be linked.

   ☛ *Lines are drawn in the direction of the flow.*

3. Release the mouse button.

   ☛ *The sequence flow normally starts from an event, for example, Start.*

4. Create a new content or find an already existing one and link it to the sequence flow.
5. Click **OK**.
   A line with an arrow appears in the diagram between the two activities. The name of the attached content appears next to the line. The arrow indicates the direction of the sequence/message flow.

NOV-6A OPERATIONAL RULES MODEL

NOV-6a specifies the operational or business rules that are constraints to an enterprise, a mission, operation, business, or an architecture.

While other NOV subviews (NOV-1, NOV-2, and NOV-5) describe the structure of a business (what the business can do, for the most part) they do not describe what the business must do, or what it cannot do.

Rules are modeled in relation to the constraining objects. A constraint can be potentially linked to any object of the architecture. This is done within the different diagrams showing the objects to be constrained or by adding new constraints via the property pages.

Constraints retrieved from a deep exploration of the architecture are displayed in the NAF navigation tree. You can also create constraints from these locations. The constraints displayed are those attached to the operational items of the architecture.

NOV-6b Operational State Transition Description

NOV-6b is a graphical method used to describe how an operational node or activity responds to various events while changing its state. The associated diagram shows the sets of events to which the architecture will respond (by taking an action to move to a new state) as a function of its current state. Each transition specifies an event and an action.

HOPEX NAF is used to describe the specific behaviors (state machines) attached to activities. A behavior is a specific concept used to describe how an activity reacts to different events from start points to end points.

> ☛ *The terms "state machine" and "behavior" are used interchangeably in this section.*

Two methods exist for creating state machines:

- from an activity
- from the navigation tree.

Setting the state machine as operational

For a state machine to be set in the operational range, it must meet one of the following constraints:

- be connected to an activity
- be explicitly marked as operational using the Architecture View Type property.

> ☛ *If the property is set using the Operational Architecture View or the All Views value, the state machine is considered an operational one. Therefore, when the user creates a new state machine in the navigation tree, the property is set to Operational Architecture Views and states described in the state machine must relate to an operational state machine.*

Other operational state machines can be created directly from an activity. In order to retrieve all the state machines, a sub-folder is added to the NOV-6b folder. New state machines can be created from this location.

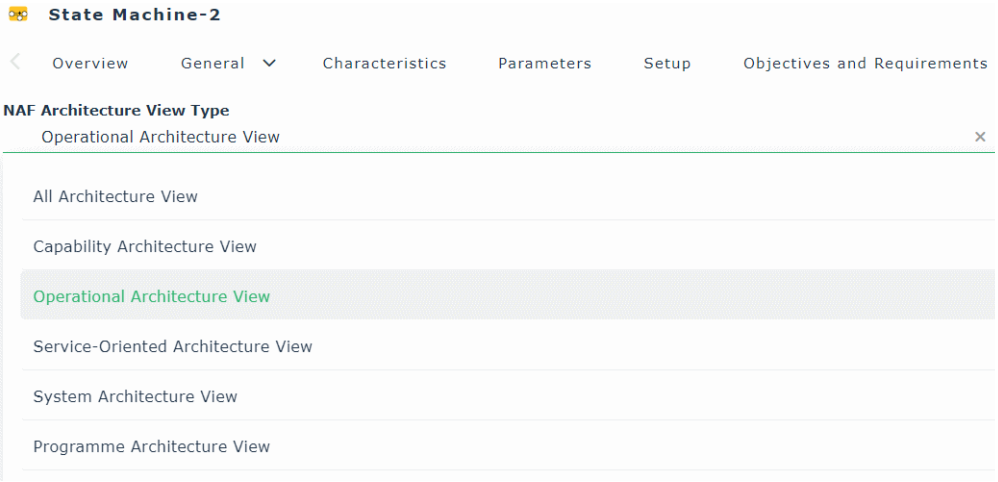
To create an operational architecture state machine:

1. In the NAF navigation tree, expand the **NOV-6b - Operational State transition Description** folder.
2. Right-click **Operational State Machines** and select **New > State Machine**.
3. In the dialog box that appears, enter the name of the state machine and click **OK**.
   The new state machine is displayed.

To change the property of the behavior:

1. Open the property page of the state machine
2. Select the **NAF > NAF State Level** page.

3. Select a view type from the **NAF Architecture View** Type field.



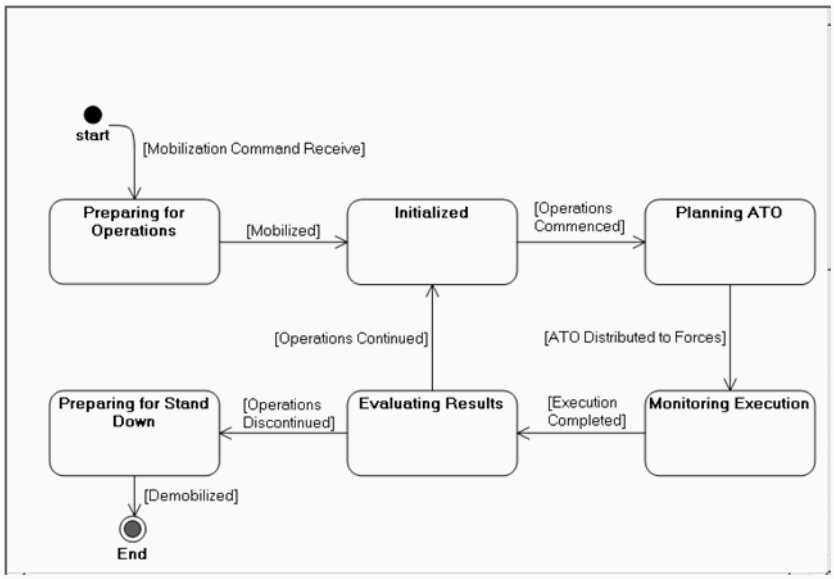
☛ *If you select "All Architecture View", the behavior also appears in the NSV-10b folder.*

Creating a State Diagram

Once a state machine has been created it can be described using state diagrams.

