HOPEX NAF

User Guide

HOPEX Aquila

MEGA
SEE THE BIGGER PICTURE

CONTENTS

NAF Technical Views Subviews 215

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.*

MODELING ARCHITECTURES IN A REPOSITORY

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 the All-Views (NAV) subviews. The NAV

subviews provide information pertinent to the entire architecture, but do not represent a distinct view of the architecture. NAV subviews set the scope and context of the architecture. The scope includes the subject area and time-frame for the architecture.

The setting in which the architecture exists comprises the interrelated conditions that compose the context for the architecture. These conditions include:

- Doctrine
- Tactics
- Techniques
- Procedures
- Relevant goals and vision statements
- Concepts of operations
- Scenarios
- Environmental conditions

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.

- ✓ NATO All View Subviews
- ✓ NATO Capability View Subviews
- ✓ NATO Operational View Subviews
- ✓ NATO System-Oriented View Subviews
- ✓ NATO System View Subviews
- ✓ NATO Technical View Subviews
- ✓ NATO Programme View Subviews
- ✓ NATO All View Subviews
- ✓ NATO Capability View Subviews
- ✓ NATO Operational View Subviews
- ✓ NATO System-Oriented View Subviews
- ✓ NATO System View Subviews
- ✓ NATO Technical View Subviews
- ✓ NATO Programme View Subviews

List of Subviews

NATO All View Subviews

Code	Subview Name	Comment
NAV-1	Overview and Summary Information	Architecture project identification, scope, purpose, viewpoint, context tools and file formats used, analytical findings
NAV-2	Integrated Dictionary	Architecture data repository with definitions of all terms used in all products
NAV-3a	Architecture Compliance Statement (Metadata)	Certifies that architecture satisfies all applicable and imposed criteria documented in the NAF to a required degree
NAV-3b	Metadata (Extensions)	Documents any deviations of the architecture's subviews from the standard subview guidelines of the NAF

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
NCV-3	Capability Phasing	Representation of available capabilities at different points in time or during specific timeframes

Code	Subview Name	Comment
NCV-4	Capability Dependencies	Capabilities, capability functions, describes the dependencies between capabilities, defines logical groupings of capabilities
NCV-5	Capability to Organizational Deployment Mapping	Capabilities, system connectivity, organizational structures, and programmatic information
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-3	Capability to Services Mapping	Mapping between the capabilities required by an Enterprise and services as defined for SOA
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

Code	Subview Name	Comment
NSOV-4c	Service Interaction Specification	Specification of how a service interacts with external agents, and the sequence and dependencies of those interactions
NSOV-5	Service Functionality	Specification of the mapping between services and the functional activities that those services support
NSOV-6	Service Composition	Specifies how services can be combined and sequenced to provide a higher level service.

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-2a	System Port Specification	Identifies the ports on each system and the protocols supported by each port
NSV-2b	System to System Port Connectivity	Identifies the connections between ports and shows the protocols and hardware specified for each connection. The ports may be on different systems.
NSV-2c	System Connectivity Clusters	Shows how individual connections between system ports are grouped into logical connections between nodes
NSV-2d	Systems Communication Quality Requirements	Specifies quality requirements applicable to communications between systems

Code	Subview Name	Comment
NSV-3	Resources-Resources Matrix	Describes relationships between systems in a given architecture; can be designed to show relationships of interest, for example, system-type interfaces, planned vs existing interfaces, etc.
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
NSV-12	Service Provision	Mapping of systems to services to identify which system contribute to the provision of which service.

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.

☛ *For more details on reports, see Using Reports.*

Take for example, the NAV-1, NOV-4 and NOV-3 subviews:

- The NAV-1 subview describes the scope and purpose of the architecture. It can be delivered as a textual document.
- The NOV-4 subview graphically describes the hierarchy and relationships in the organization. This is graphically described in a **HOPEX** organizational chart.
- Finally, in an NOV-3 subview, the expected result is a tabular representation of the exchange between operational nodes.

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 > Business Process and Architecture Modeling > 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

☛ *See also, Activating the Objective and Requirements Options.*

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
- ✓ Using Reports

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 menu, 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.

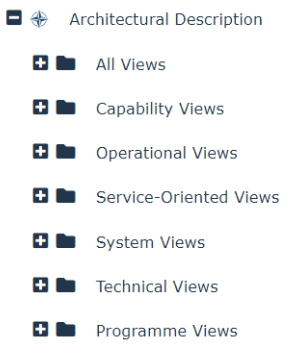
  ☛ *see Description of the Dictionary Tree*

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.

For example, in the "All Views" folder the NAV-1, NAV-2, NAV-3a and NAV-3b folders are displayed. The items of the different subview folders could include reports, documents, diagrams, etc.

Description of the Dictionary Tree

While creating architectures, it is extremely important that consensus be made about the terms used for any item defining the architecture. This is why the DoD created a set of official definitions for all common terms used in the defense sector.

HOPEX NAF is supplied with a **HOPEX** dictionary of the official NAF definitions. It contains more than 6,000 terms as well as over 6,000 acronyms and abbreviations, most of which have been approved by NATO and are therefore relevant in the context of NAF Projects with military orientation.

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

- From the navigation menu, 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

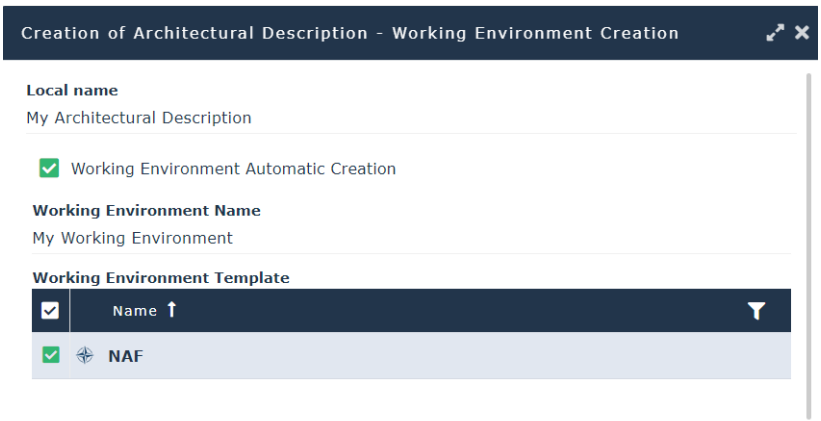
Creating a NAF Architecture

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**.

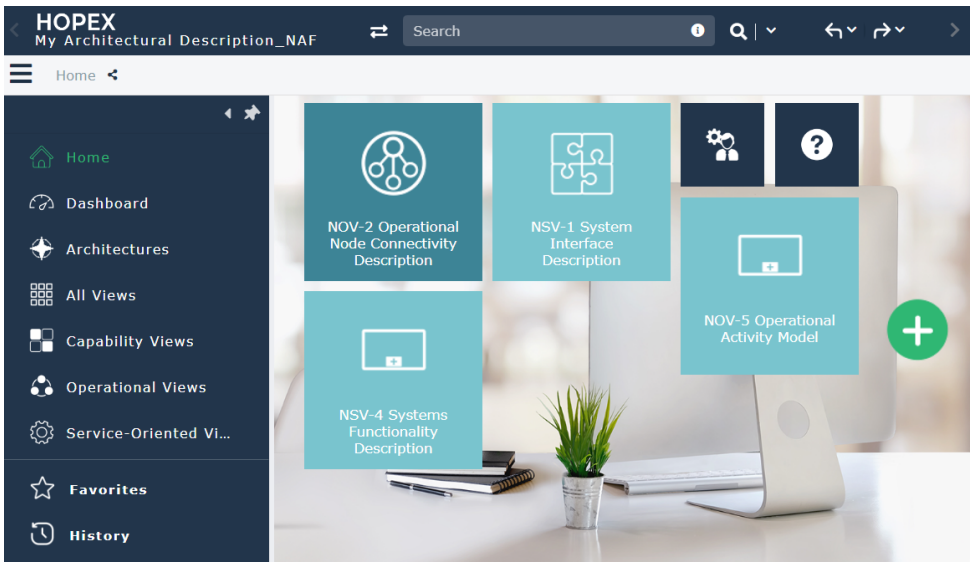
A working environment for the architecture is automatically created.

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.



USING REPORTS

The major purpose of the NAF framework is to generate documentation of subviews of a given architecture in a comparable format. This section details how you can generate and use documentation.

Using Reports to Describe the Architecture

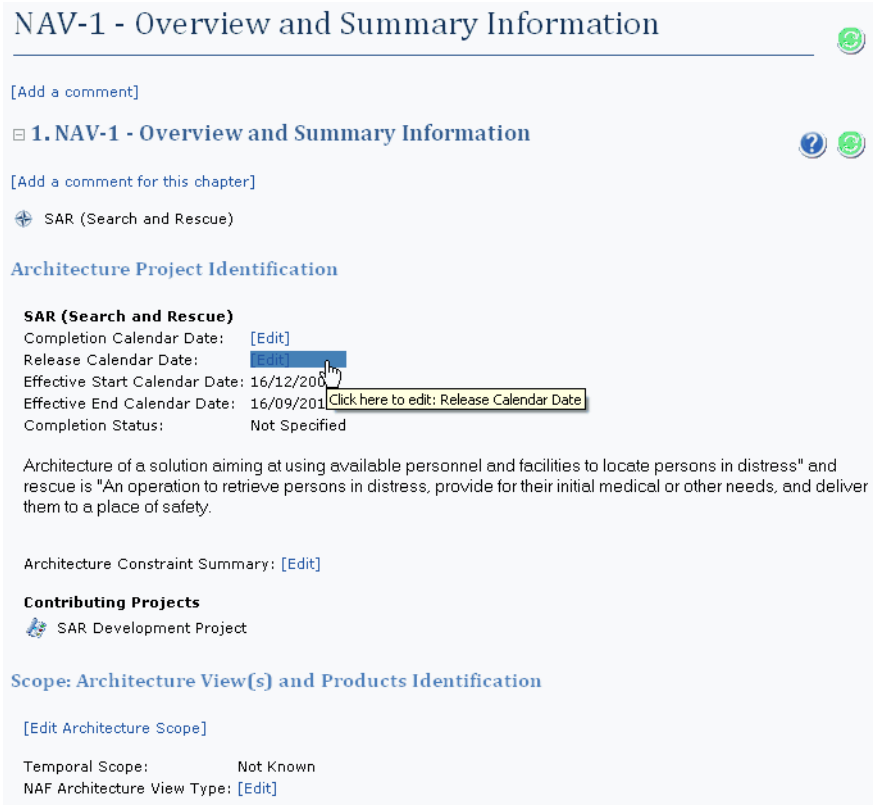
Reports can be used in two ways in **HOPEX NAF**. You can:

- generate them for the reader to understand architecture designs
- use them as a working tool to design architectures.

In the second perspective, you can change the properties of the objects cited in the report using the **Edit** button.

For example, in the NAV-1 subview, the properties used to generate the report are followed by an **Edit** button when the value is not set. This button gives access to the property page of the appropriate object so that the user can fill in the missing information. When you pass the mouse cursor over the word **Edit**, a tooltip appears with the name of the property to be updated.

☛ ***Edit*** *buttons are not displayed on generated web pages or in Word documents.*



Creating Reports

When creating reports, you have the option of doing so:

- for each subview of the NAF architecture, one by one, or
- for all the subviews of the architecture at one go.

Create a report

To create a report:

1. From the **Architectures** navigation tree, expand the architecture concerned, as well as the **Views** folder containing the subview for which the report is required.
2. Expand the subview folder.
3. Right-click the **Reports** folder and select **New > Report**.

   ☛ *The prefix for the name of the report depends on the subview associated. For example, you will select **New > NAV-1 Report** to create a report for the NAV-1 subview.*
4. In the dialog box that appears, enter the name of the report.

   ☛ *A name is automatically proposed and the template on which the report is to be based is automatically selected.*

5. Click **Next**.
6. If you wish to create subsets of the report, do so in the window that appears and click **Next**.
7. Verify that the type of report you wish to generate is selected and click **Finish**.

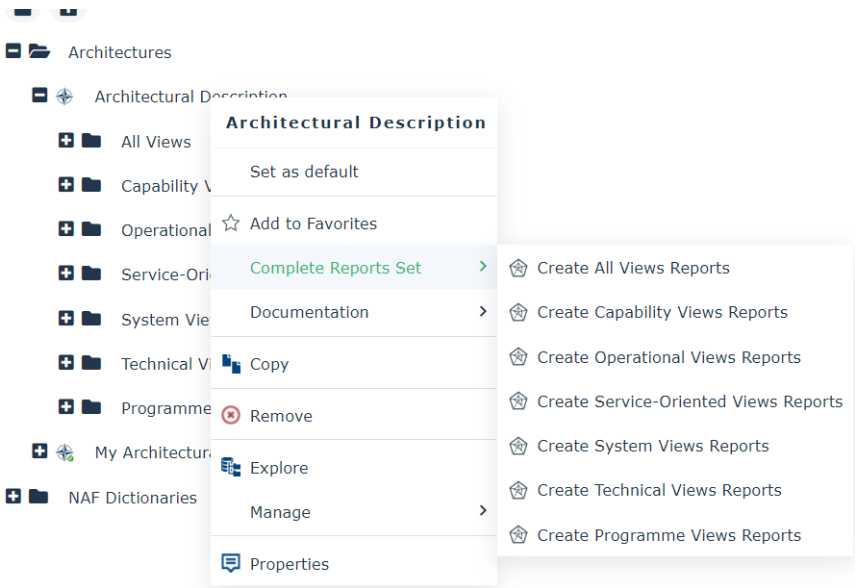
   ☛ *Normally if you do not wish to add subsets to report parameters, click **Finish** in the dialog box in which you enter the name of the report.*

   Your report is created and appears in the NAF navigation window under the folder of the corresponding subview.
8. Double-click the name of the report in the tree to open the report and view its contents.

Creating reports for complete subview sets of an architecture

To create all the reports for all the subviews of a particular View of the architecture:

1. In the **Architectures** navigation window, right-click the architecture concerned.
2. In the pop-up menu that appears, select **Complete Reports Set** and select the report creation option which corresponds to the View concerned, for example, "Create All Views Reports".



All the reports for the All Views subviews are created and appear in the tree under the All Views folder.

MS Word Reports

To keep track of the status of reports, a report can be converted into an RTF or PDF Document (called MS Word Report). This MS Word report is simply a format

conversion of reports. Versions of the MS Word report can be created so that evolutions of the architecture design are stored.

Converting Reports into MS Word Reports

To convert a report into an MS Word report:

1. In the Architectures navigation window, right-click the report to be converted and select **Open as > RTF**.
   The new MS Word report is generated and opens.
2. Name and save the MS Word report in the desired location.
   This MS Word report is disconnected from HOPEX.

Managing Versions of MS Reports

Converting reports into MS Word reports is useful if you wish to distribute generated documents (for example, diagrams and their descriptions) to persons without access to **HOPEX** or if you simply want to keep different versions of the same MS Word report.

The generated RTF MS Word report reflects the state of objects at a given moment and can no longer be modified in **HOPEX** as all links with **HOPEX** are removed. To have your MS Word report reflect modifications of the report, simply convert the report again and save the new MS Word report.

Of course you have the option of saving each MS Word report as a new version to allow you to retrieve information relating to objects at a particular point in time.

The generated MS Word reports and their different versions can be added to **HOPEX** as business documents allowing you to easily access them from the **HOPEX** desktop.

3

HOPEX NAF CONCEPT SUMMARY

CONCEPTS USED IN SUBVIEWS

Code	Main Concepts Used
All Views	
NAV-1 Overview and Summary Information	Enterprise Phases Libraries Timelines
NAV-2 Integrated Dictionary	All architecture-related concepts
NAV-3a Architecture Compliance Statement (Metadata)	All architecture-related concepts
NAV-3b Metadata (Extensions)	Metamodels
Operational Views	
NOV-1 High-Level Operational Concept Description	Library Diagrams External References Notes
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
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

Code	Main Concepts Used
NCV-4 Capability Dependencies	Capability dependencies Capabilities
NCV-5 Capability to Organizational Deployment Mapping	Projects Resource Architectures Org-Units
NCV-6 Capability to Operational Activities Mapping	Business Capabilities Functional activities/processes
Programme Views	
NPV-1 - Programme Portfolio Relationships	Projects
NPV-2 - Programme to Capability Mapping	Projects Time Periods
Service-Oriented Views	
NSOV-1 Service Taxonomy	(Operational) Services
NSOV-2 Service Definitions	Service Definitions
NSOV-3 Capability to Services Mapping	Capabilities Services
NSOV-4a Service Constraints	(Service) Constraints (Service) Requirements
NSOV-4b Service State Model	(Service) State Machines
NSOV-4c Service Interaction Specification	(Service) Interaction scenarios
NSOV-5 Service Functionality	Services Functional Processes/Activities
NSOV-6 Service Composition	Services Functional Processes/Activities
Technical Views	
NTV-1 - Standards Profile	Standards
NTV-2 Standards Forecast	Master Plans
NTV-3 - Standard Configurations	Standards
System Views	

Code	Main Concepts Used
NSV-1 System Interface Description	Applications Artifacts Resource Architectures
NSV-2a System Port Specification	Artifacts Resource Architectures
NSV-2b System to System Port Connectivity	Applications Artifacts Resource Architecture
NSV-2c System Connectivity Clusters	Communication channel clusters
NSV-2d Systems Communication Quality Requirements	(Quality) requirements
NSV-3 Resources-Resources Matrix	System (components)
NSV-4 Systems Functionality Description	System Processes Systems Functions
NSV-5 Systems Function to Operational Activity Traceability Matrix	System Processes Functional Processes/ Functional Activities
NSV-6 Systems Data Exchange Matrix	(System) Contents
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
NSV-12 Service Provision	Physical Resources Services

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.

This section indicates the concepts relating to NAF subviews and the **HOPEX** concepts used to implement them.

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.
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 Subviews

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

- ✓ NAV-1 Overview and Summary Information
- ✓ NAV-2 Integrated Dictionary
- ✓ NAV-3a Architecture Compliance Statement (Metadata)
- ✓ NAV-3b Metadata (Extensions)

NAV-1 Overview and Summary Information

NAV-1 provides executive-level summary information in a consistent form that allows quick reference and comparison among architectures. It includes assumptions, constraints, and limitations that may affect high-level decision processes involving the architecture.

The information included in this subview also includes:

- the **Enterprise Phases** attached to the architecture as enterprise items.

  An enterprise phase is a current or future state of a "Whole Life Enterprise" or another "Enterprise Phase". It can be either spatial (Enterprise Structure) or temporal (Enterprise Temporal Part).

- **Libraries**

  Libraries are collections of objects used to split MEGA 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.

  ☛ *For more information on libraries, see The NAV-1 Library Organization Chapter).*

- **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 Portfolio & Planning** and **HOPEX IT Portfolio Management** documentation.*

The NAV-1 Overview and Summary Information Chapter

This chapter lists all the overview information related to the architecture. It is based on the following structure:

Architecture Project Identification

Identifies the name, the contributing projects and the architects of the architecture, as well as the organizations developing the architecture. It also includes the assumptions and constraints, identifies the approving authority and the completion date, and records the level of effort and costs (projected and actual) required to develop the architecture.

Scope: Architecture View(s) and Product Identification

Identifies the views and subviews that have been developed and the temporal nature of the architecture, such as the time frame covered, whether by specific years or by designation such as current, target, transitional, and so forth. Scope also identifies the organizations that fall within the scope of the architecture.

Purpose and Viewpoint

Explains the purpose and the objectives of the architecture, which include, for example, what the architecture will demonstrate, the types of analyses to be applied, the expected performers of the analyses, the decisions to be made on the basis of the analyses, the decision makers and the resulting actions. The viewpoint from which the architecture is developed is identified.

Context

Describes the setting in which an architecture exists. Context includes things such as mission, doctrine, relevant goals and vision statements, operation concepts, scenarios, information assurance context, other threats and environmental conditions, and geographical areas addressed, where applicable. Context also identifies the authoritative sources of the rules, criteria, and conventions that were followed.

Tools and File Formats Used

Identifies the tools used to develop the architecture as well as the file names and formats used for the architecture and each product. This includes the **HOPEX** release version used, the name of the environment and the name of the repository.