📖 *A state diagram describes the evolution over time of an object of a certain class, and shows its behavior in response to interactions with other objects (internal or external to the studied system) in its environment.*

The figure below is an illustration of a state diagram. Each arrow corresponds to a state and each graphical circle corresponds to the transition.



Example of a State Diagram (Conduct Joint Force Targeting States)

NOV-6c Operational Event-Trace Description

NOV-6c provides a time-ordered examination of the information exchanges between the participating operational nodes of a particular scenario. Each event-trace diagram should have an accompanying description that defines the particular scenario or situation.

> 🕮 *An interaction scenario lists all the communications that takes place within an operational node for a particular scenario (context).*

HOPEX NAF models missions and organizational processes via operations. These processes describe who does what to perform missions, regardless the service delivered to the customer.

The starting point of the description is a process.

> 🕮 *A process is a set of operations performed by org-units within a company or organization, to produce a result. It is depicted as a sequence of operations, controlled by events and conditions.*
>
> ☛ *For more information on processes properties, see the **HOPEX Business Process Analysis** documentation: Processes.*

To create an interaction scenario:

1. In the NAF navigation tree, expand the **Operational Views > NOV-6c Operational Event-Trace Description** folder.
2. Right-click the **Operational Interaction Scenarios** folder and select **New> Interaction Scenario**.
3. Enter a name and click **OK**.

To create an operational interaction scenario diagram:

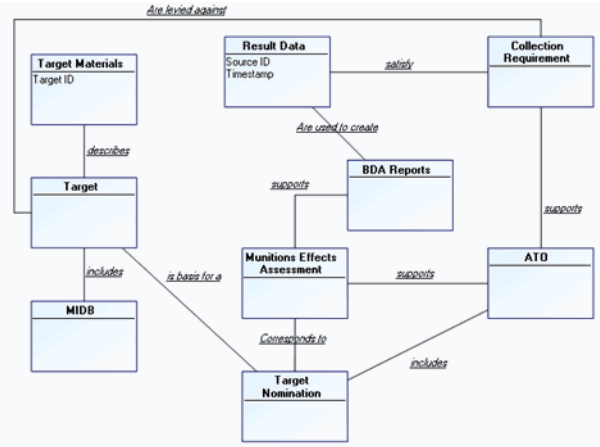
- Right-click the interaction scenario you have just created and select **New > Operational Interaction Scenario Diagram**.
  From here you can add:
  - operational nodes
  - operational node components
  - message instances

NOV-7 Information Model

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

HOPEX NAF can be used to design data models in NOV-7. The data model is a concept that can be graphically designed in Data model diagrams. The user creates new data models from the NAF navigation tree.

The data models are described by one or more data model diagrams, which contain entities with their attributes and the associations between the entities.



Example of a Data Diagram

The entities should correlate with the NOV-3 information elements. These elements are modeled with the **HOPEX** content objects and the content can be linked to a data model.

☛ *See NOV-3 Operational Information Requirements.*

In turn, NOV-7 should be used as an input to NSV-11, which captures logical and physical data models.

☛ *See NSV-11a Logical data Model, and NSV-11b Physical Data Model.*

NAF SERVICE-ORIENTED VIEWS SUBVIEWS

The NAF Service-Oriented Views subviews that can be generated in the **HOPEX NAF** are as follows:

- ✓ NSOV-1 Service Taxonomy
- ✓ NSOV-2 Service Definitions
- ✓ NSOV-4a Service Constraints
- ✓ NSOV-4b Service State Model
- ✓ NSOV-4c Service Interaction Specification

A service is a type of delivered functionality, specified independently of the capabilities that provide it.

☛ *A service may or may not have a physical effect on its environment.*

The different services are classified as:

- operations
- application services
- information services

☛ *See Services can be created from the "Root Services" and "All Services" folder of the NSOV-1 Service Taxonomy folder.*

NSOV-1 Service Taxonomy

The Service Taxonomy subview is designed to organize knowledge according to the service perspective, and to facilitate the harmonization of services across several domains (or several architectures).

A taxonomy is basically a system of classification. Its general purpose is for organizing one's knowledge of something into categories of similar things, in order to understand something better through comparison with other similar things. In the Service-Oriented View, the service taxonomy represents the operational domain's knowledge, as described in the Operational View, in terms of services, structured in some useful way. The definitions of the services are defined in NSOV-2.

☛ *See NSOV-2 Service Definitions.*

With **HOPEX NAF**, you can create services with categories.

You can also create structure and tree diagrams to decompose and represent the hierarchy of these services. Service points and service components can then be added to the diagrams.

Services can be created from the "Root Services" and "All Services" folder of the **NSOV-1 Service Taxonomy** folder.

Creating a service

To create a service:

1. Right click the Service folder concerned and select **New > Business Service**.
2. Name the Service and specify the category to which it belongs. You have the choice between:
   - "Application"
   - "Information"
   - "Operational".

   ☛ *In this sub-view you need to select "operational".*
3. Click **OK**.
   The new service appears in the folder of the Service from which it was created as well as in the folder that corresponds to the specified category.

   ☛ *All business services are also placed in the **All Services** folder.*

If no category has been specified for the service, it will be placed in the **Undefined** folder.

Creating a business service structure diagram

To facilitate the description process, a service can be decomposed into sub-parts which are assembled together in an interaction system. The different service layers can be graphically described in diagrams.

To do so, you can create the business Service Structure Diagram or the Business Service Tree Diagram and add the different service points required as well as the Business service Compositions.

☛ *The modeling principles are similar to those used for the operational view and service points are used to express the service exchange protocols to be fulfilled in order to benefit from the service.*

NSOV-2 Service Definitions

The NSOV-2 Service Definitions subview is intended to assist in delineating and defining services in order to understand the operational domain in terms of services supporting operational activities.

This subview provides a list of service definitions which correspond to elements captured in other subviews such as:<

- NOV-4 (users/ service consumers),
- NOV-7 (information objects),
- and NOV-3 (constraints on information exchanges).

Service definitions can be verified by checking if information needlines (NOV-2), information exchanges (NOV-3) and operational activities (NOV-5) are adequately supported.

Service definition includes, among other things, the definition of ports and interfaces, and connecting the interfaces to data models. The services in this subview are referenced in NSV-12.

The main entry point of the NSOV-2 view is the Service Definition folder. This concept matches the definition of the exchange protocol that must be fulfilled to benefit from the exposed services.

> 🕮 *An exchange protocol defines how a consumer and a provider must behave in order to carry out a particular service. These exchange protocols are connected to the service points. The service that displays this service point will play the role of the provider in the exchange protocol.*

Service definitions can be created:

- directly in the Service Definition folder,
- when creating interactions in a service structure diagram.

To create a service definition from the Service Definition folder:

1. In the NAF navigation tree, click **Service-Oriented Views > NSOV-2**.
2. Right-click the "All Service Definitions" folder and select **New > Service Definition**.
3. In the dialog box that appears enter the name of the service definition.
4. Click **OK**.

The new service definition appears.

NSOV-4a Service Constraints

Purpose

The purpose of the Service Constraints View (NSOV-4a) is to specifies constraints that apply to implementations of services.

Contents

The service constraints report is an alphabetical list of constraints and requirements defined at the service level.

> 🕮 *A constraint is represented by a control or a business rule that must be applied during processing.*
>
> 🕮 *A requirement is a need or expectation explicitly expressed, imposed as a constraint to be respected within the context of a project. This project can be a certification project or an organizational project or an information system project.*

NSOV-4B SERVICE STATE MODEL

Purpose

The purpose of the Service State Model View (NSOV-4b) is to specify the possible states a service may have, and the possible transitions between those states.

Parameter

- Architecture

Contents

The service state model report details the state machines defined at the service level.

📖 *A state machine is a set of states and the transitions governing the state changes that can match any time-dependent object.*

NSOV-4c Service Interaction Specification

The purpose of the Service Interaction Specification View (NSOV-4c) is to specify how a service interacts with external agents, and the sequence and dependencies of those interactions.

NAF System Views Subviews

The NAF System Views subviews that can be generated in the **HOPEX NAF** are the:

- ✓ NSV-1 System Interface Description
- ✓ NSV-2c System Connectivity Clusters
- ✓ NSV-4 Systems Functionality Description
- ✓ NSV-5 Systems Function to Operational Activity Matrix
- ✓ NSV-6 Systems Data Exchange Matrix
- ✓ NSV-7 System Quality Requirements Description
- ✓ NSV-8 Systems Configuration Management
- ✓ NSV-9 Technology and Skills Forecast
- ✓ NSV-10a Resources Constraints Specifications
- ✓ NSV-10b Resources State Transition Description
- ✓ NSV-10c Resources Event-Trace Description
- ✓ NSV-11a Logical data Model
- ✓ NSV-11b Physical Data Model

NSV-1 System Interface Description

The purpose of the System Interface Description is to illustrate which systems collaborate, and in what way they do so, to support the operational domain's information and information exchange needs as defined in the Operational View; most notably in NOV-2 and NOV-3.

NSV-1 links together the Operational Viewpoint and the System Viewpoint by depicting which systems and system connections realize which information exchanges. A system is defined as any organized assembly of resources and procedures united and regulated by interactions or interdependences to accomplish a set of specific functions. The term system in the System Viewpoint is used to denote software intensive systems (Federation of Systems (FoS), System of Systems (SoS), subsystems, and system components) and can include web services, network components and other hardware components, such as routers, satellites and network segments.

A system's services are accessed through the system's interfaces. Generally, an interface is a contract between the providers and consumers of (system) services. With software intensive systems, this contract is a declaration of a coherent set of public system functionalities. The system's interfaces specify the system's behaviour without specifying implementation aspects. An NSV-1 connection between system interfaces is the systems representation of an NOV-2 needline or NOV-3 information exchange. A single needline or information exchange may translate into multiple connections between system interfaces.

An NSV-1 documents:

- Systems and their interfaces
- System use dependencies between interfaces
- System collaborations (systems interacting with each other through their interfaces)
- Distributions of software systems to hardware systems
- Connections between hardware systems
- Patterns (optional); standard system collaborations that have been proven to be sound solutions to known problems).

Creating Capability Configurations

A capability configuration consists of a Resource Architecture attached to a capability of the NAV architecture.

This resource architecture, which is created in the NSV-1 subview, usually presents solutions for the operation of the architecture with different deliverables. These deliverables can include projects. Projects are defined in the NPV-1 subview.

To create a capability configuration:

1. Expand the **System Views > SV-1 System Interface Description > Resource Architectures > All Resource Architectures**.
2. Open the property page of a resource architecture.

3. In the **Characteristics > Characteristics** sub-page, **Configured Capability** section, use the **New** or **Connect** button to create or link the capability to be included in the capability configuration.
   The new capability configuration automatically appears in the NCV-3 Capability Phasing **Capability Configurations** folder. It takes the name of the architecture resource that supports the capability.