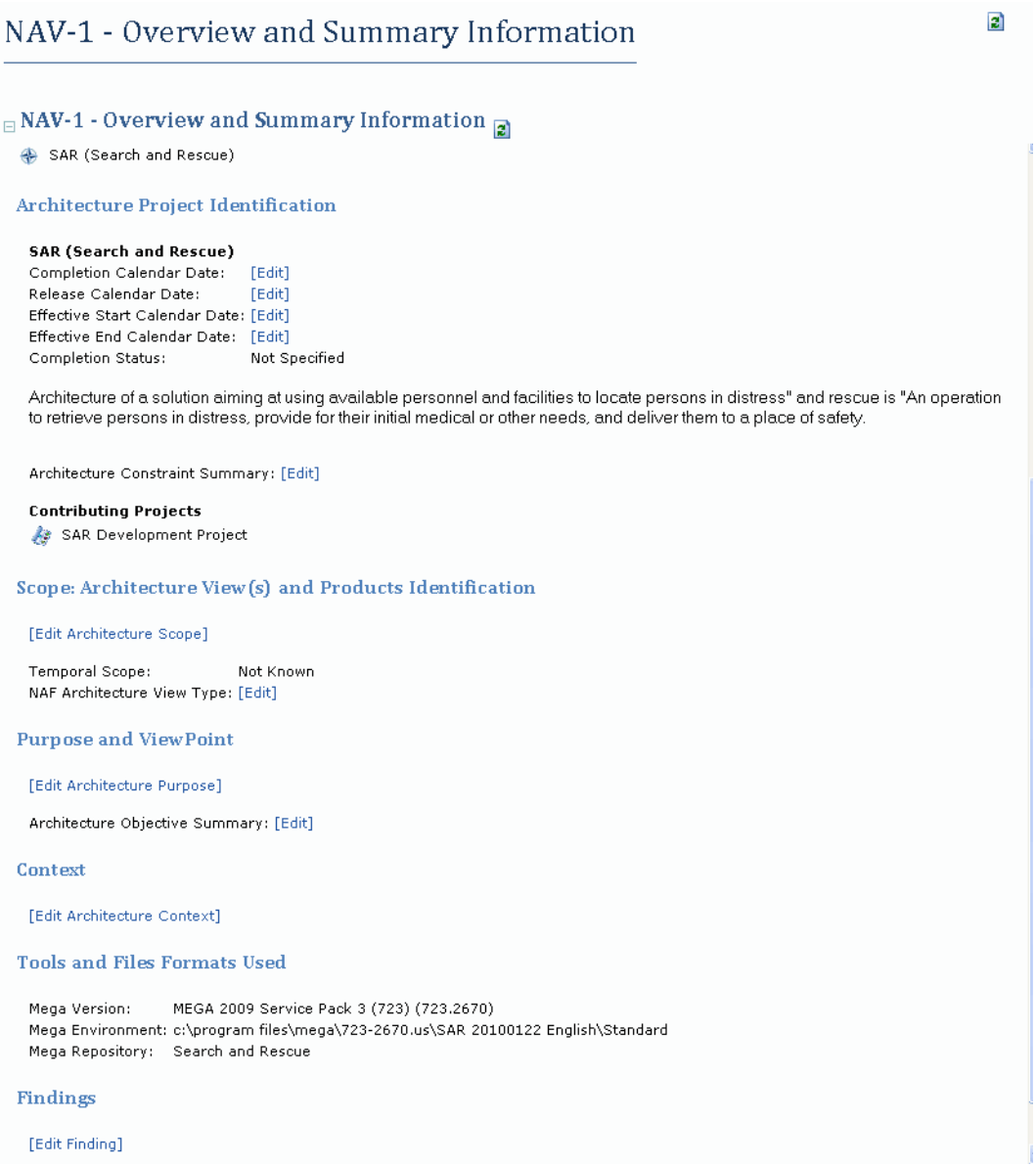
Findings

States the findings and the recommendations that have been developed based on the architectural effort. During the course of developing an architecture, several versions of a product may be produced. An initial version may focus the effort and document its scope, the organization involved, and so forth. After other subviews within an architecture's scope have been developed and verified, another version may be produced to document adjustments to the scope and to other aspects of the architecture that may have been identified. Cost information, such as integration costs, equipment costs and other costs can be included in the findings.

Example

The figure below shows an example of an NAV-1 Overview and Summary Information chapter.

The chapter is generated within **HOPEX** so [**Edit**] sections were inserted to set the missing property values.



NAV-1 - Overview and Summary Information

NAV-1 - Overview and Summary Information

SAR (Search and Rescue)

Architecture Project Identification

SAR (Search and Rescue)
Completion Calendar Date: [Edit]
Release Calendar Date: [Edit]
Effective Start Calendar Date: [Edit]
Effective End Calendar Date: [Edit]
Completion Status: Not Specified

Architecture of a solution aiming at using available personnel and facilities to locate persons in distress" and rescue is "An operation to retrieve persons in distress, provide for their initial medical or other needs, and deliver them to a place of safety.

Architecture Constraint Summary: [Edit]

Contributing Projects
SAR Development Project

Scope: Architecture View(s) and Products Identification

[Edit Architecture Scope]

Temporal Scope: Not Known
NAF Architecture View Type: [Edit]

Purpose and ViewPoint

[Edit Architecture Purpose]

Architecture Objective Summary: [Edit]

Context

[Edit Architecture Context]

Tools and Files Formats Used

Mega Version: MEGA 2009 Service Pack 3 (723) (723.2670)
Mega Environment: c:\program files\mega\723-2670.us\SAR 20100122 English\Standard
Mega Repository: Search and Rescue

Findings

[Edit Finding]

Example of an NAV-1 chapter

The NAV-1 Library Organization Chapter

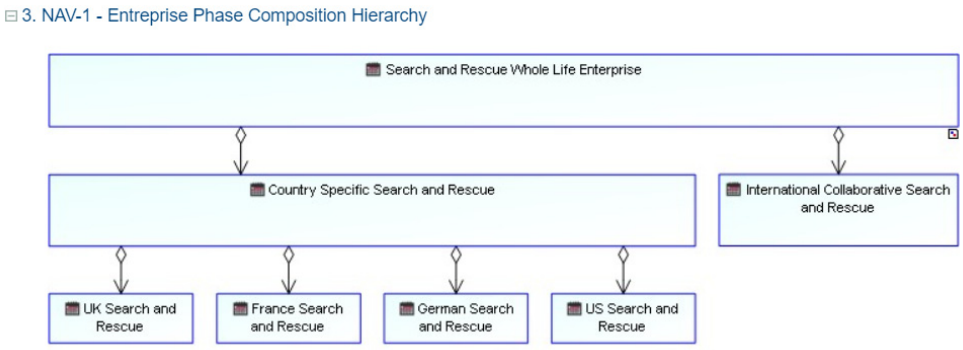
Architectures and data are contained in libraries. Architectures are however dependent on each other and sometimes need to use data contained in other architectures. As such, the libraries containing the different architectures are connected to each other through access links that express the need for data contained in another architecture.

This Library organization chapter shows how the library containing the NAF architecture is structured hierarchically along with the other libraries that it accesses for data.

The NAV-1 Enterprise Phase Composition Hierarchy Chapter

The NAV-1 - Enterprise Phase Composition Hierarchy chapter gives details of how the enterprise of the architecture is structured.

The structure of the enterprise is displayed through the use of enterprise phases in the corresponding tree diagram.



A table is displayed with the enterprise phases in hierarchical form with their name and comment. The associated Master Plans with their milestones are also listed.

The following table shows the hierarchy of the entreprise phases.

	Milestone Dates	Comment
Search and Rescue Whole Life Enterprise		[Edit]
Country Specific Search and Rescue (As in Search and Rescue Whole Life Enterprise)		[Edit] **Composition:** [Edit]
France Search and Rescue (As in Country Specific Search and Rescue)		[Edit] **Composition:** [Edit]
German Search and Rescue (As in Country Specific Search and Rescue)		[Edit] **Composition:** [Edit]
UK Search and Rescue (As in Country Specific Search and Rescue)		[Edit] **Composition:** [Edit]
UK SAR Vision		[Edit]
As-Is		[Edit]
Distant Future		[Edit]
UK SAR Phase 0	3/1/2012	[Edit]
UK SAR Phase 1	1/12/2014	[Edit]
UK SAR Phase 2	1/6/2016	[Edit]
US Search and Rescue (As in Country Specific Search and Rescue)		[Edit] **Composition:** [Edit]
International Collaborative Search and Rescue (As in Search and Rescue Whole Life Enterprise)		[Edit] **Composition:** [Edit]

The NAV-1 Enterprise Phase Dictionary Chapter

The NAV-1 - Enterprise Phase Dictionary chapter presents a list of all the enterprise phases of the architecture.

Each enterprise phase is presented with its long and short name and its comment. The associated Master Plans are also listed.

NAV-2 INTEGRATED DICTIONARY

The NAV-2 Integrated Dictionary contains definitions of terms used in architecture descriptions.

It is not necessarily a literal dictionary. It consists of textual definitions in the form of a glossary, a repository of architecture data, and the taxonomies and metadata (i.e., data about architecture data) of this data, including metadata for tailored products, associated with the architecture products developed.

> ☛ *Metadata refers to architecture data types, which are possibly expressed in the form of a physical schema. In this document, architecture data types are referred to as architecture data elements.*

The **HOPEX** Modeling platform enables users to retrieve all items connected to a set of starting items. This feature is used to automatically build the entire dictionary from a given architecture. Two reports are proposed in relation to each of the following situations:

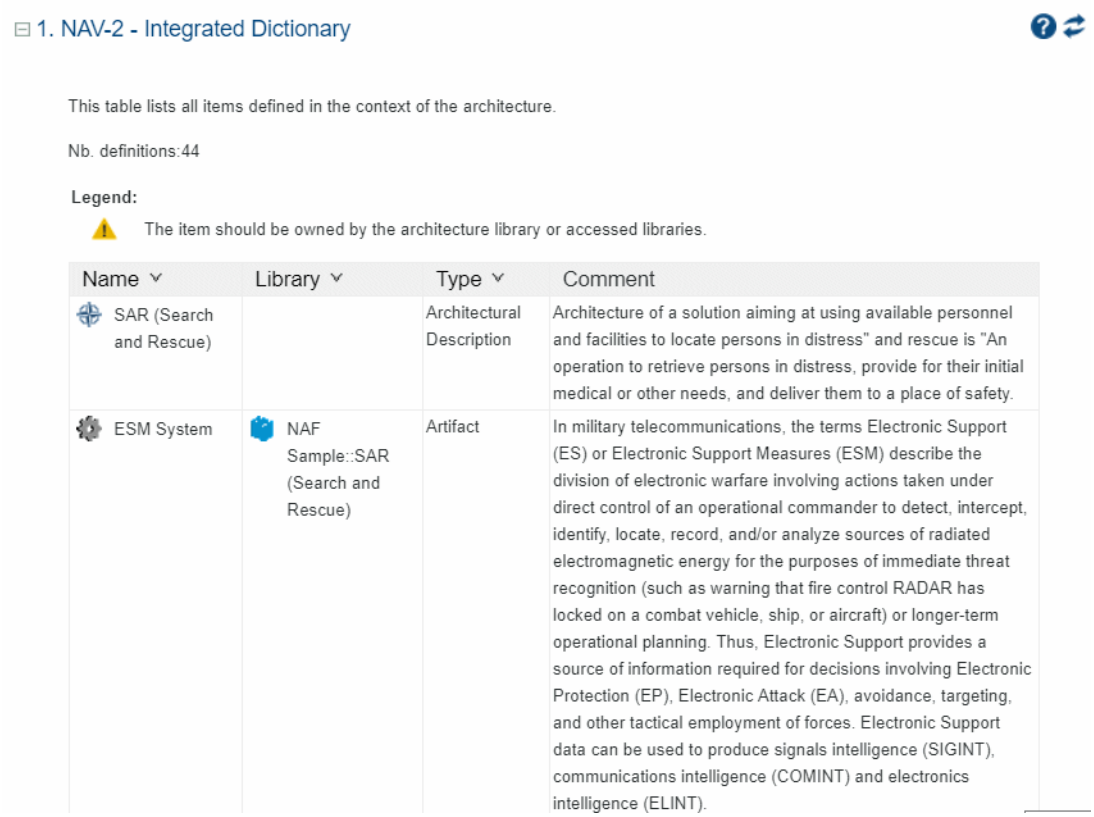
- The user wants to view the commented items
- The user wants to see all items of the architecture in order to determine what must be commented

The NAV-2 Integrated Dictionary Chapter

This chapter addresses the reader who wants to retrieve the definitions of all the items defined or commented within the architecture.

The report generation starts a search for all objects directly and indirectly connected to the analyzed architecture. Only items with comments are considered.

A list of these commented items is generated with their name, comment and type (metaclass). The list is preceded by the number of commented items included.



1. NAV-2 - Integrated Dictionary

This table lists all items defined in the context of the architecture.

Nb. definitions:44

Legend:

The item should be owned by the architecture library or accessed libraries.

Name	Library	Type	Comment
SAR (Search and Rescue)		Architectural Description	Architecture of a solution aiming at using available personnel and facilities to locate persons in distress" and rescue is "An operation to retrieve persons in distress, provide for their initial medical or other needs, and deliver them to a place of safety.
ESM System	NAF Sample::SAR (Search and Rescue)	Artifact	In military telecommunications, the terms Electronic Support (ES) or Electronic Support Measures (ESM) describe the division of electronic warfare involving actions taken under direct control of an operational commander to detect, intercept, identify, locate, record, and/or analyze sources of radiated electromagnetic energy for the purposes of immediate threat recognition (such as warning that fire control RADAR has locked on a combat vehicle, ship, or aircraft) or longer-term operational planning. Thus, Electronic Support provides a source of information required for decisions involving Electronic Protection (EP), Electronic Attack (EA), avoidance, targeting, and other tactical employment of forces. Electronic Support data can be used to produce signals intelligence (SIGINT), communications intelligence (COMINT) and electronics intelligence (ELINT).

Example of an Integrated Dictionary Chapter

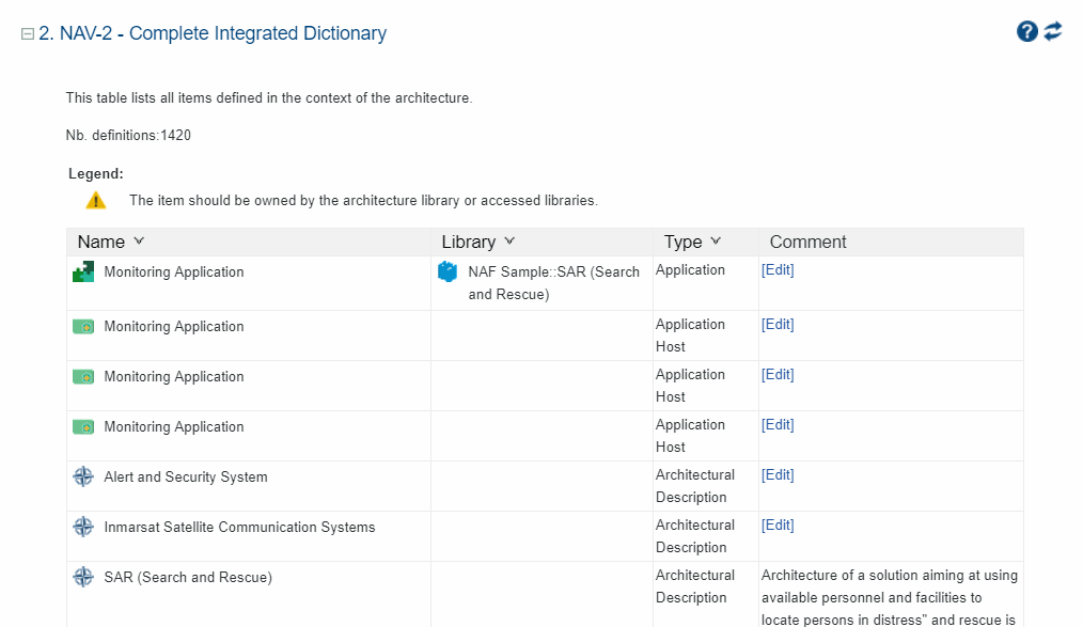
The NAV-2 Complete Integrated Dictionary Chapter

The Complete Integrated Dictionary chapter is based on the same principle as the Integrated Dictionary report. However, items without comments are not removed from the list.

This chapter is used as a working tool to develop the architecture. The goal is to help the architecture designer retrieve the items for which comments are to be added.

☛ *If a report is based on the NAV-2 template, it is possible that the Complete Integrated Dictionary report chapter will not be check-marked in any other context than the **HOPEX** reporting one. To correct this, in*

*the report **Properties**, select the "Report Chapters" tab and change the check-marks.*



2. NAV-2 - Complete Integrated Dictionary

This table lists all items defined in the context of the architecture.

Nb. definitions:1420

Legend:

The item should be owned by the architecture library or accessed libraries.

Name	Library	Type	Comment
Monitoring Application	NAF Sample::SAR (Search and Rescue)	Application	[Edit]
Monitoring Application		Application Host	[Edit]
Monitoring Application		Application Host	[Edit]
Monitoring Application		Application Host	[Edit]
Alert and Security System		Architectural Description	[Edit]
Inmarsat Satellite Communication Systems		Architectural Description	[Edit]
SAR (Search and Rescue)		Architectural Description	Architecture of a solution aiming at using available personnel and facilities to locate persons in distress" and rescue is

Example of a Complete Integrated Dictionary

NAV-3A ARCHITECTURE COMPLIANCE STATEMENT (METADATA)

The NAV-3a subview is used to certify (officially state) that an architecture satisfies all the applicable (externally) imposed criteria to a required degree. The criteria in question are documented in the official NAF documentation.

Its aim is also to communicate to the stakeholders of the architecture effort, and to other architects, that an architecture meets the requirements of the NAF.

The NAV-3a Report Template

There are different parameters required for checking the compliance of architecture items with NAF regulations:

- The **Architecture** parameter which specifies the architecture to be analyzed.
- The **Regulations** parameter, which is optional. You can select the regulations for which you want to check object compliance. However, if this parameter is not set, the architecture objects are checked against all the NAF regulations noted in the database.
- The **Types to Check** parameter. This is also optional. This parameter is used to select the types of objects to be checked. If no types are selected, all the items retrieved from the architecture exploration are checked.
- The **Display All Checked Items** parameter, which is optional. This parameter enables the display of all the checked items, whether their status is OK or not. If this parameter is not set, by default, all the items retrieved that are OK are not displayed.

The NAV-3a NAF Regulation Compliance Chapter

This chapter shows to what extent architecture items are compliant with the regulations that are specified for them. It presents the compliance information in charts with statistics.

Several levels can be defined for the modeling rules:

- **Suggestion**: verification of this rule is not mandatory. As such, this rule only applies in the rules application report.
- **Recommendation**: this rule must be respected. If not an alert (warning) will be displayed.
- **Requirement**: this rule must be respected. If not an error signal will be displayed.

> ☛ *Rules of the "Suggestion" level are not taken into account for object validity definition. They therefore have no influence on the state indicator shown in diagrams or navigators.*

Understanding the statistic results

When the compliance of an object with a particular rule has been checked and the object complies with the rule, the status is marked as OK. If the object does not comply with a particular rule, it takes the rule level that has been defined for the rule (suggestion, recommendation or requirement).

An assessment is then carried out on the object with all the rules that are attached to it.

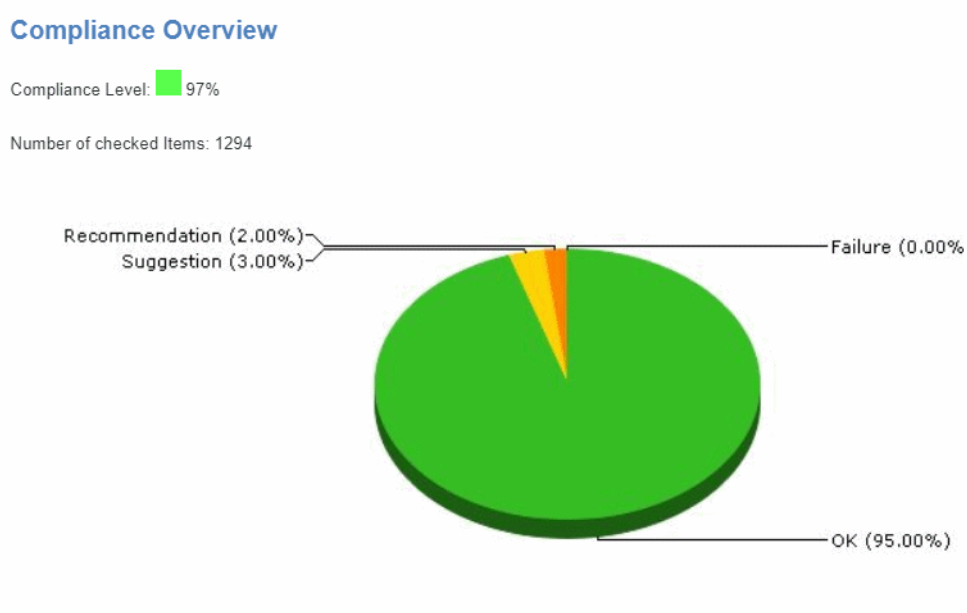
- If the object does not comply with at least one "requirement" rule, the object is given the Failure status.
- If the object complies with all its set "requirement" rules and does not comply with at least one "recommendation" rule, it is given the Recommendation status.
- If the object complies with all its set "requirement" and "recommendation" rules but does not comply with at least one "suggestion" rule, it is given the Suggestion status.
- If the object complies with all its set rules, it is given the OK status.

Contents of the report

The NAF Regulation Compliance report begins with an overview of the level of compliance of architecture items with the regulation rules.

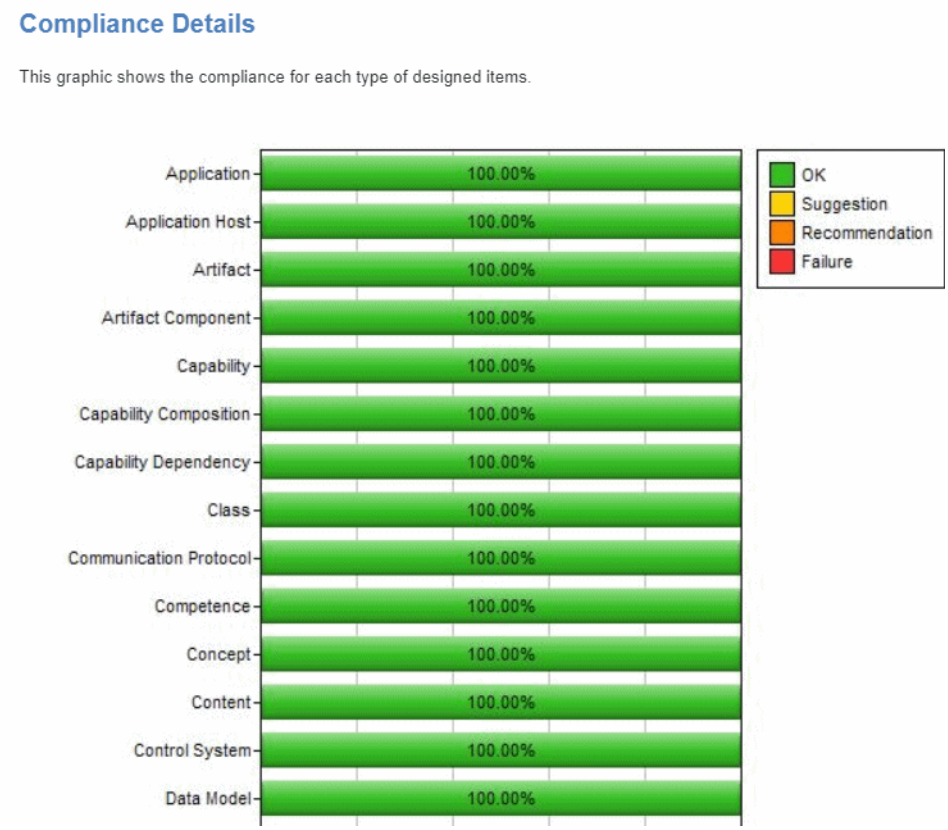
This section displays the following statistics:

- The global percentage of object compliance. Depending on the percentage value a colored square indicates the degree of compliance
- The number of objects checked
- A pie chart with the status percentages: OK, Suggestion, Recommendation, Failure
- A table with the number and percentage of objects with the different rule compliance statuses.