   ☺ *Capability configurations can also be created in the creation wizard during the creation of resource architectures.*

Linking a System Process to a Resource Architecture

To specify that a system process is performed by a specific resource architecture:

1. Expand the **System Views > NSV-1 System Interface Description > Resource Architectures > Root Resource Architectures** folder.
2. Right-click the desired resource architecture and click **New > Process Performance**.
3. In the creation box, click on the arrow on the far right of the **Performed Process** box and select **Connect**.

   ☛ *For more information on system processes, see NSV-4 Systems Functionality Description.*

Creating an Application Structure Diagram

☛ *An Application Structure Diagram displays graphically the first level components of an Application, the access points (service/request points) and the connections between the components.*

To create an application structure diagram:

1. In the NAF navigation tree, expand **System Views > NSV-1 System Interface Description > Applications**.
2. Right-click an application and select **New > Internal Architecture (Application Structure Diagram)**.

Before starting to draw your diagram, ***you need to display the sub-application view***.

To display sub-applications:

1. From the diagram toolbar, click the **View and Details** button.
2. In the window that appears, select the **Sub-applications** check-box.

NSV-2c System Connectivity Clusters

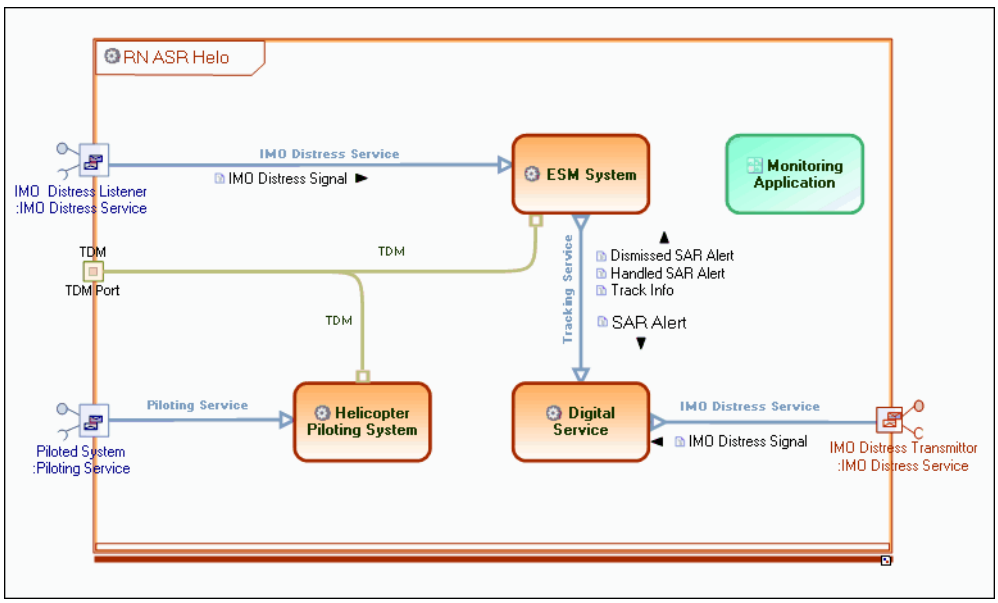
The purpose of the System Connectivity Clusters subview is to define the connectivity requirements between nodes. This subview is then used to estimate requirements for physical routing and bandwidth.

An NSV-2c subview provides a different viewpoint of information already specified in the NOV-2, NOV-3, NSV-1, NSV-2a and NSV-2b subviews.

The NSV-2c subview is useful when planning physical connections and routings between nodes. It is also intended to assist with the analysis of the connectivity between systems within or between nodes.

In **HOPEX NAF**, connectivity clusters are communication channels connected to a communication port that dispatches information to at least two sub-items.

In the example below, the RN ASR Helo system displays a TDM port that connects the ESM System and the Helicopter Piloting Systems.

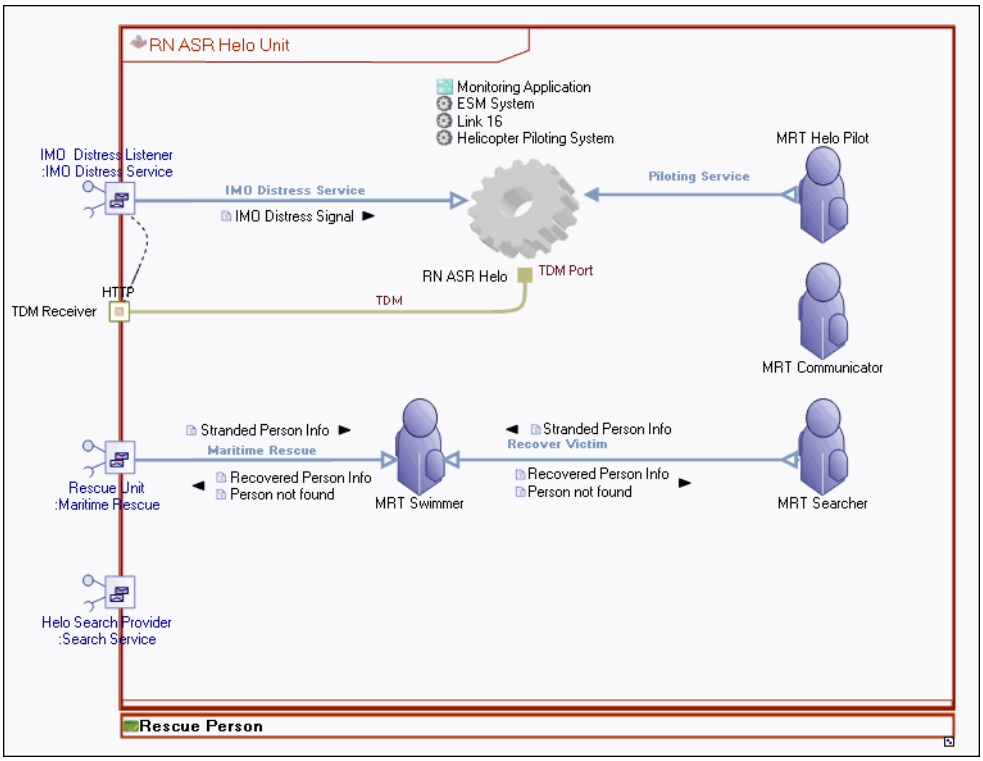


At the upper level, the RN ASR Helo system is used in the RN ASR Helo Unit system architecture.

The communication channel starting from this usage instance, sends all the TDM data to the TDM port. The data is then dispatched to either the ESM system or the

Helicopter Piloting System depending on the nature of the data.

The upper level communication channel is therefore considered a cluster of the two sub-systems.



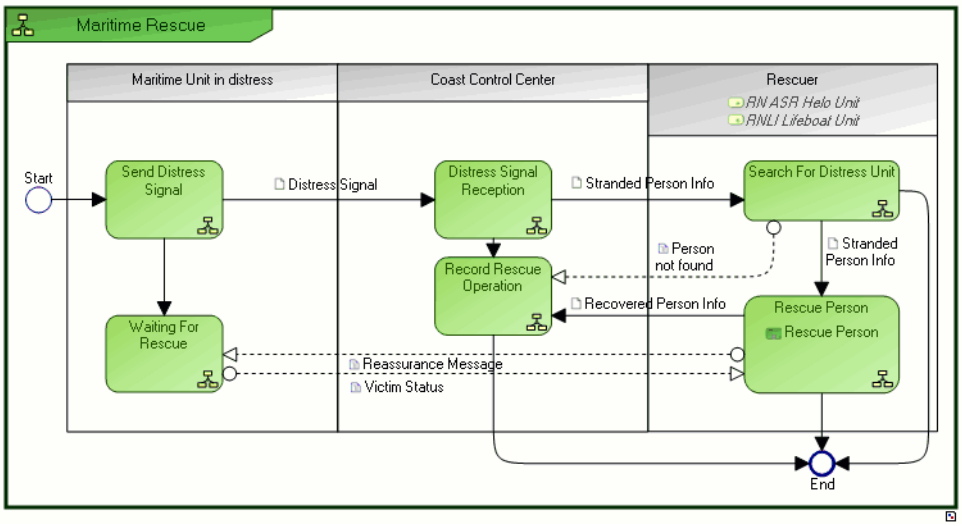
NSV-4 SYSTEMS FUNCTIONALITY DESCRIPTION

The NSV-4 subview documents system functional hierarchies and system functions, and the system data flows between them. While NSV-1 products describe the available interface between systems, they do not explain how these interfaces are used to exchange information or how data flows between the systems. The NSV-4 subview shows how data is transferred through system functions and binds these system functions to the system. Similarly, data transferred by functions can be described in a service perspective.

The **HOPEX NAF** models system functions and their flow of data via system processes.

A system process describes a set of system functions (tasks) performed by the IT system. The system functions are linked to each other by messages that transport data.

The figure below illustrates the "Maritime Rescue" system process which is involved in the performance of several tasks. These tasks are placed on the different participants which are more or less responsible for carrying them out. The different information exchanged between the tasks and the corresponding data flow is indicated in the diagram.



Example of a System Process Diagram

System processes are the system counterpart to operational processes. System process modeling is similar to NOV-5 and NOV-6a process modeling (BPMN-based).

☛ *For more information see NOV-5 Operational Activity Model and NOV-6a Operational Rules Model.*

System processes can be linked to resource architectures. This way you can specify that a system process is performed by a specific resource architecture.

☛ *For more information, see Linking a System Process to a Resource Architecture.*

NSV-5 Systems Function to Operational Activity Matrix

The NSV-5 Systems Function to Operational Activity Matrix represents the mapping of operational activities to system functions thus demonstrating how system functions support the conducting the operational activities.

Operational activities do not necessarily map one-to-one to system functions and as such NSV-5 forms an integral part of the eventual complete mapping from operational capabilities to systems. NSV-5 is an explicit link between the NATO Operational View and the NATO System View. The operational activities are drawn from NOV-5 while the system functions are drawn from NSV-4. The relationship between operational activities and system functions can also be many-to-many where one operational activity may relate to multiple system functions, and one system function may relate to multiple operational activities.

NSV-5 Matrices

Here you can create a matrix to ***map Operational Activities to System Tasks***.

The matrix displays:

- in columns, a subset of ***Operational Activities***
- in rows, a subset of ***System Tasks***

Select or unselect a cell to make the mapping between two objects.

NSV-5 Reports

Here you can create a report for a synthesis of the mapping of the ***Operational activities*** to the ***System Functions*** of the current architecture.

☛ *The report is in read-only mode (contrary to the matrix which is used to make the mapping).*

NSV-6 Systems Data Exchange Matrix

NSV-6 specifies the characteristics of the system data exchanged between systems. This subview focuses on automated information exchanges (from NOV-3) that are implemented in systems. Non-automated information exchanges, such as verbal orders, are only depicted in NOV subviews.

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

NSV-6 relates to, and grows out of NOV-3. The operational characteristics for the NOV-3 information exchanges are replaced with the corresponding system data characteristics.

Performance attributes for the operational information exchanges are replaced by the actual system data exchange performance attributes for the automated portions of the information exchange.