Status	OK	Suggestion	Recommendation	Failure
Number	1233	35	22	4
Percentage	95%	3%	2%	0%

The next section details the level of compliance for the different types of architecture objects.



The final section of the chapter is a table which gives details about the architecture items that do not have the OK status.

This table shows the items of the architecture that do not comply the rules.

Status	Object Name	Object Type	Details
	Maritime SAR Rescue	Interaction Scenario	...
	Detection	Participant	...
	Rescue Provider	Participant	...
	IMO Distress Signal	Message	...
	End	Event	...
	Start	Event	...
	Functional Activity	Functional Activity	...
	Functional Activity	Functional Activity	...
	Functional Activity	Functional Activity	...
	Functional Activity	Functional Activity	...
	Functional Activity	Functional Activity	...
	Functional Activity	Functional Activity	...
	Functional Activity	Functional Activity	...

NAV-3B METADATA (EXTENSIONS)

The purpose of the NAV-3b Metadata Extensions subview is to document any deviations of the architecture's subviews from the standard subview guidelines of the NAF, in terms of deviations from the NAF metamodel (NMM), which underpins all standard subviews of the NAF.

Even though the NAV-3b subview is used to stress the differences between the NMM definition and the **HOPEX NAF** implementation, the default behavior will be to integrate the metamodel diagrams describing the **HOPEX** implementation of the NMM.

The user can add his/her own set of metamodels and can also note the differences as a link comment in the report instance or by adding comments in the diagrams themselves.

The NAV-3b Report Template

The NAV-3b report template comes with two parameters:

- The **Architecture** parameter which specifies the architecture to be analyzed.
- The **Metamodels** parameter, which is optional. This parameter is used to reduce the scope of the study to the selected metamodels. These metamodels describe the **HOPEX** implementation of the NAF architecture description. If this parameter is not set, all the metamodels used to support the NAF subviews of the architecture are taken into account.

The NAV-3b NAF Metamodel Chapter

The NAF Metamodel chapter lists the metamodels of **HOPEX NAF** used to support the different NAF subviews of the specified architecture. The list is arranged in alphabetical order.

A section is dedicated to each metamodel. This section presents:

- a diagram of the metamodel
- a list of all the concepts (metaclasses) used in the metamodel.
  Each concept is presented with a definition as well as a list of its metaAttributes and their comments.
- associations that can be made between metaclasses.

  ☛ *For each association, the source and target metaclass as well as the source and target Roles are also displayed.*

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-5 Capability to Organizational Deployment Mapping
- ✓ 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.*

Objects Types Used in the NCV-1

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.*

☛ *Make sure the corresponding option has been activated. For more information, see "Activating the Objective and Requirements Options", page 26.*

The NCV-1 Report Template

The NCV-1 report template comes with three parameters.

- The **Architecture** parameter, which specifies the architecture to be analyzed.
- The **Objective Subset** is an optional parameter. You can limit the scope of the study to a subset of the selected objectives by setting one or more objectives. All the objectives of the architecture are used in the study if this parameter is not set.
- The **Master Plan Subset**, which is an optional parameter. This parameter is used to reduce the scope of the study to a subset of the selected Master Plan. If this parameter is not set, all the Strategic Master Plans are taken into account.

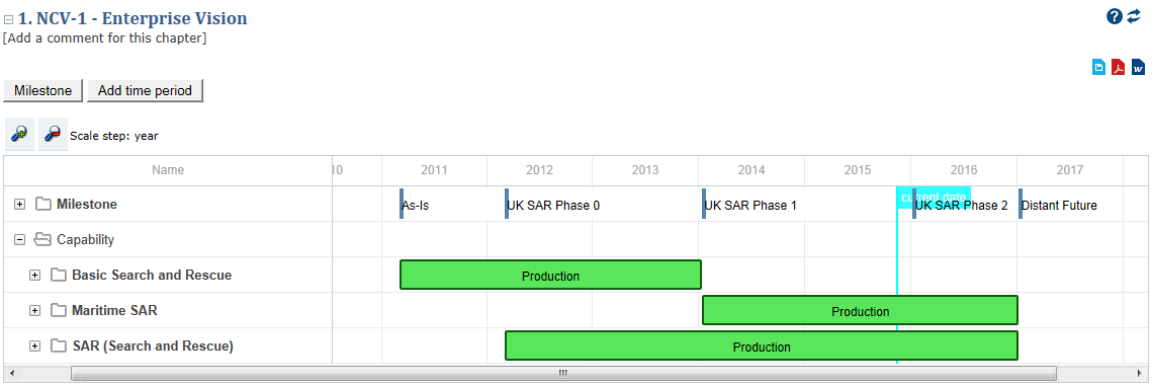
For each notion introduced in the NCV-1 subview (objective, strategic master plan), there is a dictionary and a hierarchical report. Each report gives details through the potential describing diagrams.

The NCV-1 Enterprise Vision Chapter

This chapter shows the details of the Strategic Master plan in the form of a Gantt Chart.

You can modifiy data of the Gantt Chart directly in the report.

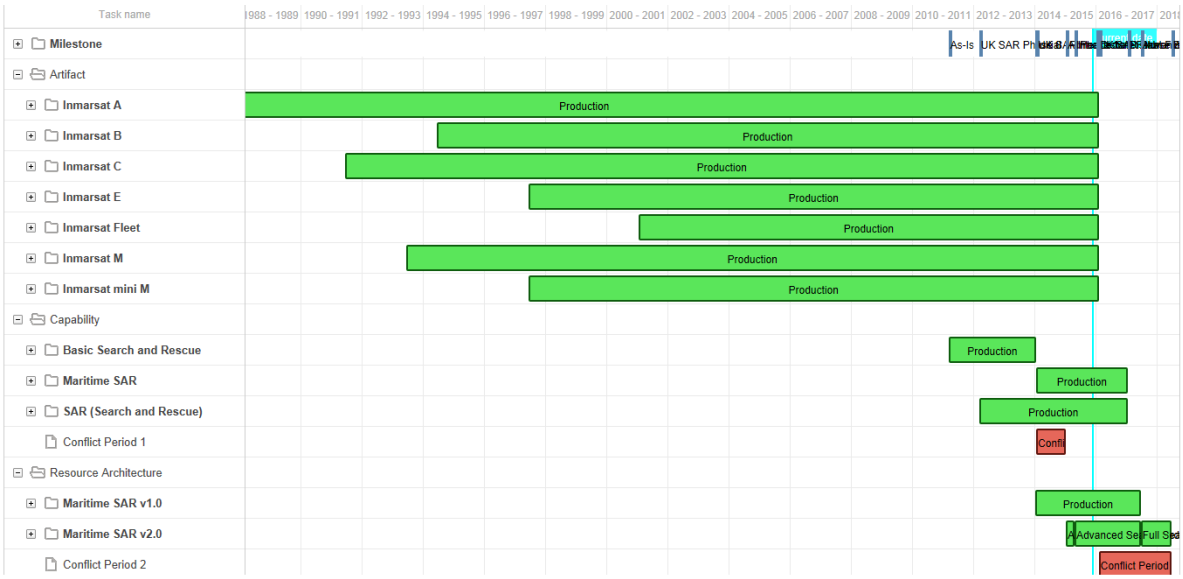
☛ *For more information on creating Master Plans, see NCV-3 Capability Phasing.*



Example of a Gantt Chart in a an NCV-1 Enterprise Vision Chapter

The NCV-1 Enterprise Vision Details Chapter

This chapter displays the conflicts of the strategic Master Plans.



Example of an NCV-1 Enterprise Vision Details Chapter

The NCV-1 Enterprise Goal Hierarchy Chapter

This chapter shows the hierarchy of the objectives of the architecture.

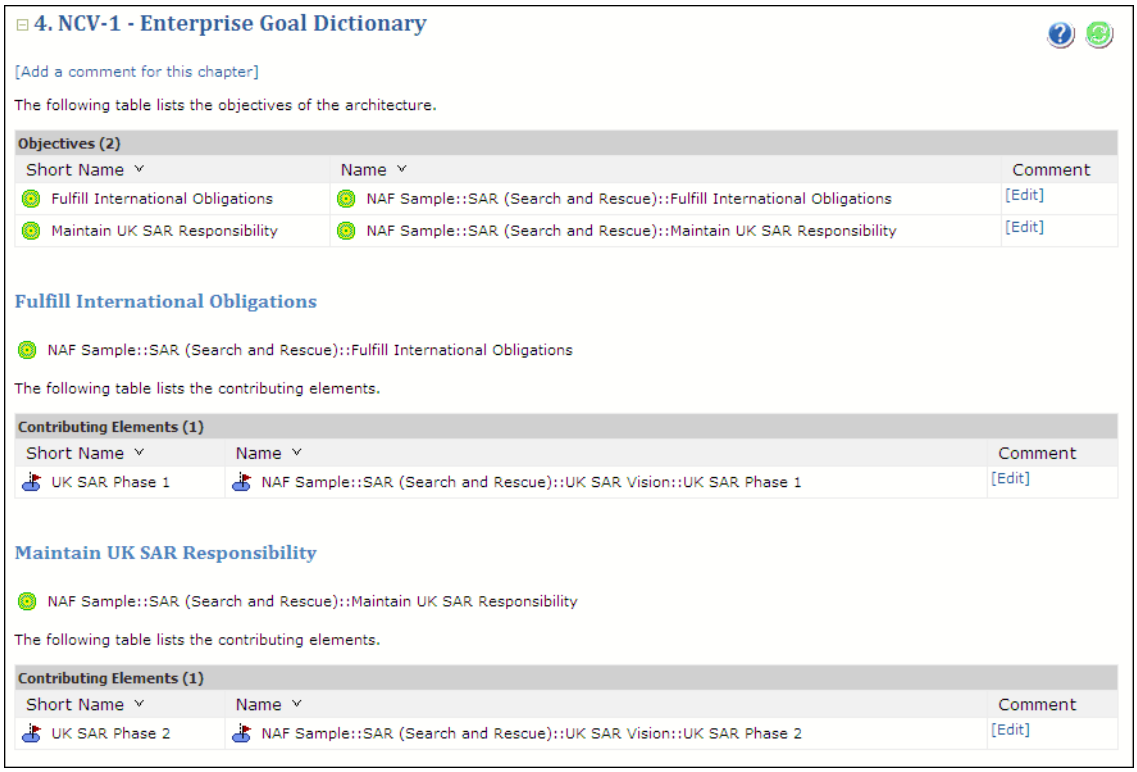
Example of an NCV-1 Enterprise Goal Hierarchy Chapter

The NCV-1 Enterprise Goal Dictionary Chapter

The Enterprise Goal Dictionary chapter presents a list of the objectives used in the report.

A paragraph is dedicated to each objective. This paragraph contains the long and short name of the objective as well as its comment. For each objective, a table is

presented with the name of the milestone to which this objective is attached, in other words where in the plan the objective is to be met.



4. NCV-1 - Enterprise Goal Dictionary

[Add a comment for this chapter]

The following table lists the objectives of the architecture.

Objectives (2)

Short Name	Name	Comment
Fulfill International Obligations	NAF Sample::SAR (Search and Rescue)::Fulfill International Obligations	[Edit]
Maintain UK SAR Responsibility	NAF Sample::SAR (Search and Rescue)::Maintain UK SAR Responsibility	[Edit]

Fulfill International Obligations

NAF Sample::SAR (Search and Rescue)::Fulfill International Obligations

The following table lists the contributing elements.

Contributing Elements (1)

Short Name	Name	Comment
UK SAR Phase 1	NAF Sample::SAR (Search and Rescue)::UK SAR Vision::UK SAR Phase 1	[Edit]

Maintain UK SAR Responsibility

NAF Sample::SAR (Search and Rescue)::Maintain UK SAR Responsibility

The following table lists the contributing elements.

Contributing Elements (1)

Short Name	Name	Comment
UK SAR Phase 2	NAF Sample::SAR (Search and Rescue)::UK SAR Vision::UK SAR Phase 2	[Edit]

Example of the NCV-1 Enterprise Goal Dictionary Chapter

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.

The NCV-2 Report Template

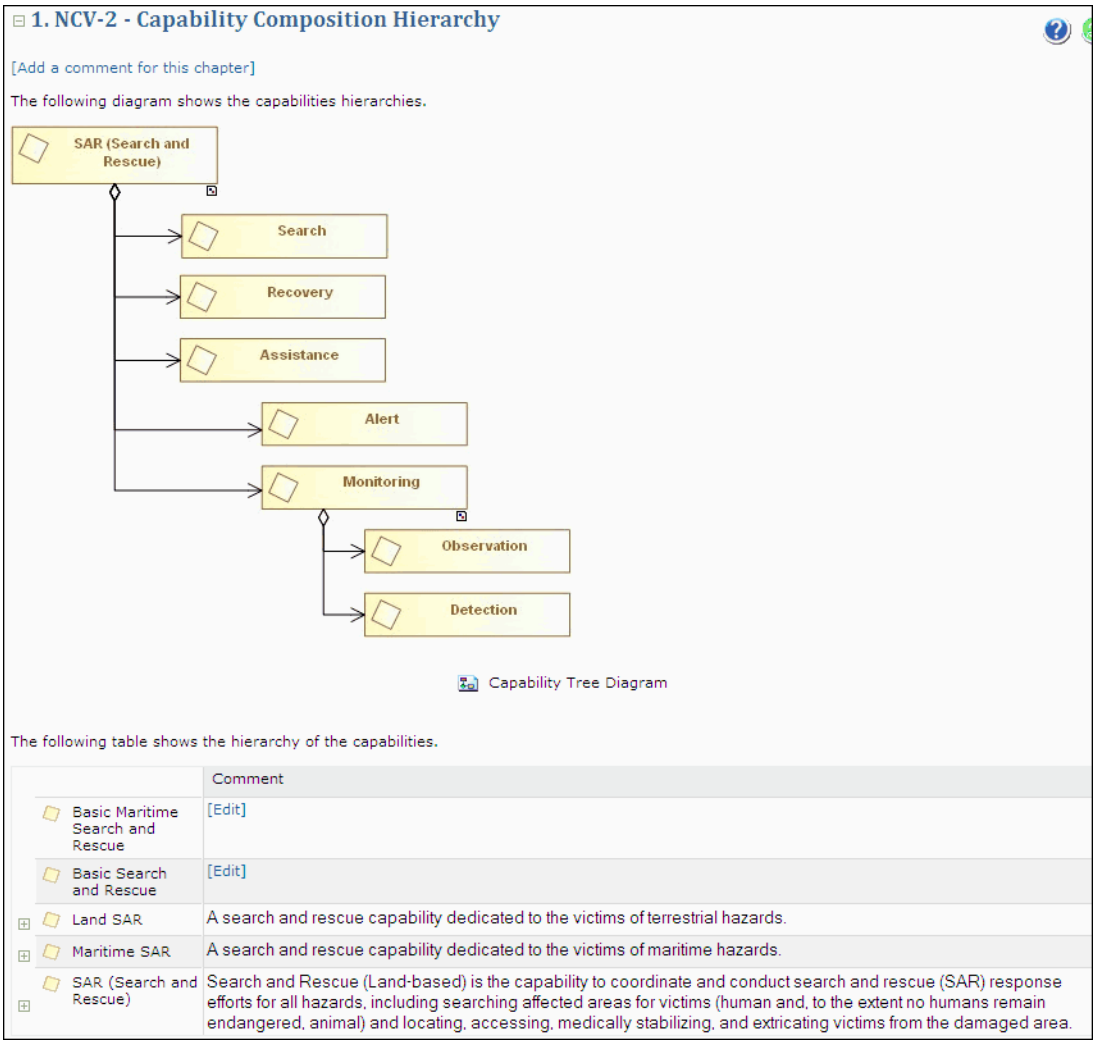
The NCV-2 report template comes with two parameters:

- The **Architecture** parameter, which specifies the architecture to be analyzed.
- The **Capabilities** parameter, which is an optional parameter. You can limit the scope of the study to a selection of capabilities by setting one or more capabilities. All the capabilities of the architecture are used in the study if this parameter is not set.

The NCV-2 Capability Composition Hierarchy Chapter

This chapter gives the structure of a set of capabilities. It can be used as a source document to develop high-level use cases and user requirements.

The information in the chapter is presented in diagram form, if the corresponding diagram exists, as well as in table form, with comments for the capabilities.



Example of the Capability Composition Hierarchy Chapter

The NCV-2 Capability Specialization Hierarchy Chapter

The NCV- 2 capability specialization hierarchy chapter shows how the capability hierarchies of the architecture are specialized.

The varied capabilities are presented in a table along with their variants. The libraries to which the capability belongs as well as the comments of these objects are also indicated.

To create new specializations (variants), use the **New > Variant** command on the capability to be varied (specialized).

The NCV-2 Capability Generalization Hierarchy Chapter

The NCV-2 capability generalization hierarchy chapter shows how the capability hierarchies of the architecture are generalized.

The variants of capabilities are presented in table form along with the varied capabilities (capability from which the variant was created). The comments and library to which the capability belong are also presented.

To add a new generalization, use the **Connect > Variant Of** command of the specialized item (variant).

The NCV-2 Capability Dictionary Chapter

The Capability Dictionary report lists all the capabilities required for the NCV-1 with their type and definition.

This chapter is divided into several sections:

- The first section consists of a list of all the capabilities included in the architecture, with their short name, long name and a comment/ definition.
- The second section is dedicated to the root capabilities.
- For each Root capability, a table is provided with a list of the sub-capabilities and their definitions. The diagrams of the root capabilities are also provided.
  The possible diagrams are:
  - Capability Tree diagram
  - Capability structure diagram

NCV-2 - Capability Dictionary

SAR (Search and Rescue)

Capabilities (10)

Short Name	Name	Comment
Alert	SAR (Search and Rescue)::SAR (Search and Rescue)::Alert::Alert	[Edit]
Assistance	SAR (Search and Rescue)::SAR (Search and Rescue)::Assistance::Assistance	[Edit]
Detection	SAR (Search and Rescue)::SAR (Search and Rescue)::Monitoring::Monitoring::Detection::Detection	[Edit]
Land SAR	SAR (Search and Rescue)::Land SAR	[Edit]
Maritime SAR	SAR (Search and Rescue)::Maritime SAR	[Edit]
Monitoring	SAR (Search and Rescue)::SAR (Search and Rescue)::Monitoring::Monitoring	[Edit]
Observation	SAR (Search and Rescue)::SAR (Search and Rescue)::Monitoring::Monitoring::Observation::Observation	[Edit]
Recovery	SAR (Search and Rescue)::SAR (Search and Rescue)::Recovery::Recovery	[Edit]
SAR (Search and Rescue)	SAR (Search and Rescue)::SAR (Search and Rescue)	[Edit]
Search	SAR (Search and Rescue)::SAR (Search and Rescue)::Search::Search	[Edit]

Land SAR

SAR (Search and Rescue)::Land SAR

The following table lists the sub-capabilities of the current capability.

Sub Capabilities (1)

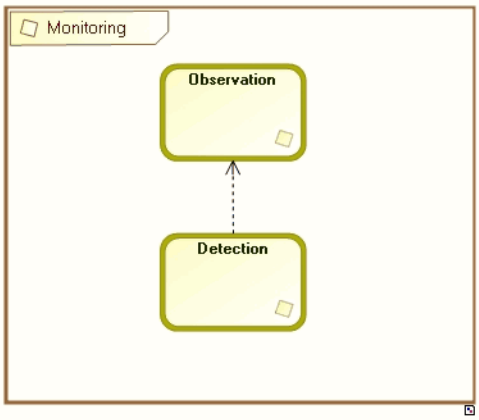
Short Name	Name	Comment
Alert (As in Land SAR)	SAR (Search and Rescue)::SAR (Search and Rescue)::Alert::Alert	[Edit] **Composition:** [Edit]
Assistance (As in Land SAR)	SAR (Search and Rescue)::SAR (Search and Rescue)::Assistance::Assistance	[Edit] **Composition:** [Edit]
Monitoring (As in Land SAR)	SAR (Search and Rescue)::SAR (Search and Rescue)::Monitoring::Monitoring	[Edit] **Composition:** [Edit]
Recovery (As in Land SAR)	SAR (Search and Rescue)::SAR (Search and Rescue)::Recovery::Recovery	[Edit] **Composition:** [Edit]
Search (As in Land SAR)	SAR (Search and Rescue)::SAR (Search and Rescue)::Search::Search	[Edit] **Composition:** [Edit]

Example of Capability Dictionary Chapter

Monitoring

SAR (Search and Rescue)::SAR (Search and Rescue)::Monitoring::Monitoring

Capability Structure Diagram



The following table lists the sub-capabilities of the current capability.

Sub Capabilities (1)

Short Name	Name	Comment
Detection (As in Monitoring)	SAR (Search and Rescue)::SAR (Search and Rescue)::Monitoring::Monitoring::Detection::Detection	[Edit] **Composition:** [Edit]
Observation (As in Monitoring)	SAR (Search and Rescue)::SAR (Search and Rescue)::Monitoring::Monitoring::Observation::Observation	[Edit] **Composition:** [Edit]

Example of Capability Dictionary Chapter with Diagram

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.*

The NCV-3 Report Template

The NCV-3 report template comes with three parameters:

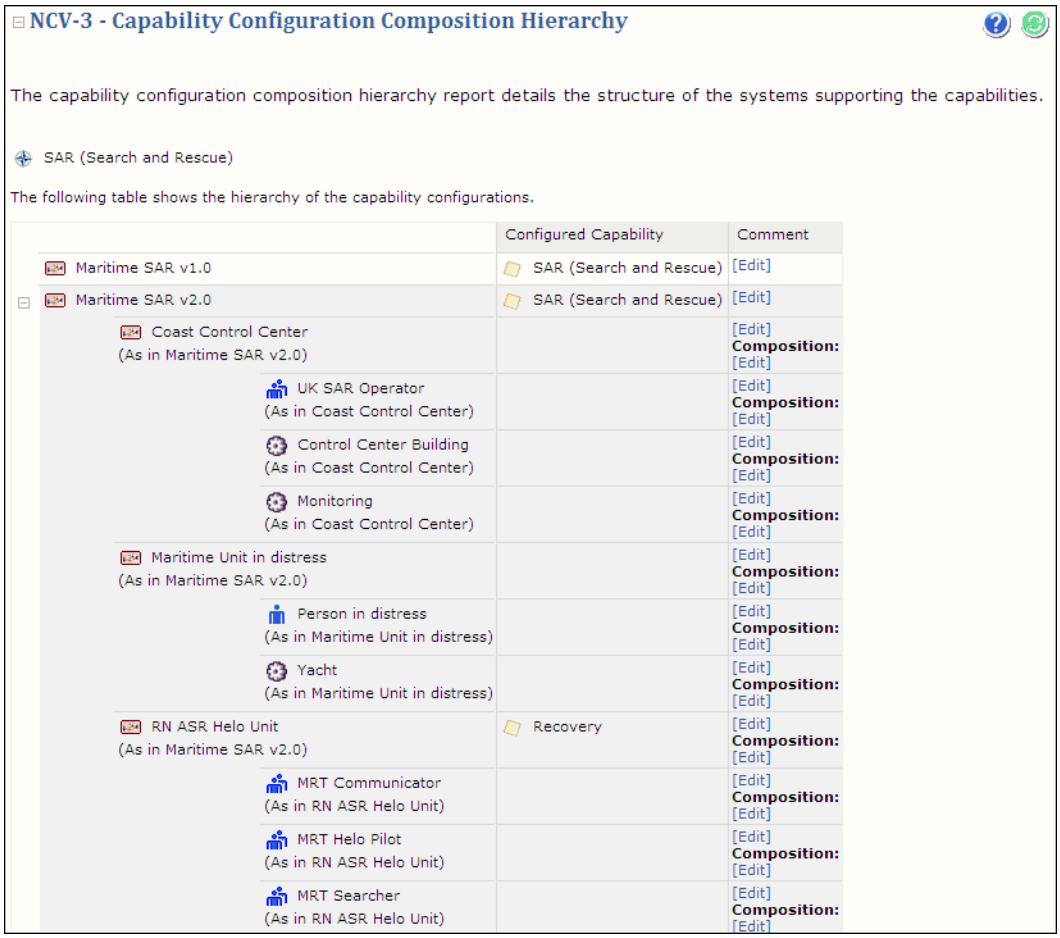
- The **Architecture** parameter, which specifies the architecture to be analyzed.
- The **Capability Configurations Subset** is an optional parameter. You can limit the scope of the study to a subset of the selected capability configurations by setting one or more capability configurations. All the

capability configurations of the architecture are used in the study if this parameter is not set.

- The **Master Plans** Subset, which is an optional parameter. This parameter is used to reduce the scope of the study to a subset of the selected Master Plans. If this parameter is not set, all the Strategic Master Plans are taken into account.

The NCV-3 Capability Configuration Composition Hierarchy Chapter

The capability configuration composition hierarchy chapter presents in tabular form, the elements to be deployed to fulfill certain capability functions. These elements are presented in relation to the hierarchy of the systems that support the capabilities.



NCV-3 - Capability Configuration Composition Hierarchy

The capability configuration composition hierarchy report details the structure of the systems supporting the capabilities.

SAR (Search and Rescue)

The following table shows the hierarchy of the capability configurations.

	Configured Capability	Comment
Maritime SAR v1.0	SAR (Search and Rescue)	[Edit]
Maritime SAR v2.0	SAR (Search and Rescue)	[Edit]
Coast Control Center (As in Maritime SAR v2.0)		[Edit] **Composition:** [Edit]
UK SAR Operator (As in Coast Control Center)		[Edit] **Composition:** [Edit]
Control Center Building (As in Coast Control Center)		[Edit] **Composition:** [Edit]
Monitoring (As in Coast Control Center)		[Edit] **Composition:** [Edit]
Maritime Unit in distress (As in Maritime SAR v2.0)		[Edit] **Composition:** [Edit]
Person in distress (As in Maritime Unit in distress)		[Edit] **Composition:** [Edit]
Yacht (As in Maritime Unit in distress)		[Edit] **Composition:** [Edit]
RN ASR Helo Unit (As in Maritime SAR v2.0)	Recovery	[Edit] **Composition:** [Edit]
MRT Communicator (As in RN ASR Helo Unit)		[Edit] **Composition:** [Edit]
MRT Helo Pilot (As in RN ASR Helo Unit)		[Edit] **Composition:** [Edit]
MRT Searcher (As in RN ASR Helo Unit)		[Edit] **Composition:** [Edit]

Example of a Capability Configuration Composition Hierarchy Chapter

The NCV-3 Capability Configuration Specialization Hierarchy Chapter

The NCV-3 capability configuration specialization hierarchy shows how the capability configurations of the architecture are specialized.

The varied capability configuration are presented in a table along with their variants. The libraries to which the capability configuration belong as well as the comments of these objects are also indicated.

☛ *To create new specializations (variants), use the **New > Variant** command on the capability configuration to be varied (specialized).*

The NCV-3 Capability Configuration Generalization Hierarchy Chapter

The NCV-3 capability configuration generalization hierarchy shows how the capability configurations of the architecture are generalized.

The variants of capability configurations are presented in table form along with the varied capability configurations (capability configuration from which the variant was created). The comments and library to which the capability configurations belong are also presented.

☛ *To add a new generalization, use the **Connect > Variant Of** command of the specialized item (variant).*

The NCV-3 Capability Configuration Dictionary Chapter

The NCV-3 Capability Configuration Dictionary lists all the capability configurations with their definitions. This list is then followed by a dedicated section for each

configuration which includes the diagram describing the configuration and the list of configured capabilities in the configuration.

NCV-3 - Capability Configuration Dictionary

The capability configuration dictionary is an alphabetical list of systems supporting the capabilities defined within the architecture.

SAR (Search and Rescue)

Capability Configurations (4)

Short Name	Name	Comment
Historical Maritime SAR	SAR (Search and Rescue)::Historical Maritime SAR	[Edit]
Maritime SAR Configuration	SAR (Search and Rescue)::Maritime SAR Configuration	[Edit]
RN ASR Helo Unit	SAR (Search and Rescue)::RN ASR Helo Unit	[Edit]
RNLI Lifeboat Unit	SAR (Search and Rescue)::RNLI Lifeboat Unit	[Edit]

Historical Maritime SAR

SAR (Search and Rescue)::Historical Maritime SAR

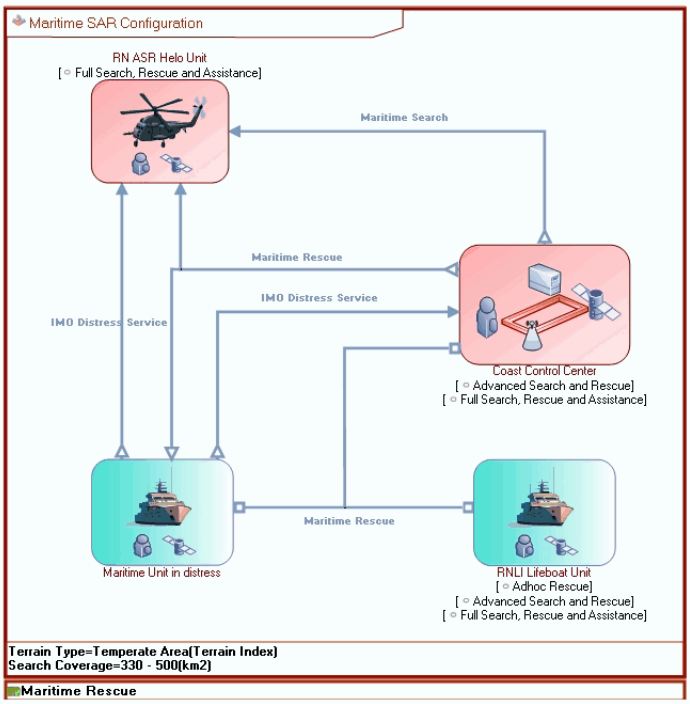
The following table lists the configured capabilities of the current capability configuration.

Configured Capabilities (1)

Short Name	Name	Comment
SAR (Search and Rescue)	SAR (Search and Rescue)::SAR (Search and Rescue)	Search and Rescue (Land-based) is the capability to coordinate and conduct search and rescue (SAR) response efforts for all hazards, including searching affected areas for victims (human and, to the extent no humans remain endangered, animal) and locating, accessing, medically stabilizing, and extricating victims from the damaged area.

Maritime SAR Configuration

SAR (Search and Rescue)::Maritime SAR Configuration



Example of a Capability Configuration Dictionary Chapter

The NCV-3 Capability Phasing Chapter

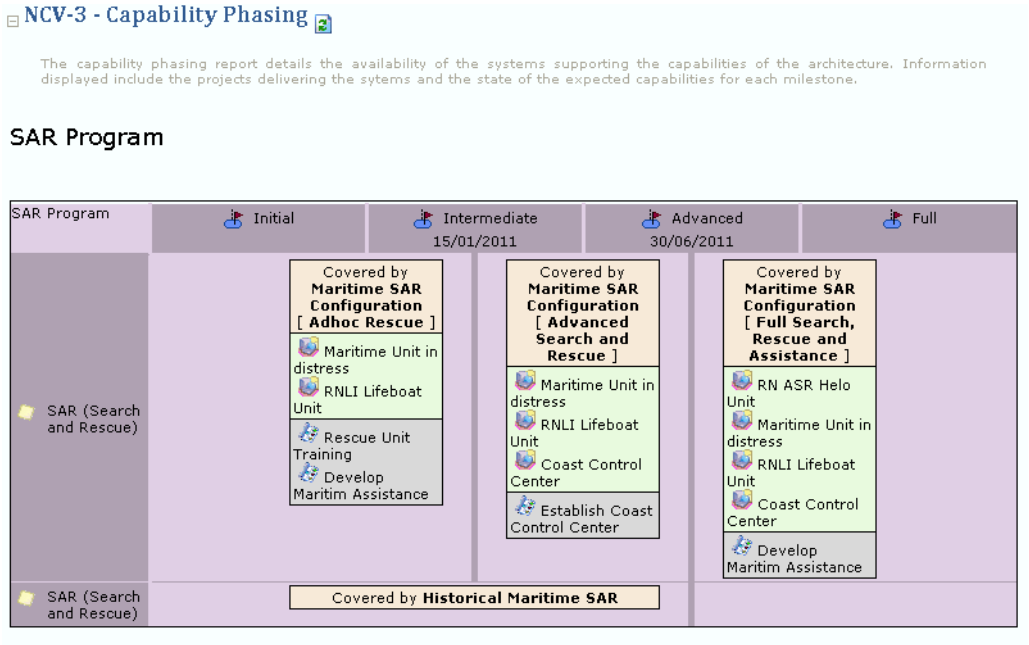
The Capability Phasing chapter presents a Gantt chart. This chart shows in what way projects, through their deliverables, contribute to implementing the resource architectures that will support the capabilities.

It displays:

- the capabilities (derived from the NCV-2 Capability Taxonomy subview) in rows
- phases defined in the Solution Master Plan in columns.

The capability phasing chapter details the availability of the systems that support the capabilities of the architecture.

These systems represent the resource architectures defined in the NSV-1 subview and attached to the capability to create the capability configuration. Information displayed also include the projects that can contribute to delivering the systems, and the state (time period) of the expected capabilities for each milestone. A matrix with the retrieved projects, their deliverables and the time period (states) in which they are expected can be viewed in the NPV-2 Programme to Capability Mapping subview. The projects retrieved are defined with their deliverables in the NPV-1 subview while the states of the expected capabilities are defined in NSV-10b.



Example of a Capability Phasing Chapter

In the Capability Phasing chapter above, the SAR (Search and Rescue) capability is covered by four different states, three of which have been detailed; "Maritime SAR Configuration [Adhoc Rescue]", "Maritime SAR Configuration

[Advances Search and Rescue], "Maritime SAR Configuration [Full Search, Rescue and Assistance]". At the Maritime SAR Configuration [Adhoc Rescue] state, the systems available to support the capability are represented by the Maritime Unit in Distress and RNLI Lifeboat Unit resource architectures. The names of the two projects contribute to implementing the capability during this state; "Rescue Unit Training" and "Develop Maritime Assistance".

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.

The NCV-4 Report Template

The NCV-4 report template comes with three parameters:

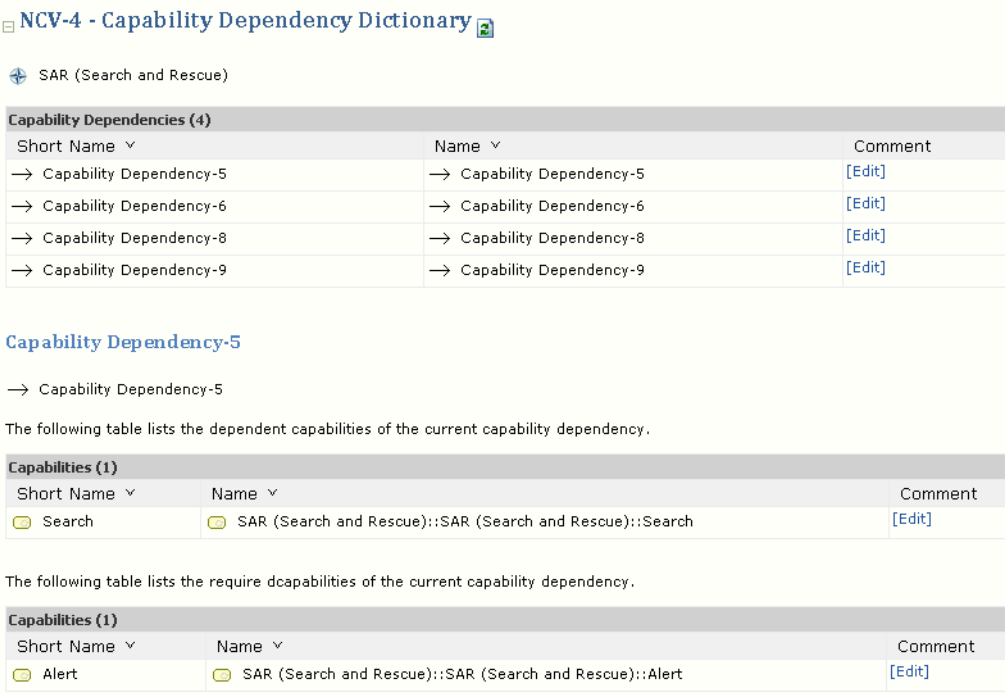
- The **Architecture** parameter, which specifies the architecture to be analyzed.
- The **Capabilities** parameter, which is an optional parameter. You can limit the scope of the study to a selection of capabilities by setting one or more capabilities. All the capabilities of the architecture are used in the study if this parameter is not set.
- The **Capability Dependency** parameter, which is another optional parameter. This parameter is used to reduce the scope of the study to the selected capability dependencies. If this parameter is not set, all the capability dependencies of the architecture are taken into account.
- The **Column Capabilities** parameter. This is also an optional parameter. This parameter enables the creation of a table with different sets of capabilities for rows and for columns.

The NCV-4 Capability Dependency Dictionary Chapter

The Capability Dependency Dictionary chapter lists in alphabetical order the dependencies established between capabilities of the architecture.

These dependencies are set within the context of an upper capability so that sub-capabilities can be dependent in one context but independent in another. lists the

capability dependencies of the architecture. Each capability dependency is described with a list of its dependent and required capabilities.



NCV-4 - Capability Dependency Dictionary

SAR (Search and Rescue)

Capability Dependencies (4)

Short Name	Name	Comment
Capability Dependency-5	Capability Dependency-5	[Edit]
Capability Dependency-6	Capability Dependency-6	[Edit]
Capability Dependency-8	Capability Dependency-8	[Edit]
Capability Dependency-9	Capability Dependency-9	[Edit]

Capability Dependency-5

Capability Dependency-5

The following table lists the dependent capabilities of the current capability dependency.

Capabilities (1)

Short Name	Name	Comment
Search	SAR (Search and Rescue)::SAR (Search and Rescue)::Search	[Edit]

The following table lists the require dcapabilities of the current capability dependency.

Capabilities (1)

Short Name	Name	Comment
Alert	SAR (Search and Rescue)::SAR (Search and Rescue)::Alert	[Edit]

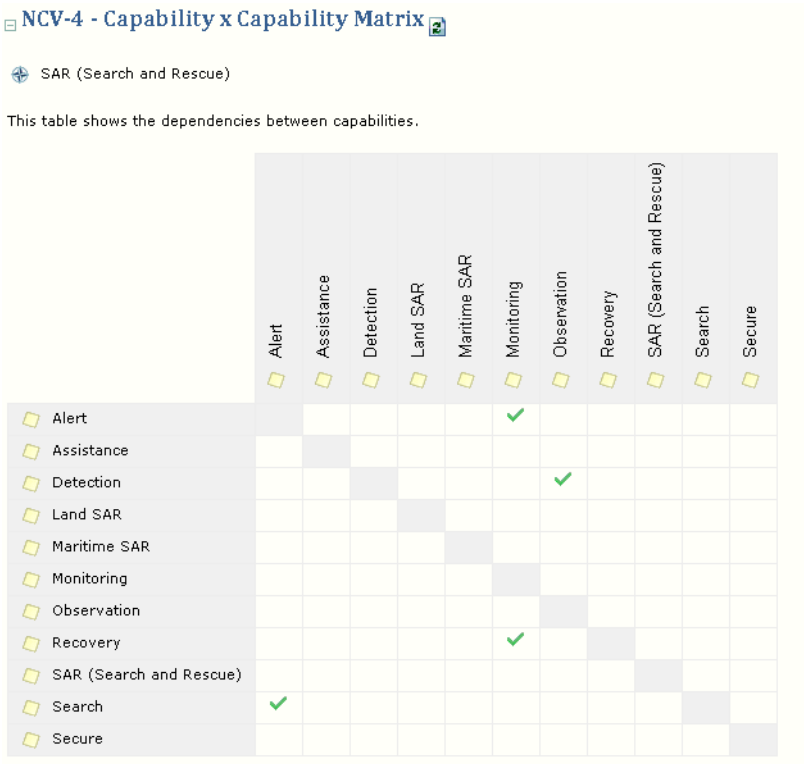
Example of a Capability Dependency Dictionary

The NCV-4 Capability x Capability Matrix Chapter

The Capability x Capability Matrix chapter provides a table which shows the dependencies that exist between different capabilities.

Dependencies are represented by checkmarks. A checkmark appears in the table between two capabilities that are referenced by at least two capability compositions that are linked by a capability dependency.

Checkmarks cannot be added to or removed from the matrix as they are aggregations of capability dependencies.



NCV-4 - Capability x Capability Matrix

SAR (Search and Rescue)

This table shows the dependencies between capabilities.

	Alert	Assistance	Detection	Land SAR	Maritime SAR	Monitoring	Observation	Recovery	SAR (Search and Rescue)	Search	Secure
Alert						✓					
Assistance											
Detection							✓				
Land SAR											
Maritime SAR											
Monitoring											
Observation											
Recovery						✓					
SAR (Search and Rescue)											
Search	✓										
Secure											

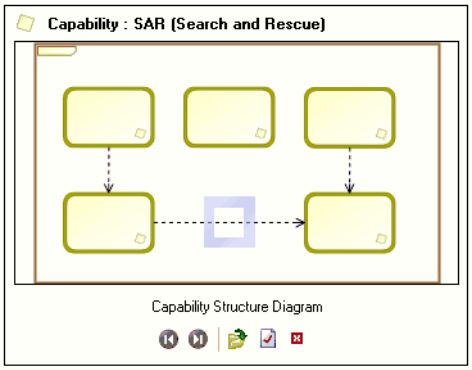
Example of a Capability X Capability Matrix

Accessing the diagram that contains a dependency

To access the Capability Structure Diagram that contain a dependency:

1. Unfold the **NCV-4 Capability Dependencies** folder and right-click the dependency whose diagram you want to access.

2. 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.



3. 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.