NSV-7 System Quality Requirements Description

The System Quality Requirements Description (NSV-7) identifies the quality requirements considered crucial to the successful achievement of the mission goals assigned to systems. Very often these requirements are the deciding factors in acquisition and deployment decisions, and figure strongly in systems analyses and simulations done to support the acquisition decision processes and system design refinement.

NSV-7 therefore specifies the quality requirements of systems, system hardware and software items, their interfaces (system data carried by the interface and the communications link details that implement the interface), and their functions. It specifies the current quality requirements and those expected or required at specified times in the future. The quality requirement categories are selected by the architect and end user community. As the complete set of quality requirements may not be known at the early stages of architecture definition, it is to be expected that this subview will be updated throughout the system's specification, design, development, testing, and maybe even its deployment and operations life-cycle phases.

NSV-7 builds on other NSV subviews by specifying quality requirements for:

- systems and interfaces (defined in NSV-1),
- system ports and communications (defined in NSV-2),
- system functions (described in NSV-4),
- system data exchange attributes (defined in NSV-6),
- data definitions (defined in NSV-11).

If the future quality expectations are based on expected technology improvements, then the quality requirements and their time periods will be coordinated by using a Systems Technology Forecast (NSV-9).

If quality improvements are associated with an overall system evolution or migration plan, then the time periods in NSV-7 are coordinated with the milestones in the Systems Configuration Management subview (NSV-8).

Please note that the requirements created in the SV-7 subview are automatically considered as "quality requirements", the **Requirement nature** property being set to "Quality".

NSV-8 Systems Configuration Management

NSV-8 captures evolution plans that describe how the system, or the architecture in which the system is embedded, will evolve over a lengthy period of time.

To successfully understand the time evolutions, timeline milestones are very important. In Information system planning, master plans are used to define long term evolutions of system that will best respond to business function demands.

The **HOPEX NAF** allows the user to create master plans that describe system evolution. A master plan is composed of milestones where life periods of architecture items are constrained. The master plans are used to describe system evolution scenarios and depending on the analysis of these scenarios, the scenarios can either be elected, rejected or postponed.

An architecture contains one or more master plans. Each of them describes the evolution of different parts of the system. A specific master plan is used to describe the overall evolution of the system. The creation of this master plan is transparent to the user. It enables the creation of top level scenarios.

For further information on Master Plan modeling, see the **HOPEX Planning** documentation.

Creating a Solution Master Plan

Solution Master Plans are created in the navigation tree. The details of these Master Plans are then displayed as a graphical representation in a Gantt Chart.

To create a Solution Master Plan in the NAF navigation tree:

1. In the **NAF** navigation bar select **System Views** > **NSV-8 > All Solution Master Plans**.
2. Click **New**.
3. Enter the name of your Master Plan.
4. Select the planning mode of your Master Plan. By default the Master plan is date-oriented.
5. Click **OK**.

Displaying the Gantt Chart

The Gantt chart is automatically created with the master plan.

To view the Gantt chart of a master plan:

- In the master plan properties, select **Gantt Chart**.
  The Master Plan is initialized with the milestones that were created. Milestone arrangement is from left to right.

You can create and modify elements directly in the Gantt Chart.

To complete the Master plan, you can:

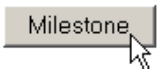
- Create milestones
- Create planned configuration capabilities
- Add the time periods of objects to be planned.
- Indicate the life-cycle status of the planned objects.

Creating milestones in a Gantt Chart

A milestone marks the date on which the results of a task are expected. Each milestone therefore corresponds to the beginning or end of availability of a resource architecture/system. Milestone arrangement is from left to right.

To create a milestone and add it to a Master plan from the Gantt Chart:

1. Click the **Milestone** button at the top of the Gantt Chart.



2. In the dialog box that appears, enter the name of the milestone.
3. Indicate the type of milestone to be created. In the case of a standard milestone, select the preceding milestone.
4. Click **OK**.
   The Milestone is created and appears in the Master Plan. This Milestone is also displayed in the Properties dialog box of the Master Plan, in the **Characteristics** tab.

   ☛ *Milestones belong to a particular master plan and cannot be reused in others.*

   If you made an error and wish to delete the milestone, click the cross on the milestone.

   ☛ *You can change the name and dates of a milestone in its **Properties** dialog box, **Characteristics** tab. Do not forget to refresh the Gantt chart to display the modifications.*

Adding time periods to a master plan and planning object life

The Gantt chart of a master plan relates to certain object types.

Adding an object to a master plan means integrating planning of the different object life cycle steps in the Gantt chart of the master plan. You can therefore plan all the steps in the life of the object.

To add time periods:

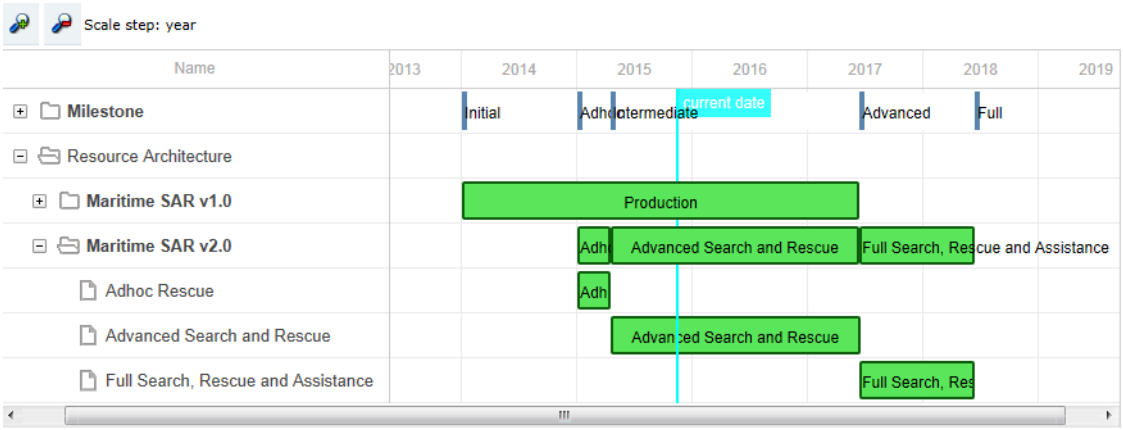
1. Above the master plan table, click the **Add Time Period** button.
2. In the **Query** dialog box that appears, use the **Find** button to find the required object, in this case a resource architecture.
3. In the list proposed, select your object and click **OK**.
   The dialog box for adding planned objects opens with the chosen object.
4. Click **Next** to continue.
   You are then required to select the life cycle of the object.
5. In the **Life Cycle to Follow**, select a life cycle, for example "Application development life cycle".
6. Specify the **Begin Date** and the **End Date**.