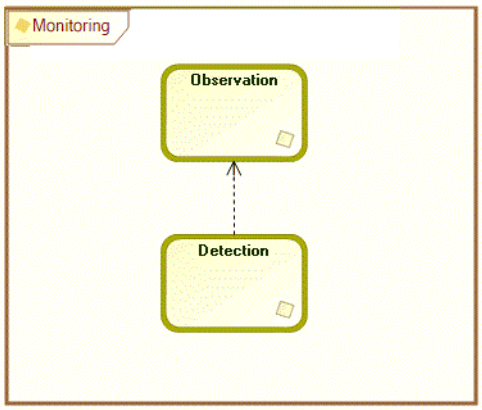
To create a dependency:

- 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

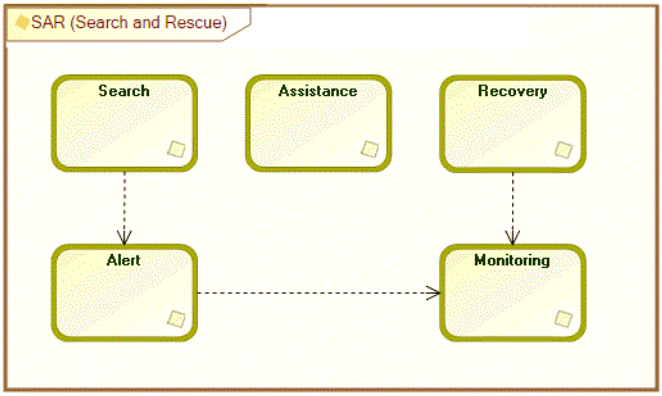
The name of the dependency is a combination of the name of the upper capability and those of the source and target capabilities of the dependency.

```
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

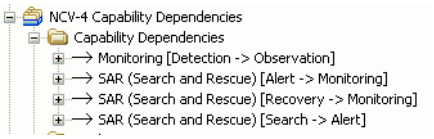
same structure diagram or where a composition is part of dependencies in different structure diagrams. Whatever the case, the naming rule is the same.

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



Example of multiple source dependency

In the NAF navigation tree, dependencies are located in the NCV-4 Capability Dependencies folder, Capability Dependencies subfolder.



Example of dependencies in the NAF navigation tree

NCV-5 Capability to Organizational Deployment Mapping

The NCV-5 capability to organizational deployment mapping model describes the mapping between capabilities and the existing or planned resources (systems) that enable these capabilities during a particular time frame.

The information used to create the NCV-5 subview is drawn from other models (NCV-2, NCV-4, NOV-2, NSV-3, etc), and includes capabilities, system connectivity, organizational structures, and programmatic information.

The NCV-5 Report Template

The NCV-5 report template comes with two parameters:

- The **Architecture** parameter, which specifies the architecture to be analyzed.
- The **Subset** parameter, which is optional. This parameter can be set with a subset of the projects in the NPV-1 subview and the resource architectures of the architecture. If this parameter is not set, all the projects and resource architectures of the architecture are taken into account.

The NCV-5 Capability to Organizational Deployment Mapping Chapter

This chapter contains a matrix with:

- the ***resource architectures*** in columns

  📖 *A resource architecture comprises equipment, IT and organizational resources required for operation of a complex infrastructure (system).*

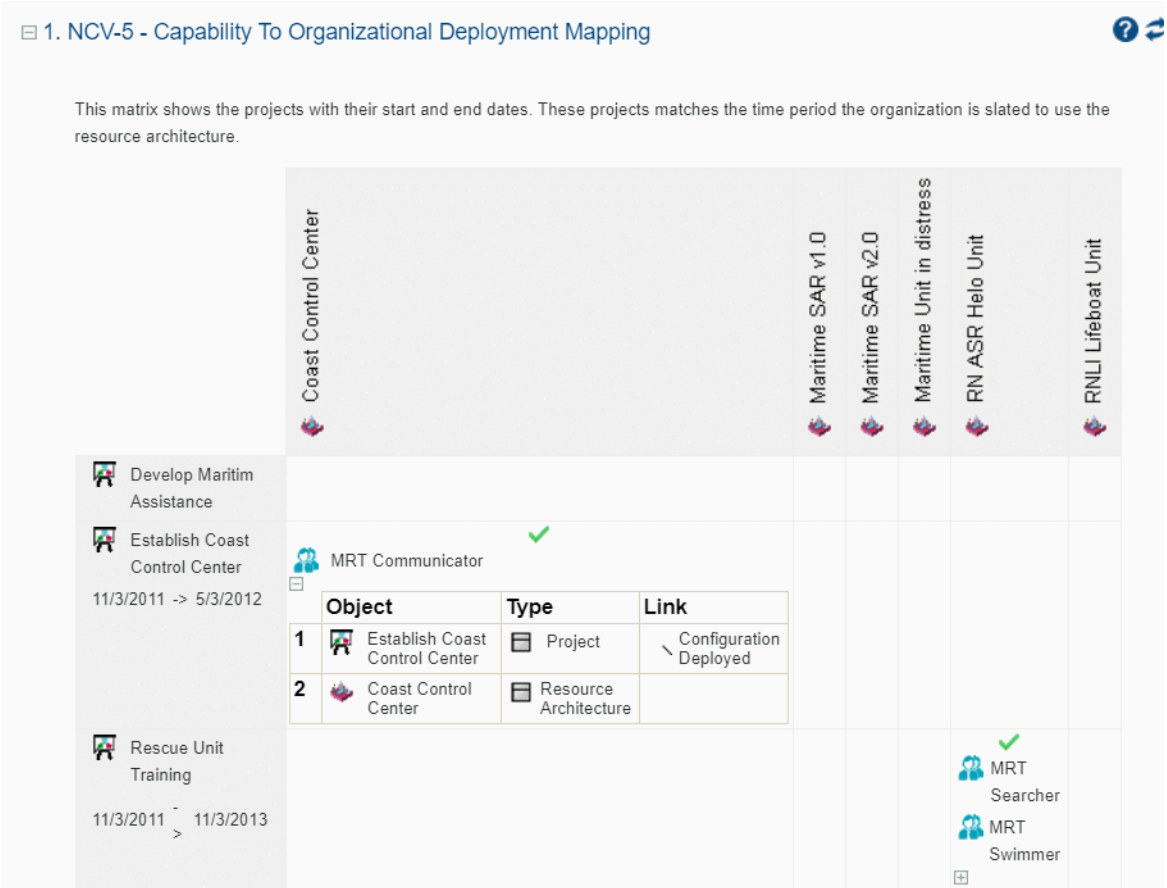
- the ***projects*** in rows.

  📖 *An entrerprise architecture project consists in a set of modeling tasks entrusted to a team in order to transforms a system, or part of a system, so as to achieve a specific objective.*

For each resource architecture-project couple, a checkmark is displayed if a link already exists.

❱ Click in the matrix to add or remove checkmarks.

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.

The NCV-6 Report Template

The NCV-6 report template comes with three parameters:

- The **Architecture** parameter, which specifies the architecture to be analyzed.
- The **Capability** Subset is an optional parameter. You can limit the scope of the study to a subset of the selected capabilities by setting one or more capabilities. All the capabilities of the architecture are used in the study if this parameter is not set.
- The **Operational Activity Subset** parameter, which is an optional parameter. This parameter is used to reduce the scope of the study to a subset of the selected functional activities and/ or functional processes. If this parameter is not set, all the functional activities and functional processes of the architecture are taken into account.

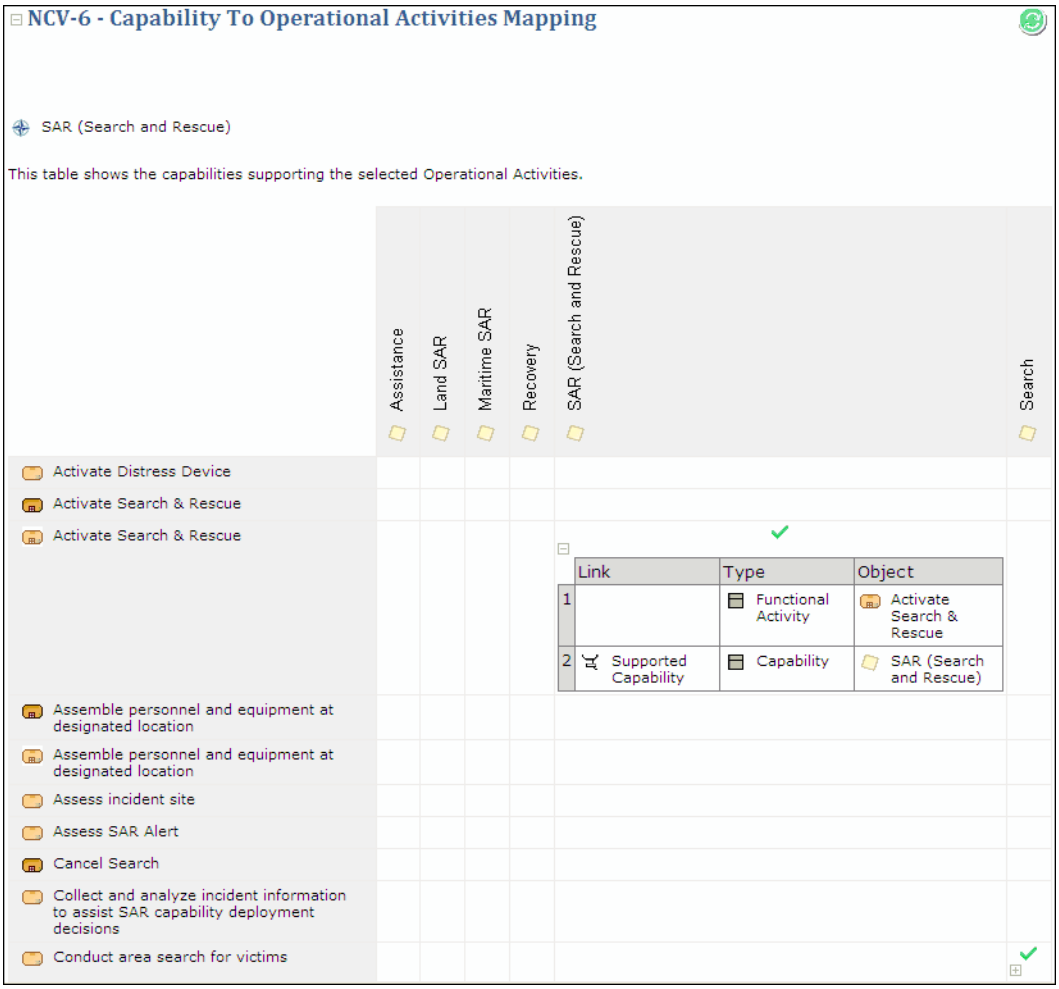
The NCV-6 Capability to Operational Activities Mapping Chapter

The Capability to Operational Activities Matrix chapter provides a table which shows the capabilities that support selected operational (functional) activities and functional processes.

This support is represented by checkmarks. A checkmark appears in the table between an operational activity or functional process and the capability that supports it.

- Click in the matrix to add or remove checkmarks.

  ☛ *When links exists a "+" appears under the checkmark. If you click on this "+" a table appears with the type of link created, and the object type and objects that are linked.*



Example of an NCV-6 Capability to Operational Activities Mapping Matrix

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.

The NOV-1 Report Parameters

The NOV-1 report template is supplied with one parameter: the architecture to be analyzed. There are no additional parameters.

The NOV-1 High-Level Operational Concept Graphic Chapter

This chapter gives access to all the general documentation regarding the architecture.

First, the chapter shows the "High-Level Operational Graphic" diagram, which gives a general understanding of the architecture.

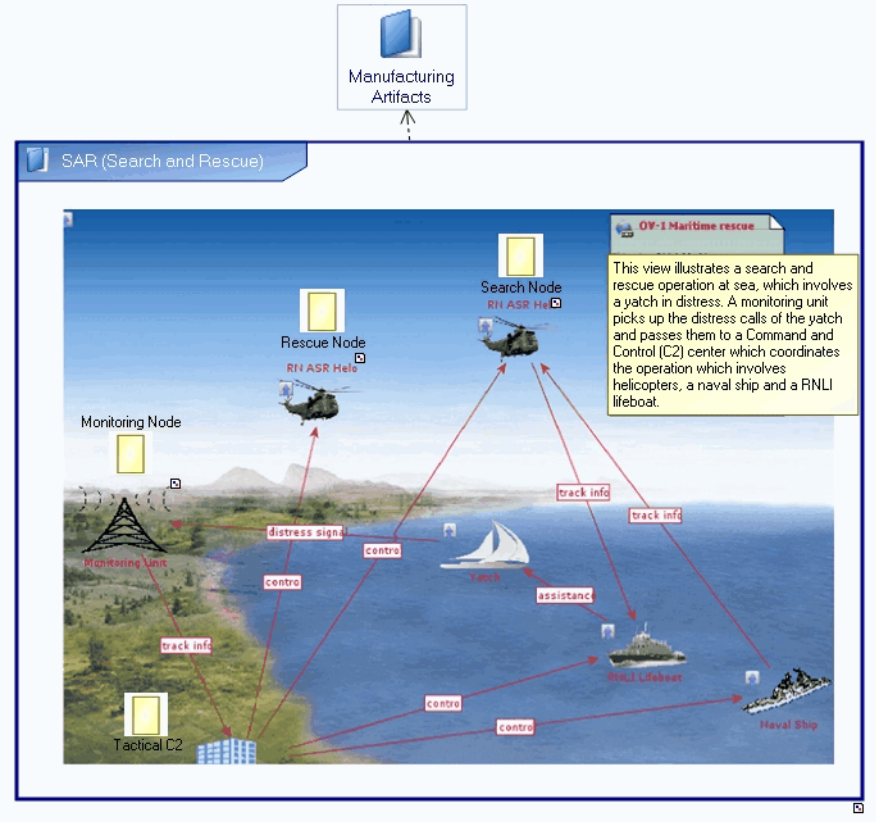
The diagram is followed by a list of all notes linked to the architecture, then a table with the external references linked to the architecture. Depending on the report

format the external references are displayed as a hyperlink (HTML) or a name with the complete address of the external document.

> ☛ *When external references are added to the "High-Level Graphic" diagram (Library diagram in **HOPEX** terminology), they are not automatically linked to the architecture. This is so because in **HOPEX** external references can be added to any diagram, diagrams which are not necessarily linked to architectures. To ensure that external references are linked to an architecture, this link must be made from the **Properties** dialog box, **General** tab, **References** sub-tab of the architecture in the NAF navigator window. The external references then appear under the architecture in the NAF navigation window.*

A table of the cited items is also inserted in the chapter with their name, type and comment.

The figure below shows an example of a Library diagram.



Example of an NOV-1 Library Diagram

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.

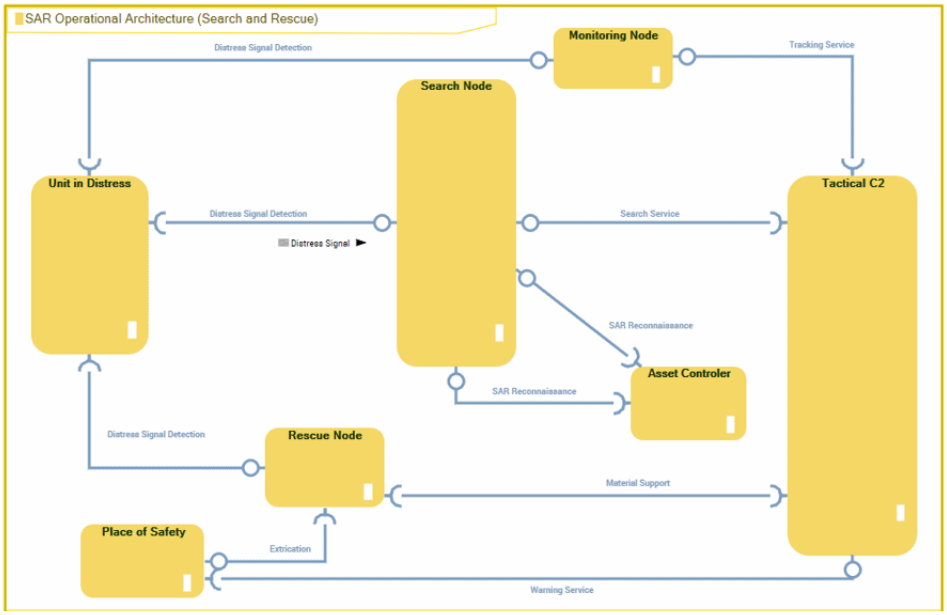
> ☛ *Operational nodes can also be initially created in the operational node structure diagram. See Creating Operational Node Structure Diagrams.*

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.

The NOV-2 Report Template

NOV-2 reports are generated from the NOV-2 report template.

It is structured to analyze the architecture to which it is associated.

However, additional and optional parameters called Subsets can be used to restrict the scope of the report. T

- **Operational Node Subset**: This parameter can be set with interactions, operational nodes and diagrams. The type of values used for the parameter determines the scope of the report:
  - **Operational Node**: only the specified operational nodes are considered. Embedded operational nodes are excluded. All incoming and outgoing interactions are included as well as the targeted operational nodes. The process of operational node collection is not reiterated on operational nodes found from interactions.
  - **Interaction**: These are the interactions that exist between operational nodes. Only the specified interactions are considered and source and target operational nodes are included.
  - **Diagram**: Operational nodes and interactions displayed in the diagrams are included. The diagrams must be described in the architecture analyzed.
- **Resource Type Subset**:

  ☛ *This parameter can be set with applications, artifacts, org-units, resource architectures.*

  The type of values used for the parameter determines the scope of the report. If no parameter is set, all the applications, artifacts, org-units and resource architectures are included in the report.
  - **Application**: Only the specified applications are considered for the report.
  - **Artifact**: These artifacts are the physical assets of the system that are neither applications or organizational elements.
  - **Org-Unit**: These can be humans or simply organizational elements. They can be internal or external to the enterprise.
  - **Resource Architecture**: These resource architectures are combinations of the physical assets and organizations used to provide a capability.

The NOV-2 Operational Node Composition Hierarchy Chapter

This chapter gives a view of the operational node hierarchies.

Operational Node Tree Diagrams are displayed with the complete structure, taking into account the optional scope.

The NOV-2 Operational Node Specialization Hierarchy Chapter

The NOV-2 operational node specialization hierarchy chapter shows how the operational nodes of the architecture are specialized.

The operational nodes are presented in a table along with their comments and the library to which they belong.

> ☛ *If a variant of an operational node is created, the varied operational node is presented in the table along with its variant.*

To create new specializations (variants), use the **New > Variant** command on the operational node to be varied (specialized).

The NOV-2 Operational Node Generalization Hierarchy Chapter

The NOV-2 operational node generalization hierarchy chapter shows how the operational nodes of the architecture are generalized.

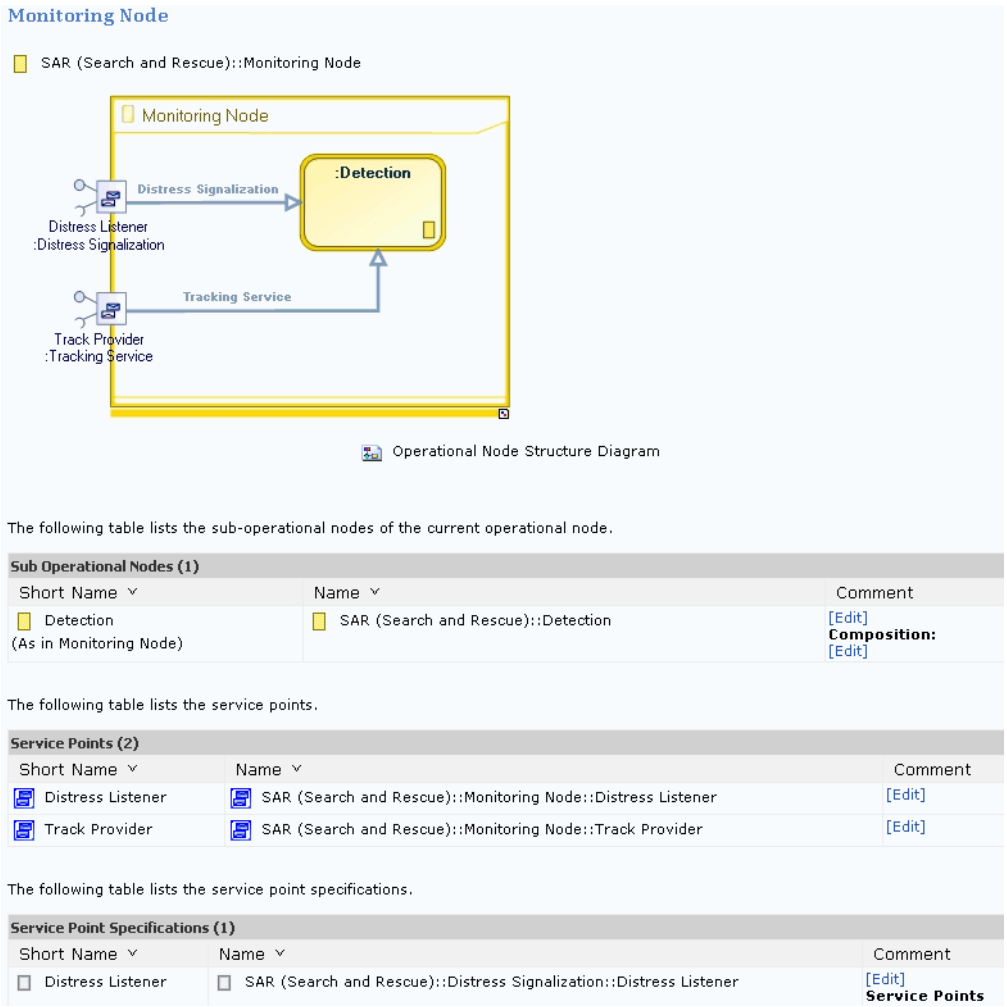
The operational nodes are presented in table form along with their comments and the library to which they belong. If a variant of an operational node exists, this variant is also presented in the table along with the varied operational node (operational node from which the variant was created).

To add a new generalization, use the **Connect > Variant Of** command of the specialized item (variant).

The NOV-2 Operational Node Dictionary Chapter

The Operational Node Dictionary lists the operational nodes defined in the architecture in an alphabetical table. This enables the retrieval of items for which the names are known.

It also displays the Operational Node Structure diagram for the different operational nodes.



Example of Operational Node Dictionary Chapter with Diagram

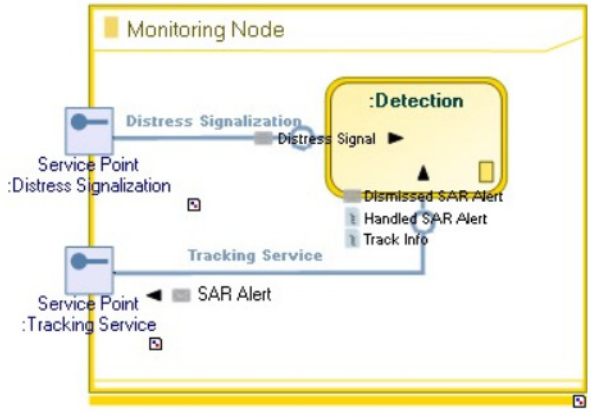
The NOV-2 Operational Node Exchange Chapter

This chapter lists the needs that exist for the exchange of information from specific operational nodes to others.

Depending on the scope defined (or the default scope: the entire architecture), the report lists a set of needlines as well as the source and target operational nodes.

For each needline, the exchanged information is also listed.

The structure diagrams in which the different needs are defined are displayed.



Operational Node Structure Diagram

The following table lists the interactions involving each operational node (operational component) with their expected information element.

Interactions			
Name	Origin Operational Nodes	Destination Operational Nodes	Information Elements
Material Support	Tactical Control Center (Tactical Control Center)	Asset Controler (Asset Controller) Rescue Node (Rescue Node)	

Example of an Operational Node Exchange Chapter

The NOV-2 Operational Node Exchange Balance Chapter

This chapter is used to verify that the exchanges defined between operational nodes are correctly designed.

All the incoming and outgoing exchanges of the operational nodes are compared and examined to reveal missing as well as unnecessary information for the exchange design.

The report lists all the internal and outside messages and interactions that detail the selected operational nodes.

The NOV-2 Operational Node Exchange Compliance Chapter

The Operational Node Exchange Compliance chapter indicates the level of compliance between the need to exchange information from one operational node to another and the information actually exchanged through the implemented missions.

Activities supported by the operational nodes actually exchange information within the context of designed missions. This information is compared with the need defined in the needlines/interactions. Missing or unexpected information is detected.

This chapter relies on the nodes defined in the NOV-2 view as well as the functional processes that describe how missions are performed.

☛ *See NOV-2 Operational Node Connectivity Description.*

These processes are described in detail in NOV-5, however, a brief description is necessary to aid in understanding the content of this chapter.

☛ *See NOV-5 Operational Activity Model.*

A node structure does not indicate how missions are performed. It only describes the nodes involved in the mission and how they are structured. Through interactions, you can guess the potential information exchanges that occur between nodes.

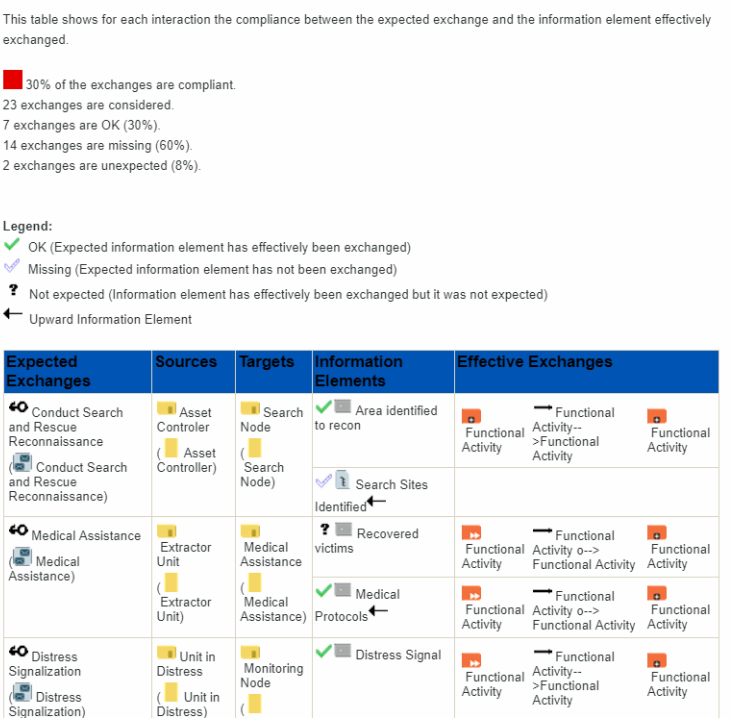
To add a dynamic perspective to nodes, a process must be described (see the NOV-5 view). This process is composed of activities that exchange information. Information is actually exchanged in the process while the node structure only describes the ability to perform such exchanges.

Activities are performed under the control of the operational nodes.

The Operational node exchange compliance report identifies three states for the information exchange:

- **Information Exchange Correctly Designed**: an information item is defined at both the node and the process levels.
- **Missing Information**: an information item can potentially be exchanged between two nodes, however, there is no process available to perform this exchange. The question is therefore, whether the interface described between the two nodes is still useful.
- **Unexpected Information**: an information item is exchanged in a process between two activities, however, there is no interaction between the nodes carrying out the activities. The question is therefore, should the information be exchanged in this manner in the process and must

the interaction be reviewed to depict the ability to make such an exchange.



This table shows for each interaction the compliance between the expected exchange and the information element effectively exchanged.

30% of the exchanges are compliant.
23 exchanges are considered.
7 exchanges are OK (30%).
14 exchanges are missing (60%).
2 exchanges are unexpected (8%).

Legend:
OK (Expected information element has effectively been exchanged)
Missing (Expected information element has not been exchanged)
Not expected (Information element has effectively been exchanged but it was not expected)
Upward Information Element

Expected Exchanges	Sources	Targets	Information Elements	Effective Exchanges
Conduct Search and Rescue Reconnaissance (Conduct Search and Rescue Reconnaissance)	Asset Controler (Asset Controller)	Search Node (Search Node)	Area identified to recon	Functional Activity — Functional Activity-->Functional Activity — Functional Activity
			Search Sites Identified	
Medical Assistance (Medical Assistance)	Extractor Unit (Extractor Unit)	Medical Assistance (Medical Assistance)	Recovered victims	Functional Activity — Functional Activity o--> Functional Activity — Functional Activity
			Medical Protocols	Functional Activity — Functional Activity o--> Functional Activity — Functional Activity
Distress Signalization (Distress Signalization)	Unit in Distress (Unit in Distress)	Monitoring Node (	Distress Signal	Functional Activity — Functional Activity-->Functional Activity — Functional Activity

Example of an NOV-2 Operational Node Exchange Compliance Chapter

The NOV-2 Operational Node Realization Chapter

The NOV-2 Operational Node Realization chapter displays in a matrix, the resources (human and non human) responsible for performing the activities required for an operational node.

The matrix displays:

- resource items in columns
- operational nodes in rows.

Checkmarks appear in the matrix to show where a particular resource is used for an operational node. You can add and remove checkmarks in the matrix.

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 Report Template
✓ The NOV-3 Operational Information Exchange Dictionary
✓ The NOV-3 Contents Linked to Operational Items Chapter
✓ The NOV-3 Operational Information Exchange Matrix Chapter

Object Types used in the NOV-3

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.*

The NOV-3 Report Template

The report template used to generate the NOV-3 report chapters is composed of three parameters: the architecture described and two optional values (needline subset and exchange properties) that reduce the scope of the study.

- The **Architecture** parameter is the architecture to be analyzed
- The **Needline Subsets** parameter is set with some of the needlines described within the architecture.

  📖 *A needline represents a collection of information exchanges.*

  ☛ *The NOV-3 report focuses on needlines and the information exchanges contained in these needlines. If no needlines are specified then all the needlines defined within the context of the architecture are taken into account.*
- The **Exchange Properties** (MetaAttributes) parameter is used to specify the interactions that should be included in the analysis.

  ☛ *If no exchange property is set, all the exchanges are taken into account.*

The NOV-3 Operational Information Exchange Dictionary

The Operational Information Exchange Dictionary chapter lists in alphabetical order all the information elements (contents) of the architecture, defined on the operational level.

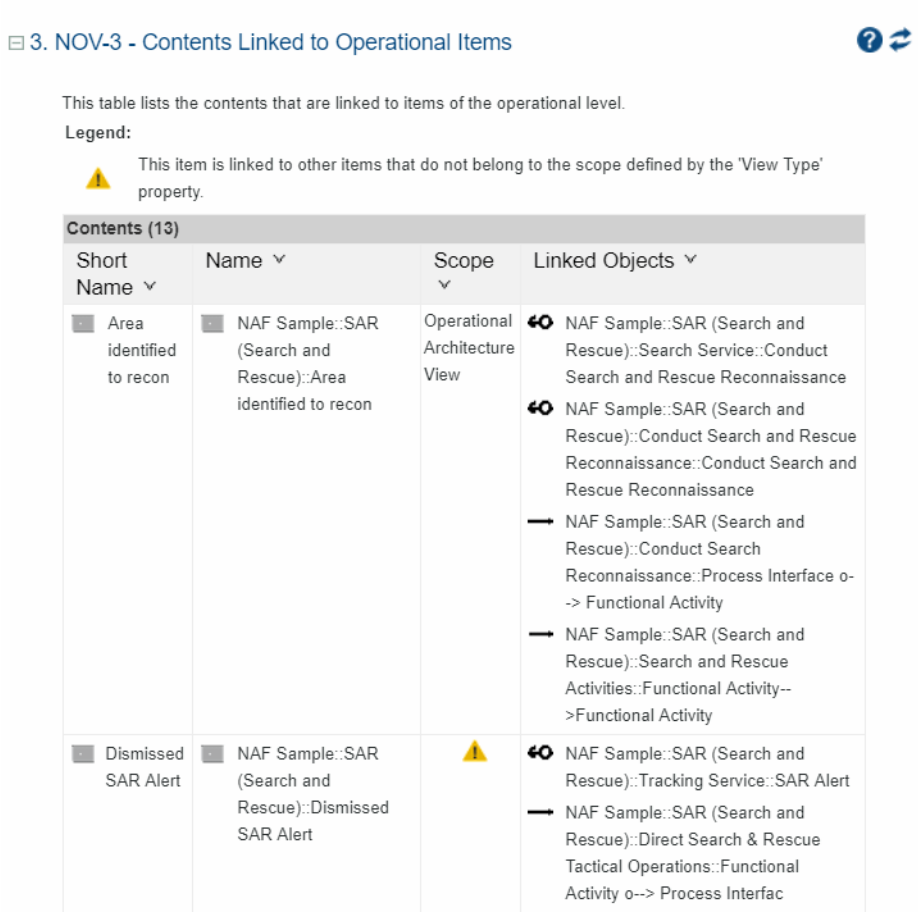
Each information element is then described in detail with a list of all the information exchanges in which it is involved. The source and target of the information is also included.

The NOV-3 Contents Linked to Operational Items Chapter

This chapter presents a table which lists the contents of the architecture that are linked to operational items.

This helps the user determine if the type of contents linked are defined according to the scope defined for this operational architecture.

This definition is based on the "View Type" property.



3. NOV-3 - Contents Linked to Operational Items

This table lists the contents that are linked to items of the operational level.

Legend:

This item is linked to other items that do not belong to the scope defined by the 'View Type' property.

Contents (13)

Short Name	Name	Scope	Linked Objects
Area identified to recon	NAF Sample::SAR (Search and Rescue)::Area identified to recon	Operational Architecture View	NAF Sample::SAR (Search and Rescue)::Search Service::Conduct Search and Rescue Reconnaissance NAF Sample::SAR (Search and Rescue)::Conduct Search and Rescue Reconnaissance::Conduct Search and Rescue Reconnaissance NAF Sample::SAR (Search and Rescue)::Conduct Search Reconnaissance::Process Interface o--> Functional Activity NAF Sample::SAR (Search and Rescue)::Search and Rescue Activities::Functional Activity--->Functional Activity
Dismissed SAR Alert	NAF Sample::SAR (Search and Rescue)::Dismissed SAR Alert		NAF Sample::SAR (Search and Rescue)::Tracking Service::SAR Alert NAF Sample::SAR (Search and Rescue)::Direct Search & Rescue Tactical Operations::Functional Activity o--> Process Interfac

Example of a Contents Linked to Operational Items Chapter

The NOV-3 Operational Information Exchange Matrix Chapter

This chapter identifies and displays the information elements and relevant attributes of information exchanges.

These exchanges are associated to their producer and consumer operational nodes and activities and to the interactions that they satisfy.

2. NOV-3 - Operational Information Exchange Matrix

This table lists for each operational content the interactions they belong to and their information exchanges.

Operational Contents						
Information Element	Interaction	Sending Operational Node	Receiving Operational Node	Information Exchange	Sending Operational Activity	Receiving Operational Activity
Area identified to recon				Process Interface o-> Functional Activity	Process Interface	Functional Activity
	Search Service	Tactical Control Center (Tactical Control Center)	Search Node (Search Node)			
	Conduct Search and Rescue Reconnaissance	Asset Controller (Asset Controler)	Search Node (Search Node)	Functional Activity-->Functional Activity	Functional Activity	Functional Activity
Distress Signal				Functional Activity o--> Distress Signal Receiver	Functional Activity	Distress Signal Receiver

Example of an Operational Information Exchange Matrix Chapter

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.

Designing Organizations

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

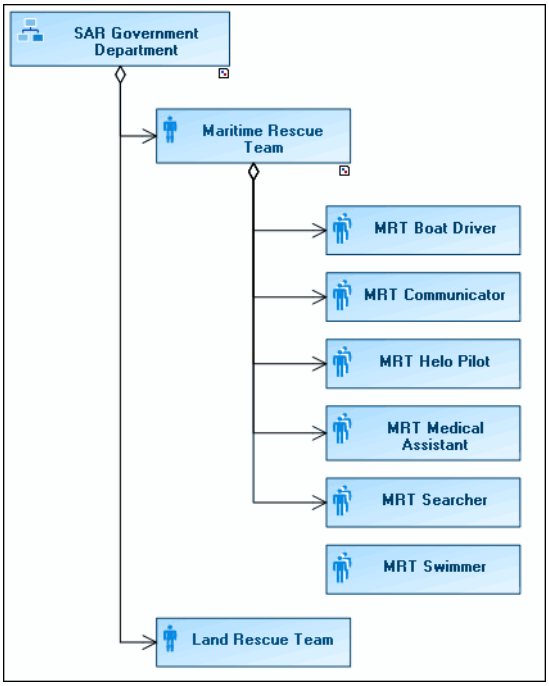
A global organization chart can be created to graphically show the hierarchy of org-units. If the hierarchy is too complex to be shown in a simple diagram, sub organizational charts can be added from any sub org-unit. Org-units are linked by dependency links that can be tuned to express a hierarchical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

The NOV-4 Report Template

Like all the other subviews, the NOV-4 report template comes with an architecture parameter that is used to specify the architecture to be described.

This parameter is completed by an optional parameter: **Org-Unit Subset**:

- If no value is set, all the org-units defined in the architecture are taken into account.
- If values are specified, only the given org-units are used to generate the report.

The NOV-4 Org-Unit Composition Hierarchy Chapter

This chapter details the hierarchy of org-units.

For each org-unit the type and comment are displayed. The table is composed of expandable items that reveal the complete structure of the root org-units.

If org-units are set in the Org-Unit Subset parameter, only these org-units are used to generate the hierarchy. They are used as the roots of the generated structures.

If a set org-unit is contained in a structure of another set org-unit then the structure for this org-unit is not repeated.

All the organizational charts including at least one of the cited org-units are inserted in the chapter.

The NOV-4 Org-Unit Specialization Hierarchy Chapter

The Org-Unit specialization hierarchy chapter shows how the different org-units of the architecture are specialized.

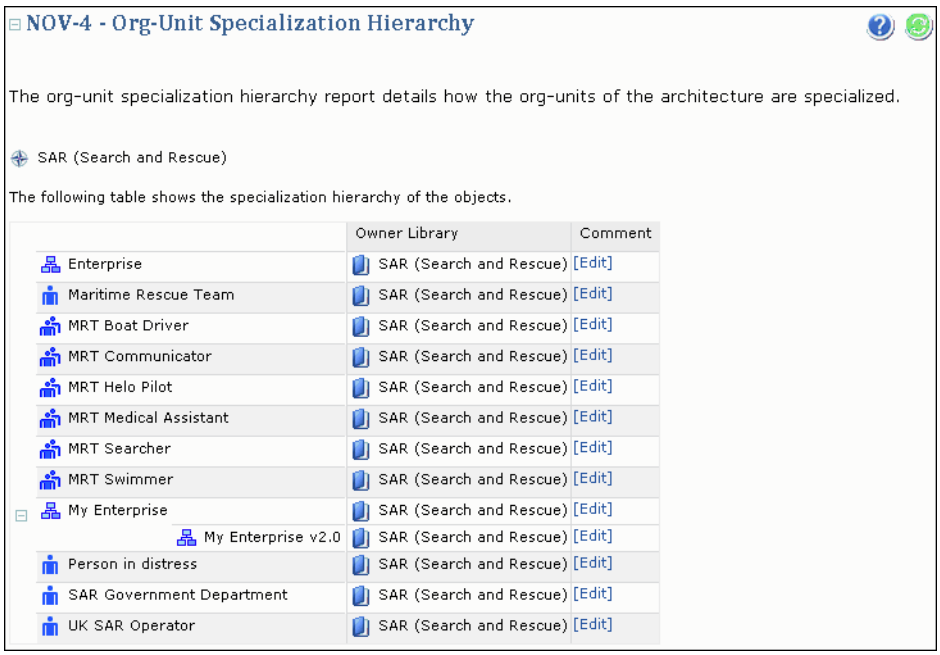
The specializations actually refer to variants of org-units. These variants are created to add more detail and specifics to fit particular scenarios of the architecture. They can be created by using the **New > Variant** command of the org-unit to be varied (specialized).

To show the specializations, the report displays:

- a table with the varied objects,
- their variants,
- the libraries to which they belong
- their comments.

> ☛ *It is important to note that not only org-units with variants are displayed in the report.*

All the org-units of the architecture are displayed if no org-unit subset was specified in the report template.



NOV-4 - Org-Unit Specialization Hierarchy

The org-unit specialization hierarchy report details how the org-units of the architecture are specialized.

SAR (Search and Rescue)

The following table shows the specialization hierarchy of the objects.

	Owner Library	Comment
Enterprise	SAR (Search and Rescue)	[Edit]
Maritime Rescue Team	SAR (Search and Rescue)	[Edit]
MRT Boat Driver	SAR (Search and Rescue)	[Edit]
MRT Communicator	SAR (Search and Rescue)	[Edit]
MRT Helo Pilot	SAR (Search and Rescue)	[Edit]
MRT Medical Assistant	SAR (Search and Rescue)	[Edit]
MRT Searcher	SAR (Search and Rescue)	[Edit]
MRT Swimmer	SAR (Search and Rescue)	[Edit]
My Enterprise	SAR (Search and Rescue)	[Edit]
My Enterprise v2.0	SAR (Search and Rescue)	[Edit]
Person in distress	SAR (Search and Rescue)	[Edit]
SAR Government Department	SAR (Search and Rescue)	[Edit]
UK SAR Operator	SAR (Search and Rescue)	[Edit]

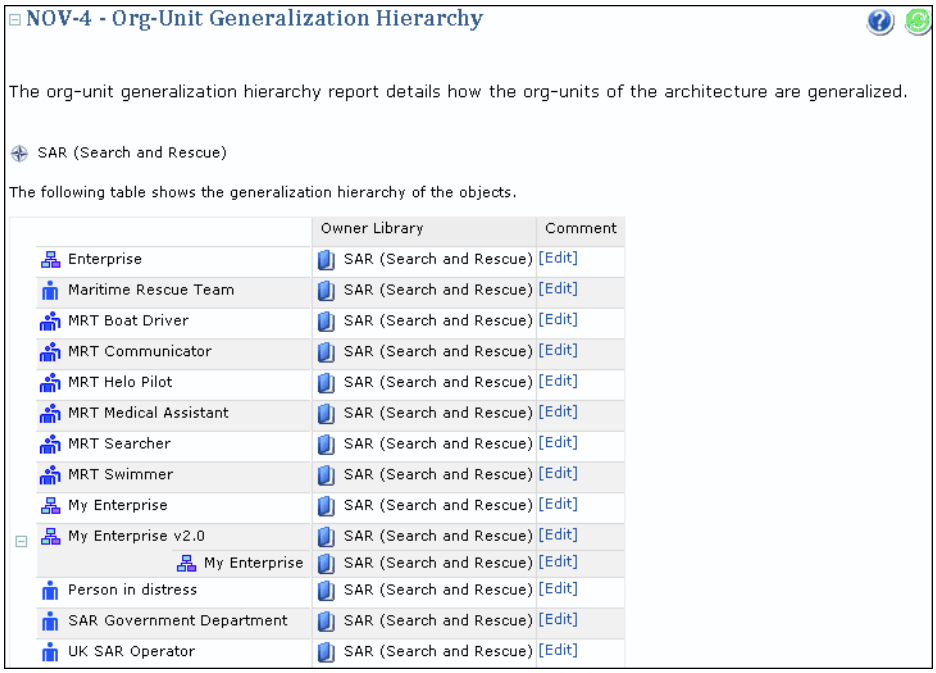
Example of an Org-Unit Specialization Hierarchy Chapter

The NOV-4 Org-Unit Generalization Hierarchy Chapter

This chapter shows how the different org-units of the architecture are generalized. The variants of org-units are presented in table form along with the varied org-units (org-units from which the variant was created).

The comments and library to which the org-units belong are also presented.

- To add a new generalization, use the **Connect > Variant Of** command of the specialized item (variant).



NOV-4 - Org-Unit Generalization Hierarchy

The org-unit generalization hierarchy report details how the org-units of the architecture are generalized.

SAR (Search and Rescue)

The following table shows the generalization hierarchy of the objects.

	Owner Library	Comment
Enterprise	SAR (Search and Rescue)	[Edit]
Maritime Rescue Team	SAR (Search and Rescue)	[Edit]
MRT Boat Driver	SAR (Search and Rescue)	[Edit]
MRT Communicator	SAR (Search and Rescue)	[Edit]
MRT Helo Pilot	SAR (Search and Rescue)	[Edit]
MRT Medical Assistant	SAR (Search and Rescue)	[Edit]
MRT Searcher	SAR (Search and Rescue)	[Edit]
MRT Swimmer	SAR (Search and Rescue)	[Edit]
My Enterprise	SAR (Search and Rescue)	[Edit]
My Enterprise v2.0	SAR (Search and Rescue)	[Edit]
My Enterprise	SAR (Search and Rescue)	[Edit]
Person in distress	SAR (Search and Rescue)	[Edit]
SAR Government Department	SAR (Search and Rescue)	[Edit]
UK SAR Operator	SAR (Search and Rescue)	[Edit]

The NOV-4 Org-Unit Dictionary Chapter

This chapter creates an alphabetical list of:

- all org-units mentioned in the Org-Unit Subset parameter