7. Click **OK**.

Application life planning appears in the master plan.

You can define absolute start and/or end dates on the time period.

When completed, the Gantt Chart displays the milestones and planned objects.



Example of an NSV-8 Gantt Chart

NSV-9 TECHNOLOGY AND SKILLS FORECAST

The NSV-9 defines the underlying current and expected supporting technologies that have been targeted using standard forecasting methods. Expected supporting technologies are those that can be reasonably forecasted given the current state of technology and expected improvements.

New technologies should be tied to specific time periods, which can correlate with the time periods used in NSV-8 milestones.

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

NSV-10A RESOURCES CONSTRAINTS SPECIFICATIONS

Systems rules are constraints on architectures, systems, system hardware/software items, and/or system functions. While other NSV subviews (NSV-1, NSV-2, NSV-4, NSV-11) describe the static structure of the System and Service Views (this is, what the systems can do), they do not describe, for the most part, what the systems must do, or what they cannot do.

At the system or system hardware/software item level, NSV-10a describes the rules which depict how the architecture or its systems should behave under specific conditions. At lower levels, it may consist of rules that specify the pre- and post-conditions of system functions. Such rules can be expressed in a textual form, for example, "If (this condition) exists, and (this event) occurs, (perform these actions)."

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

The NAF navigation tree shows the constraints grabbed during a deep exploration of the architecture. However, constraints cannot be created from these locations. The constraints displayed are those attached to system items of the architecture (system functions, system nodes, etc.).

NSV-10b Resources State Transition Description

The NSV-10b is a graphical method of describing a system (or system function) response to various events by changing its state.

The explicit time sequencing of system functions in response to external and internal events is not fully expressed in NSV-4. NSV-10b can be used to describe the explicit sequencing of the system functions. Alternatively, NSV-10b can be used to reflect explicit sequencing of the actions internal to a single system function, or the sequencing of system functions in relation to a specific system.

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

The **HOPEX NAF** can be used to describe specific state machines attached to tasks. A state machine is a specific concept used to describe how a task reacts to different events from the start points to the end points.

The system state machine diagram basically represents the sets of events to which the systems in the architecture will respond (by taking an action to move to a new state) as a function of its current state. Each transition specifies an event and an action.

NSV-10c Resources Event-Trace Description

Resources Event-Trace Descriptions (NSV-10c) are used for moving from the initial systems design to the next level of detail, to help define a sequence of system interactions, and to ensure that each participating system or human role has the necessary information it needs, at the right time, in order to perform its assigned functionality.

NSV-10c provides a time-ordered examination of system data elements exchanged between participating systems (external and internal) or human roles, as a result of a particular scenario or situation. Each particular scenario or situation may reflect system-specific aspects or refinements of critical sequences of events described in the NATO Operational View.

NSV-11A LOGICAL DATA MODEL

The purpose of data models is to enable the analysis, design and implementation of the data presentation, handling and storage functionality of an information system. A data model is the representation of an information model in a form that is specific to a particular paradigm or theory on the representation, storage and handling of data, often reflecting a certain type of data store or repository technology. Data models are often distinguished as logical or physical data models.

The Logical Data Model is a generalized formal structure in computer science. As such it directly reflects the paradigm or theory oriented mapping from the information model to the data model.

The Logical Data Model enables the analysis of a system's data definition aspect, without any consideration of implementation specific or product specific issues. It is also used to provide a common dictionary of data definitions to consistently express subviews wherever logical-level data elements are included in the descriptions.

NSV-11B Physical Data Model

The Physical data model is used for the analysis of a system's data implementation aspect, with consideration for a specific product. They can also be used to:

- Provide as much detail as possible on the data elements exchanged between systems to reduce the risk of interoperability problems.
- Provide the data structures for use in the system design process, if necessary.
- Provide a common dictionary of data implementation elements (e.g. tables and records in a relational database schema) to consistently express subviews wherever physical-level data elements are included in the descriptions.

The physical data model specifies how the logical data model will be instantiated in a particular product. The most predominant of such products are the relational database management systems, such as those supplied by Oracle and Microsoft (in which case the physical data model is often called a 'database schema'). Object repository products also exist, but are less often encountered.

The essential elements of a physical data model (in the case of a relational database) are: tables, records and keys. In a true object-oriented data model, all data elements are expressed as objects; whether they are classes, instances, attributes, relationships, or events.

NAF TECHNICAL VIEWS SUBVIEWS

The NAF Technical Views subviews that can be generated in the **HOPEX NAF** are as follows:

- ✓ NTV-1 - Standards Profile
- ✓ NTV-2 Standards Forecast
- ✓ NTV-3 - Standard Configurations

NTV-1 - STANDARDS PROFILE

The Standards Profile (NTV-1) provides a list of standards that guide and constrain the implementation of systems as defined in the various subviews of the NATO System View.