- all org-units of the architecture if no subset is mentioned.

For each org-unit in the table, the list of direct sub-org-units is displayed.

Diagrams are also inserted in the chapter.

The NOV-4 Competence Dictionary Chapter

This chapter presents list of all the competences of the architecture. The org-units that are linked to the different competences are also displayed.

The NOV-4 Org-Unit Responsibility Chapter

This report chapter lists the org-units of the architecture or those mentioned in the Org-unit subset of the analysis report.

It details the operations that are performed by these org-units and indicates the organizational processes for which they are responsible.

Statistical values concerning the maximum, minimum and average number of operations performed per org-unit and the equivalent values for the assigned processes are also displayed.

Organizational processes are assigned in the Properties dialog box of the Org-units, Responsibility section.

The operations performed by org-units are retrieved in the report through the use of participants. Org-units are attached to participants in the Organizational Process Diagram. Operations are then added to participants in the diagrams.

The list of Organizational Processes is found in the NOV-6c subview. For more information, see NOV-6c Operational Event-Trace Description.

The NOV-4 Type and Actual Organizations Mapping Overview Chapter

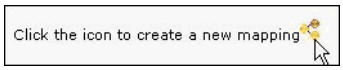
The org-unit "type" is the generic org-unit or an org-unit template created for a particular purpose and on which actual org-units can be based.

Org-units can then be created and mapped to these "types". These are the "actual" organizational resources.

This chapter gives an overview of the mapping levels that exist between org-unit types and the actual organizational resources (org-units). It also gives the percentage of mapping compliance between the types and actual org-units.

To create a mapping:

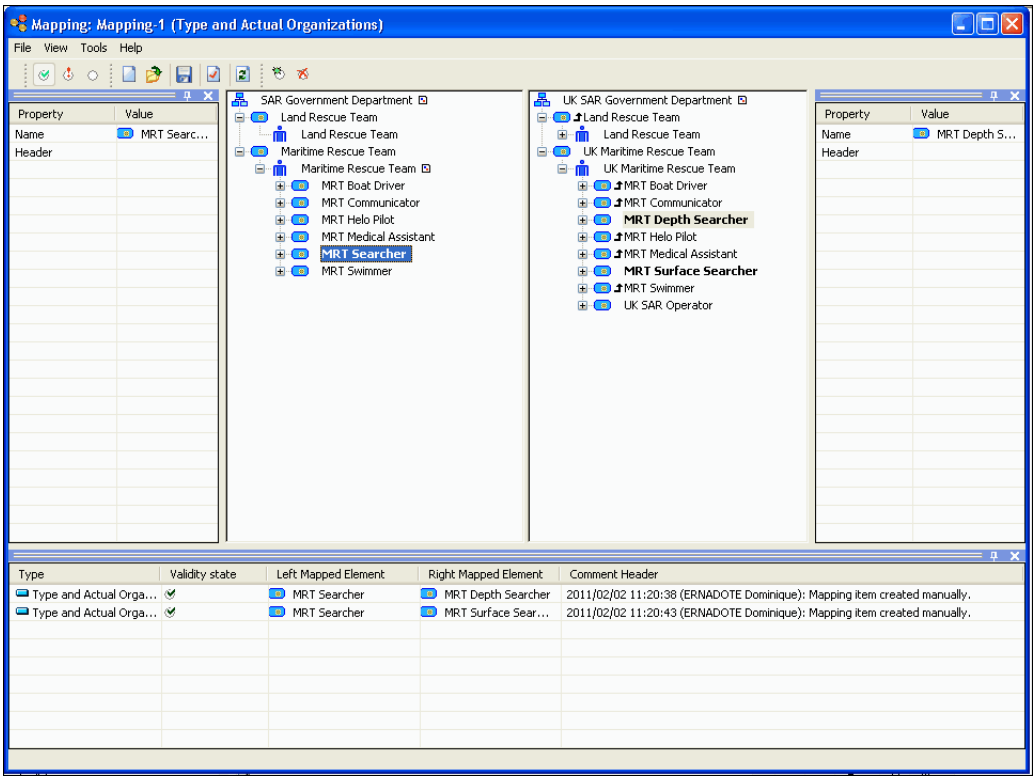
1. Click the corresponding icon from the report chapter.



2. Select the deployed item then the deployment item and click **OK**.
3. Right-click the mapping create from the report and select **Mapping Editor**.

In the figure below, the two middle columns contain the generic and the actual org-units: "types" to left and "actual" to the right. The generic "SAR Government Department" organization is mapped to the "UK SAR Government Department" (mapping not indicated) and the "MRT Searcher" org-unit component type is mapped to the "MRT

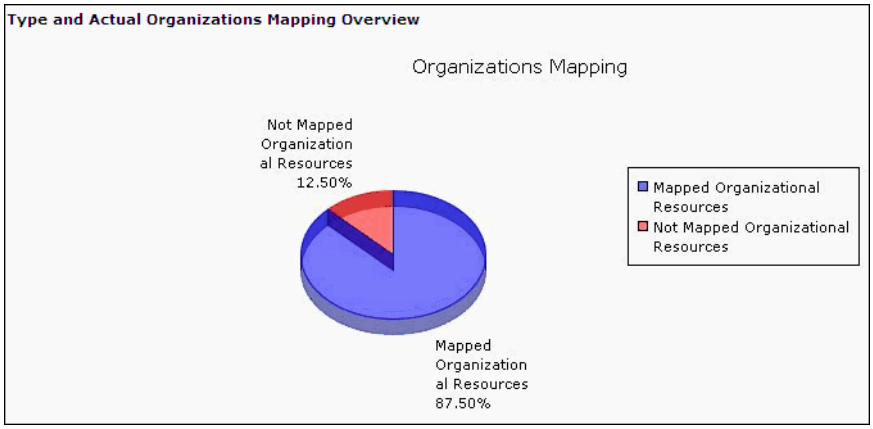
Depth Searcher" and the "MRT Surface Searcher" org-unit components.



☛ *To verify if a mapping exists for an org-unit type, right-click the type and click **Locate**. If no mapping exists, the **Locate** command is grayed.*

The Org-Unit Type and Actual Organizations Mapping Overview chapter includes:

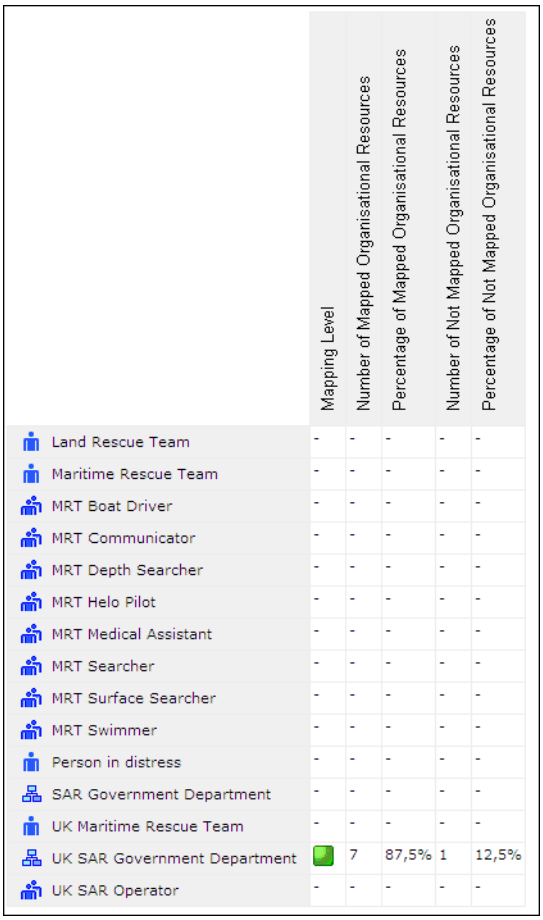
- a pie chart that indicates the percentage of mapped and unmapped organizational resources.



- a list of the organizational resources of the architecture
- a table that displays the mapping for the highest level of organizational resources (the organization).

The figure below shows that mappings exist for the "UK SAR Government Department" organization: 7 org-units are

mapped, which account for 87.5% of the organization's organizational resources.



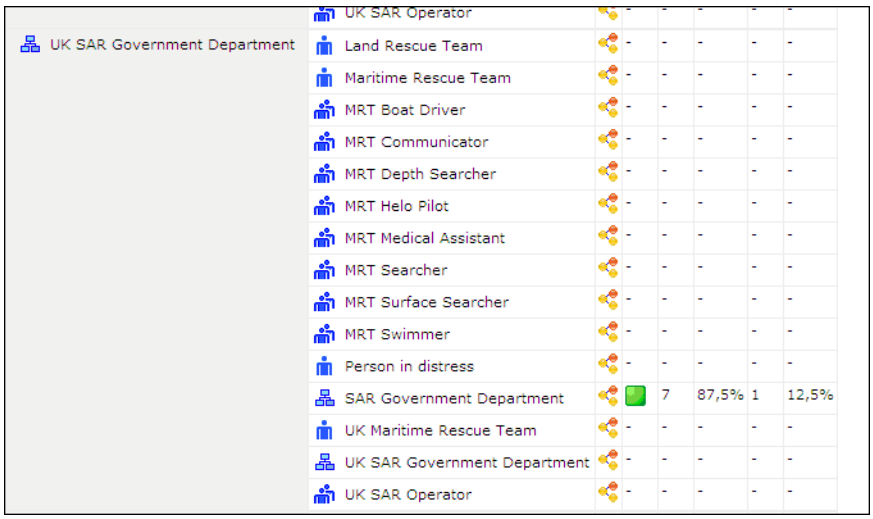
	Mapping Level	Number of Mapped Organisational Resources	Percentage of Mapped Organisational Resources	Number of Not Mapped Organisational Resources	Percentage of Not Mapped Organisational Resources
Land Rescue Team	-	-	-	-	-
Maritime Rescue Team	-	-	-	-	-
MRT Boat Driver	-	-	-	-	-
MRT Communicator	-	-	-	-	-
MRT Depth Searcher	-	-	-	-	-
MRT Helo Pilot	-	-	-	-	-
MRT Medical Assistant	-	-	-	-	-
MRT Searcher	-	-	-	-	-
MRT Surface Searcher	-	-	-	-	-
MRT Swimmer	-	-	-	-	-
Person in distress	-	-	-	-	-
SAR Government Department	-	-	-	-	-
UK Maritime Rescue Team	-	-	-	-	-
UK SAR Government Department		7	87,5%	1	12,5%
UK SAR Operator	-	-	-	-	-

The NOV-4 Type and Actual Organizations Mapping Details Chapter

This chapter gives more details of the mapping between the org-unit types and the actual organizational resources.

It shows the mapping levels of the different org-unit types with their actual organizations. This information is presented in table form.

In the figure below, the "UK SAR Government Department" organization is mapped to the "SAR Government Department" type.



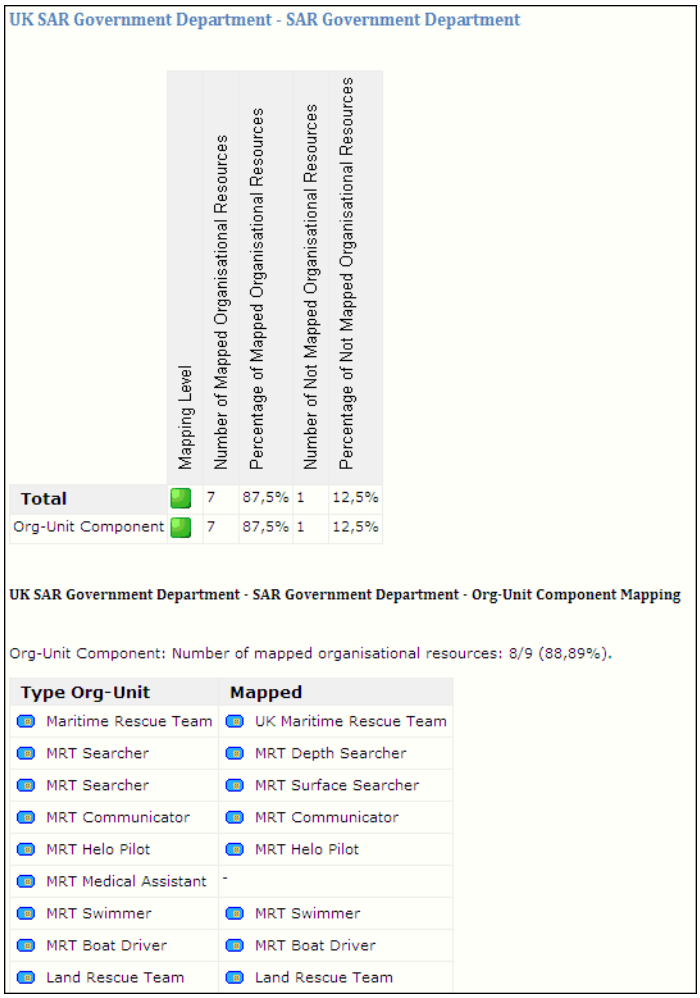
The NOV-4 Type and Actual Organizations Mapping Deep Details Chapter

This chapter gives deep details of the mapping between the organization type and the actual organization. A list of the mapped organizational resources is provided.

In this chapter each organizational resource (org-unit) has a dedicated paragraph with a table that indicates its mapping with a type.

The mapped organizational resources (org-unit components) of this organizational resource are indicated with their corresponding type.

The figure below indicates the mapping between the organizational resources of an organization with the corresponding org-unit types.



UK SAR Government Department - SAR Government Department

	Mapping Level	Number of Mapped Organisational Resources	Percentage of Mapped Organisational Resources	Number of Not Mapped Organisational Resources	Percentage of Not Mapped Organisational Resources
Total		7	87,5%	1	12,5%
Org-Unit Component		7	87,5%	1	12,5%

UK SAR Government Department - SAR Government Department - Org-Unit Component Mapping

Org-Unit Component: Number of mapped organisational resources: 8/9 (88,89%).

Type Org-Unit	Mapped
Maritime Rescue Team	UK Maritime Rescue Team
MRT Searcher	MRT Depth Searcher
MRT Searcher	MRT Surface Searcher
MRT Communicator	MRT Communicator
MRT Helo Pilot	MRT Helo Pilot
MRT Medical Assistant	-
MRT Swimmer	MRT Swimmer
MRT Boat Driver	MRT Boat Driver
Land Rescue Team	Land Rescue Team

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.

Creating Functional Process Diagrams

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

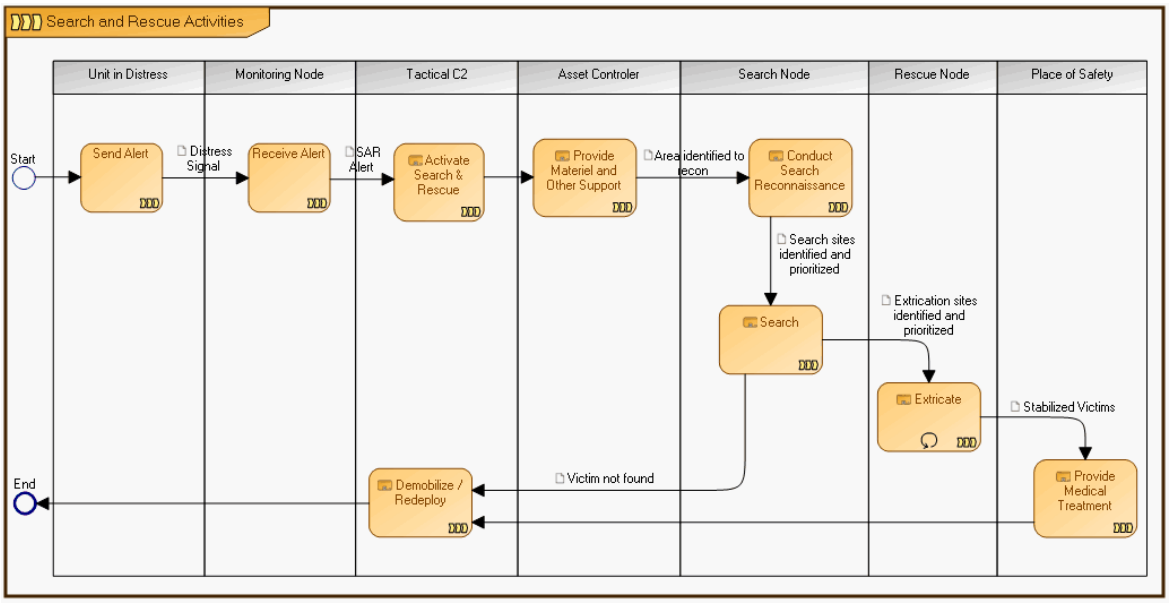
A functional process is an end-to-end collection of functional activities that creates an outcome for a customer, who may be the ultimate customer or an internal end-user of the functional process.

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 (see figure below). 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.

The figure below is an example of a Functional Process Diagram. It displays the Search process of the architecture. In this figure the process contains activities that are assigned to different operational nodes.

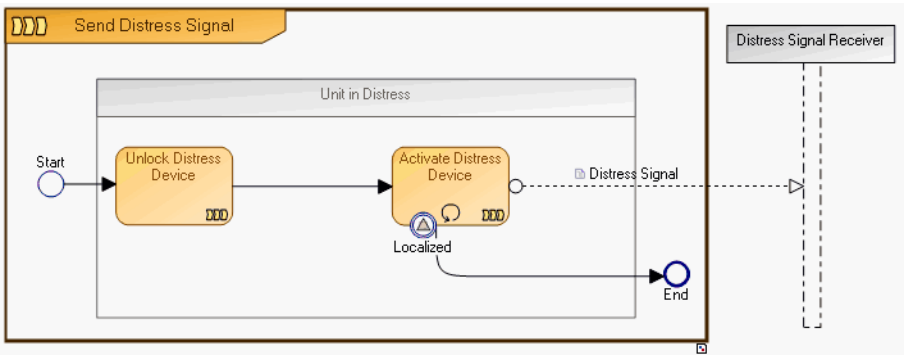
☛ *Note that activities can sometimes call different processes. See the **HOPEX Business Process Analysis** documentation for more information on creating process diagrams.*



Example of a Functional Process diagram

The figure below shows the details of the **Send Distress Signal** activity through another dedicated Functional

Process diagram. In this figure, the described activity contains two sub-activities.



A Detailed Illustration of the NOV-5 Send Distress Signal Activity

To create the "Functional Process Diagram":

1. In the NAF navigation tree, expand **NOV-5 Operational Activity Model > Root Functional Pocesses**.
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.

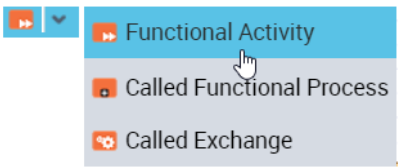
☛ *The purpose of the participant is not to describe activities but to help detail the external interactions of the process.*

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.

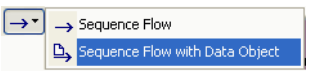
Sequence flows are used to indicate the steps to be followed (sequence of activities) in a process.

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.

Retrieving Information in the NOV-5 Report Chapters

To ensure that interesting information appears in the report chapters, and especially the NOV-5 Operational Activity Exchange and Operational Activity Exchange Balance chapters, the modeler should ensure that the functional processes to be studied and / or included in the analyses at least have the following data:

- There should be messages with content between at least two activities of a functional process
- The participants in functional processes should be associated to operational nodes.

  ☛ *The operational nodes are found in NOV-2. See NOV-2 Operational Node Connectivity Description.*

- The operational nodes associated to functional processes should have interactions that are based on the content of the functional processes.

It can be argued that the most interesting section of the NOV-5 Operational Activity Exchange chapter is the table that lists the different exchanges sent from or to functional processes or activities with the information element. If there are no messages with content between operational activities, there will be no exchange to show in the chapter.

Likewise if there are no incoming or outgoing exchanges to study there will be nothing to display in the NOV-5 Operational Activity Exchange Balance chapter.

The NOV-5 Report Template

The NOV-5 report template is supplied with the following parameters:

- The **Architecture** parameter defines the architecture to be analyzed.
- The **Operational Activity Subset**, which is an optional parameter, is used to reduce the scope of the study.

  ☛ *This parameter can be set with either just a few activities or a few functional processes. If it is not set, all activities are taken into account.*

The NOV-5 Operational Activity Composition Hierarchy Chapter

This chapter shows the structure of functional processes through the operational (functional) activities they contain.

When functional processes are selected, the trees are decomposed as follows: functional process to activities to sub-activities and so on.

The NOV-5 Operational Activity Specialization Hierarchy Chapter

The NOV-5 operational activity specialization hierarchy chapter shows how the functional processes of the architecture are specialized.

The functional processes are presented in a table along with their comments and the library to which they belong. If variants of functional processes are created, the varied functional processes are also presented in the table along with their variants.

To create new specializations (variants):

- Use the **New > Variant** command on the functional process to be varied (specialized).

The NOV-5 Operational Activity Generalization Hierarchy Chapter

The NOV-5 operational activity generalization hierarchy chapter shows how the functional processes of the architecture are generalized.

The functional processes are presented in table form along with their comments and the library to which they belong. If a variant of a functional process exists, this variant is also presented in the table along with the varied functional process (functional process from which the variant was created).

To add a new generalization:

- Use the **Connect > Variant Of** command of the specialized item (variant).

The NOV-5 Operational Activity Dictionary Chapter

This chapter displays a list of all the functional processes and activities used in the study.

Each activity and functional process is displayed with its short name, long name and comment.

The report includes:

- a list of the all the functional processes in the architecture.
- a detailed paragraph for each functional process with:
  - the describing functional process diagram
  - a list of the involved activities
  - sub processes if they exist
  - called functional processes
- A list of all the activities in the architecture with paragraphs to describe each activity.
- a detailed paragraph for each activity with
  - the describing diagram
  - sub-activities if they exist

☛ *The content of each paragraph varies depending on the type of item being described.*

The NOV-5 Operational Activity Exchange Chapter

This chapter:

- shows the diagrams that display information exchanges between functional processes and activities.
- details the information exchange between the selected activities.

> ☛ *All messages or interactions connecting activities are displayed in a list with their name, comment and source and target activities.*

The following table lists the exchanges from or targeting functional processes or activities with their information element.

Exchanges

Name	Sources	Targets	Information Elements