In **HOPEX NAF**, a standard is a specific independent notion that can be created in order to establish a list. The standard notion encompasses all the technical and non technical items that are recommended by the company for designing the architecture.

A standard can be linked to the definition of repository items such as applications, databases, artifacts, resource architectures and communication protocols. This link, even though non mandatory, is useful in helping to provide a more accurate definition of the standard.

> For example, if a computing engineer process states that it is mandatory to use UML for application design, then UML can be defined in the tool as a standard. This standard will not, however, be linked to any object of the repository.

The sections below indicate how standards can be described. For more information on the metamodel behind the standard notion, see NTV-1 Standards Profile and NTV-2 Standards Forecast.

Standard Decomposition

A standard can be decomposed into sub-parts. Each part is called a Standard Component. A standard component can reference another existing standard, in which case the name of the component can be automatically created from the name of the standard on which it is based (this is not mandatory and the user can rename the component).

For example, the DNS standard is defined from different smaller standards:

- DNS
  - IETF STD 13:1987
  - RFC 1034:1987
  - RFC 1035:1987

Standard and Approving Organizations

A standard is approved/created by an organization. For example, HTML is created by the W3C and UML is created by the OMG.

Standard Usability

Declaring a list of standards can be useful, however, comparing this list to the standard that is used to implement a resource architecture is more beneficial. This comparison can be made if the components of the architecture detail the standards

on which they are based. To make the comparison, a special link exists between architecture items and standards.

> ☛ *An application, for example, can be cited as a defining item for a standard while another application can be based on this standard.*

```
The Open Office tool can be cited as the standard
documentation tool while a business application used to
generate meeting reports can be based on this standard.
```

NTV-2 Standards Forecast

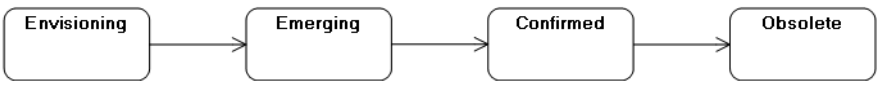
The purpose of the Standards Forecast subview (NTV-2) is to:

- identify emerging, obsolete and fragile standards,
- to assess their impact on the architecture and its constituent elements. A forecast that addresses emerging standards gives insight into the direction that the project will take.

In **HOPEX NAF**, the standards forecast gives the details of a specific type of master plan: the ***standard master plan***.

When a master plan is created from the NTV-2 navigation tree, this master plan automatically has the "Standard" type checked. For other kinds of master plans the user can check other types, however, the "Standard" type remains checked.

The standards defined in the NTV-1 subview can then be planned in the different states. **HOPEX** provides a default state and for standards. The following states are normally appropriate for standards:



Similarly to the "preparation", "production", and "retirement" states that can be linked, the states above are linked to equivalent stereotypes that classify them:

- **Envisioning**, **Emerging**: the standard is not available in a stable state for users but work is being done to achieve this stable state. The applied Stereotype is "Preparation".
- **Confirmed**: the standard is in a mature state and can be confidently used by the users. The applied Stereotype is "Production".
- **Obsolete**: the standard is no longer available and the user should consider another standard or a new release of the standard. The applied Stereotype is "Retirement".

Milestones and Time Periods

Business milestones are usually useless in the description of standard forecasts as they are not developed by the designer team but supplied by external organizations (for example NAF is defined by the NATO and HTML is defined by the W3C).

So, even though milestones can still be defined in standard master plans, it is recommended that you only have a few of them representing the very big steps in the standard evolution and that you set the start and end dates on the periods to match the different states of the standard.

Customization

Similarly to any time-dependent item, a specific state machine can be created with a set of customer-defined states.

This machine can then be attached to the standard metaclass if it applies to any standard (verify that the designer has the appropriate rights to see the metaclass) or directly to a specific standard.

If defining a new set of states, check that all the defined states are classified via the three stereotypes:

- "Preparation",
- "Production"
- "Retirement".

> ☛ *The **Setup** tab of a behavior item allows you to set the metaclasses that can be associated to the states.*

NTV-3 - Standard Configurations

The purpose of the Standard Configurations Description (NTV-3) is to describe standard configurations that are applied to or emerge from the architecture effort, used or encountered in any of the subviews developed in the architecture effort.

NTV-3 is intended to capture and explicitly describe configurations that are of value to the ongoing or to future architecture projects. It is also the intention of this subview to provide a single point to address and promote the use of standard configurations.

NAF Programme Views Subviews

The NAF Programme Views subview that can be generated in the **HOPEX NAF** is: NPV-1 - Programme Portfolio Relationships.

The purpose of the Programme Portfolio Relationships subview (NPV-1) is to provide details on the relationships among ***Projects*** within major NATO programmes. It shows how projects are grouped to form a coherent acquisition programme and summarizes the interdependencies among projects and the links between project phases. It is also used to identify the level of maturity to be achieved at each stage of a programme's life-cycle.

Projects are implemented to produce required deliverables that enable reaching different states.

Creating Projects

To create a project:

1. In the NAF navigation tree, expand the **Programme Views**.
2. In the **Root Projects** folder, create a project and click **OK**.

Creating Project Composition Diagrams

To create a project composition diagram:

1. Right-click a project and select **New > Diagram > Project Composition Diagram**.
2. Click **OK**.
3. In the Insert toolbar, click the **Project** button and click in the project frame to create several projects.
4. Create **Project Dependencies** between the created projects if needed.
5. Create the **Deliverables** to be produced by your project.

Deliverables can be attached to time periods to show at what stage during the entire architecture project they should be produced. To do so, use the property pages of the deliverable.