Functional Activity o--> Functional Activity	Functional Activity	Functional Activity	Medical Protocols
Functional Activity o--> Process Interface	Functional Activity	Process Interface	Extrication sites identified and prioritized
Process Interface o--> Functional Activity	Process Interface	Functional Activity	Area identified to recon
Process Interface o--> Functional Activity	Process Interface	Functional Activity	Extrication sites identified and prioritized
Functional Activity o--> Process Interface	Functional Activity	Process Interface	Victim not found
Functional Activity-->Functional Activity	Functional Activity	Functional Activity	Stabilized Victims
Process Interface o--> Functional Activity	Process Interface	Functional Activity	SAR Alert
Functional Activity-->Functional Activity	Functional Activity	Functional Activity	Victim not found
Process Interfac o--> Functional Activity	Process Interfac	Functional Activity	SAR Alert
Functional Activity o--> Process Interfac	Functional Activity	Process Interfac	Dismissed SAR Alert

The NOV-5 Operational Activity Exchange Balance Chapter

This chapter helps determine the compliance of the exchanges defined within functional processes and the exchanges received and/or sent by these functional processes.

It lists all the exchanges modeled within and outside each functional process between the contained and detailing activities.

Different icons with arrows are used to give valid information relating to each message and interaction. These arrows help determine what is missing from and what is unnecessary for the exchange design.

The information of the chapter is presented in table form with a list of the operational activities of the architecture. For each operational activity the balance between the incoming and outgoing information is shown.

NOV-5 - Operational Activity Exchange Balance

The operational activity exchange balance report checks the correctness of the exchange design.
A comparison of all the incoming and outgoing exchanges defined inside and outside the processes allows to find out what is missing in the design and what is not useful.

SAR (Search and Rescue)

The following table shows for each operational activity the balance between around and inside information elements incoming or outgoing the operational activity.

Legend:

- The incoming information element is managed around the balanced item.
- The incoming information element is managed inside the balanced item.
- The incoming information element is managed both around and inside the balanced item.
- The outgoing information element is managed around the balanced item.
- The outgoing information element is managed inside the balanced item.
- The outgoing information element is managed both around and inside the balanced item.
- The around exchange is coming in the balanced item.
- The inside exchange is coming in the balanced item.
- The around exchange is going out the balanced item.
- The inside exchange is going out the balanced item.

Exchange Balance

Balanced Items	Information Elements	Around Exchanges	Inside Exchanges
Activate Search & Rescue (Activate Search & Rescue)	SAR Alert	Receive Alert Receive Alert o--> Activate Search & Rescue	Process Interface o--> Receive Search and Rescue (SAR) alert Receive Search and Rescue (SAR) alert
Conduct Search Reconnaissance (Conduct Search Reconnaissance)	Area identified to recon	Provide Material and Other Support Provide Material and Other Support o--> Conduct Search Reconnaissance	Process Interface o--> Assess incident site Assess incident site
	Search sites identified and prioritized	Conduct Search Reconnaissance o--> Search Search	Develop map of search area Develop map of search area o--> Process Interface
Demobilize / Redeploy (Demobilize / Redeploy)	Victim not found	Search Search o--> Demobilize / Redeploy	

Example of an NOV-5 Operational Activity Exchange Balance Chapter

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.

The NOV-6a Report Template

The NOV-6a report template is supplied with one parameter only: the architecture for which the report is required. There are no additional parameters.

The template comes with two chapters:

- the NOV-6a Operational Rule Model chapter
- the NOV-6a Rules Linked to Operational Items chapter.

The NOV-6a Operational Rule Model Chapter

This chapter lists the constraints and requirements attached to the operational items of the architecture.

> 🕮 *Constraints are the external constraining elements that set the terms for whatever solutions are implemented.*
>
> 🕮 *Requirements are the needs and requests of the user that define the contract or agreement to be carried out.*

The lists of requirements and constraints are displayed in alphabetical order with an additional column for the comments. A paragraph is added for each listed constraint and requirement to explain their constrained objects.

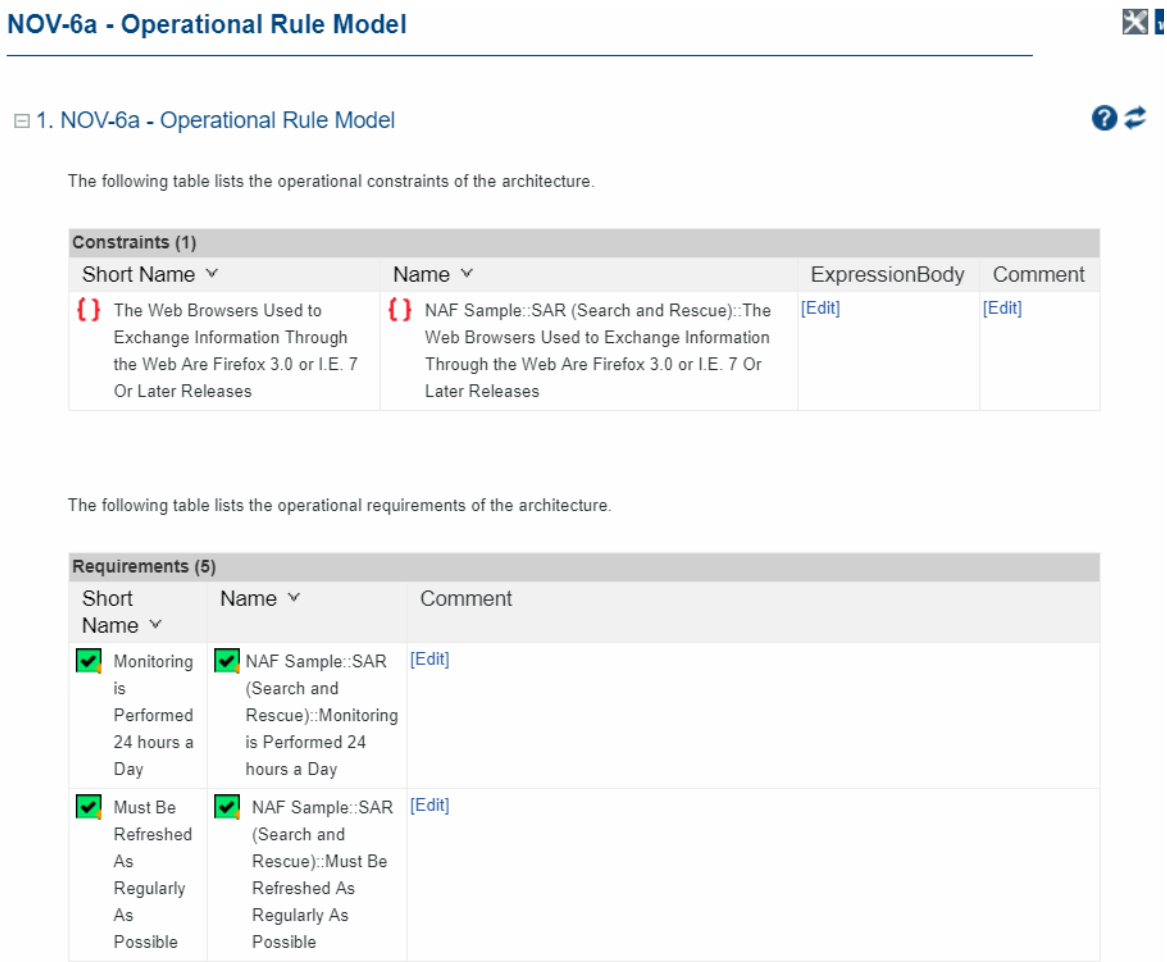
If constraints and requirements are created from objects in Operational node structure diagrams (NOV-2), these constraints and requirements can be retrieved

in the Optional Rules Model chapter provided the "View Type" property is set to Operational.

☛ *For more information, see NOV-2 Operational Node Connectivity Description.*

The constraints are retrieved in the Operational Constraints folder while the requirements are retrieved in the Operational Requirements folder of the NOV-6a navigation tree.

☛ *For more information, see NOV-6a Operational Rules Model.*



NOV-6a - Operational Rule Model

1. NOV-6a - Operational Rule Model

The following table lists the operational constraints of the architecture.

Constraints (1)

Short Name	Name	ExpressionBody	Comment
The Web Browsers Used to Exchange Information Through the Web Are Firefox 3.0 or I.E. 7 Or Later Releases	NAF Sample::SAR (Search and Rescue)::The Web Browsers Used to Exchange Information Through the Web Are Firefox 3.0 or I.E. 7 Or Later Releases	[Edit]	[Edit]

The following table lists the operational requirements of the architecture.

Requirements (5)

Short Name	Name	Comment
Monitoring is Performed 24 hours a Day	NAF Sample::SAR (Search and Rescue)::Monitoring is Performed 24 hours a Day	[Edit]
Must Be Refreshed As Regularly As Possible	NAF Sample::SAR (Search and Rescue)::Must Be Refreshed As Regularly As Possible	[Edit]

Example of the NOV-6a Operational Rule Model Chapter

The NOV-6a Rules Linked to Operational Items Chapter

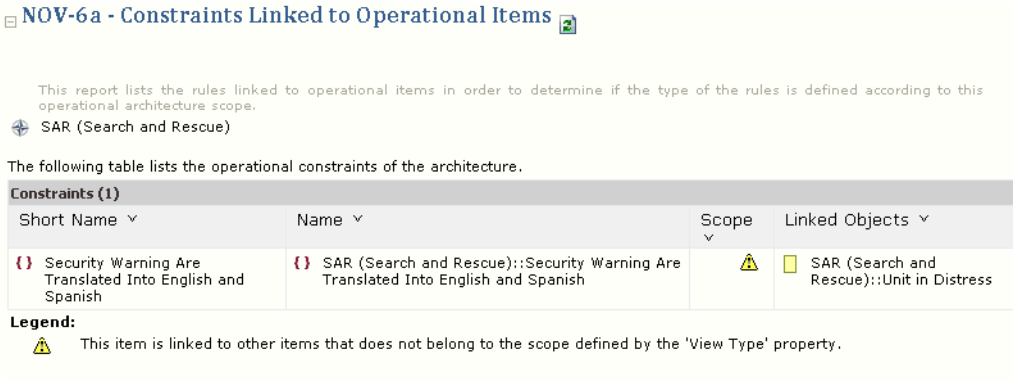
It is also possible to generate a chapter that lists the potential constraints linked to an operational item.

This helps the user determine if the type of constraints linked are defined according to the scope defined for this operational architecture.

☛ *This definition is based on the "View Type" property.*

Viewing the scope of the constraint

If a constraint is not typed as Operational, a warning icon is displayed for this constraint in the **Scope** column. Although not typed as operational, the constraint appears in the table because it is linked to an operational item. This happens if the constraint was not created from the NAF navigation tree but from a diagram and then attached to an operational item of this diagram.



NOV-6a - Constraints Linked to Operational Items

This report lists the rules linked to operational items in order to determine if the type of the rules is defined according to this operational architecture scope.

SAR (Search and Rescue)

The following table lists the operational constraints of the architecture.

Constraints (1)

Short Name	Name	Scope	Linked Objects
Security Warning Are Translated Into English and Spanish	SAR (Search and Rescue)::Security Warning Are Translated Into English and Spanish	⚠	SAR (Search and Rescue)::Unit in Distress

Legend:

⚠ This item is linked to other items that does not belong to the scope defined by the 'View Type' property.

Example of a Rules Linked to Operational Items Chapter

Setting a constraint as operational

The "View Type" property for constraints with a warning can be changed. To do so:

1. from the table, right-click the constraint and open its property page.
2. Select the NAF sub-page then **NAF State Level**.
3. From the **NAF Architecture View Type**, select "Operational Activity View".
4. Refresh the report to show the changes.

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.*

Using State Machines

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.

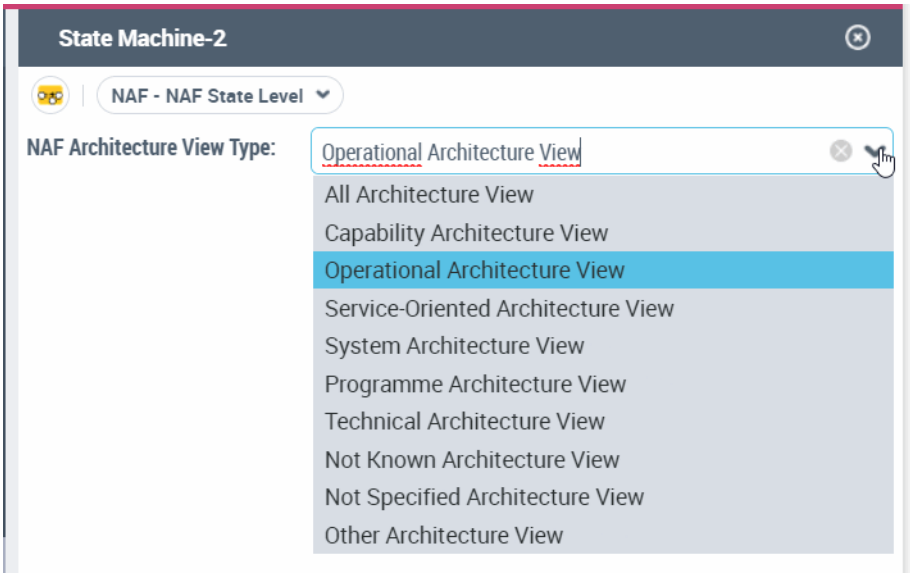
To create an operational architecture state machine:

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. From the drop-down list select the **NAF > NAF State Level** page.
3. Select a view type from the **NAF Architecture View** Type field.



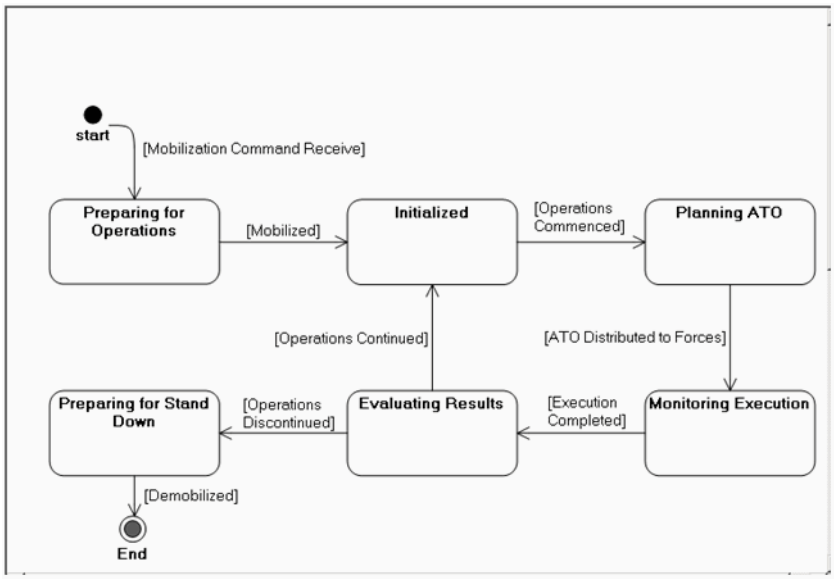
☛ *If you select "All Architecture View", the behavior also appears in the NSV-10b folder. See The NSV-10b Resources State Transition Description Chapter.*

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)

The NOV-6b Report Template

The NOV-6b report template comes with two parameters:

- The **Architecture** parameter, which specifies the architecture to be analyzed.
- The **Activity Subset** parameter, which is an optional parameter. This parameter is used to reduce the scope of the study to a subset of selected activities. In this case, only the state machines attached to the selected activities are considered. If this parameter is not set, all the state machines attached to an activity are taken into account.

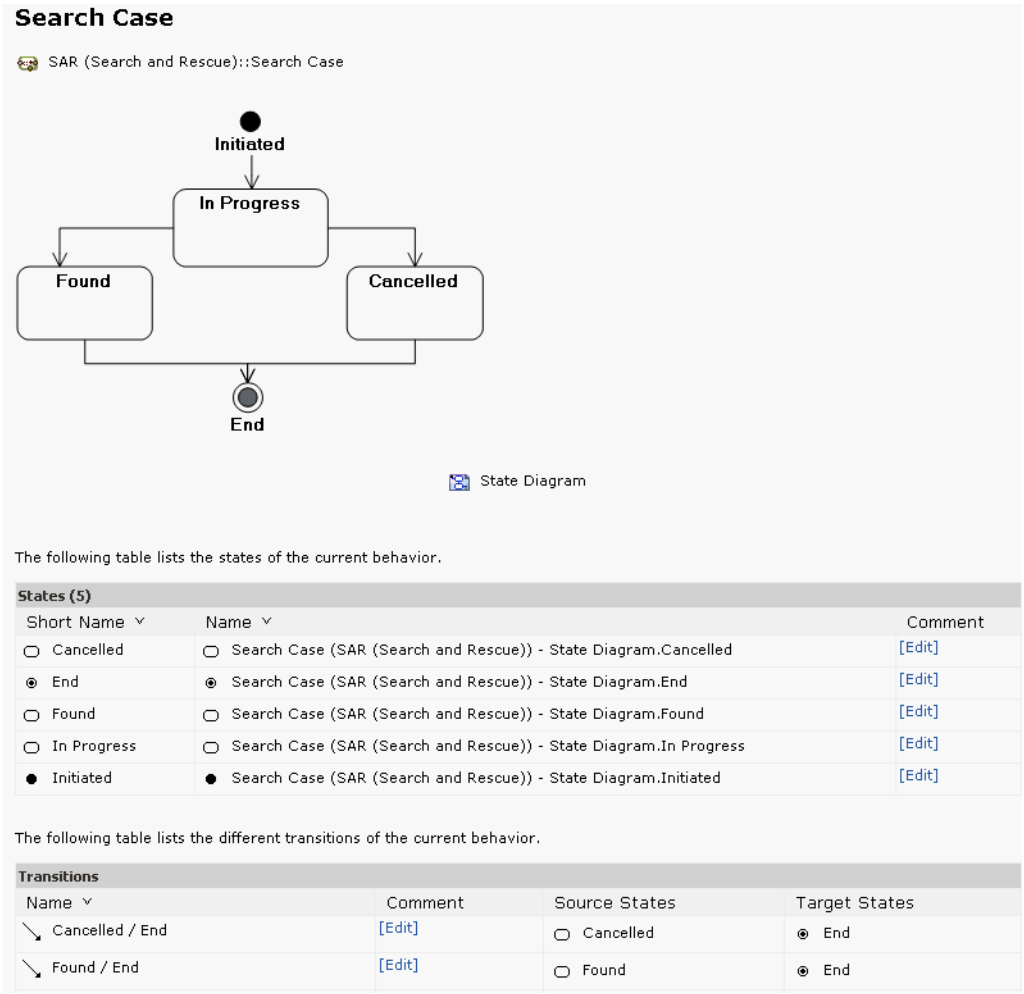
The NOV-6b Operational State Transition Description Chapter

This chapter lists all the state machines attached to the selected activities. The report starts with a simple table of the state machines (short name, long name, comment) and the linked activities.

For each state machine which has a describing diagram or at least one event, a dedicated paragraph is generated with its title being the name of the state machine.

This paragraph displays:

- the comment of the state machine
- the (commented) describing diagrams
- the events (states) with their name and comment in a table.
- the successions (transitions) with their name, comment and source and target events.



Search Case

SAR (Search and Rescue)::Search Case

State Diagram

The following table lists the states of the current behavior.

States (5)

Short Name	Name	Comment
Cancelled	Search Case (SAR (Search and Rescue)) - State Diagram.Cancelled	[Edit]
End	Search Case (SAR (Search and Rescue)) - State Diagram.End	[Edit]
Found	Search Case (SAR (Search and Rescue)) - State Diagram.Found	[Edit]
In Progress	Search Case (SAR (Search and Rescue)) - State Diagram.In Progress	[Edit]
Initiated	Search Case (SAR (Search and Rescue)) - State Diagram.Initiated	[Edit]

The following table lists the different transitions of the current behavior.

Transitions

Name	Comment	Source States	Target States
Cancelled / End	[Edit]	Cancelled	End
Found / End	[Edit]	Found	End

Example of an Operational State Transition Description Chapter

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 an ***organizational process***.

> 🕮 *An organizational 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 organizational processes properties, see the **HOPEX Business Process Analysis** documentation: Processes.*

- ✓ The NOV-6c Report Template
- ✓ The NOV-6c Operational Event-Trace Composition Hierarchy Chapter
- ✓ The NOV-6c Operational Event-Trace Specialization Hierarchy Chapter
- ✓ The NOV-6c Operational Event-Trace Generalization Hierarchy Chapter
- ✓ The NOV-6c Operational Event-Trace Dictionary Chapter
- ✓ The NOV-6c Operational Event-Trace Exchange Chapter
- ✓ The NOV-6c Operational Event-Trace Exchange Balance Chapter

Creating an Operational Interaction Scenario Diagram

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

The NOV-6c Report Template

The NOV-6c Report Template supplies reports to detail organizational process implementation. It is defined with three parameters:

- The **Architecture** parameter, which specifies the architecture to be analyzed.
- **Interaction Scenario Subset** parameter, which is optional. This parameter is set with Interaction scenarios and is used to reduce the scope of the study. If no value is set for this parameter, all the interaction scenarios are taken into account for the report.
- The **Organizational Process Subset** parameter, which is optional. This parameter is set with organizational processes and is used to reduce the scope of the study to those selected. If a value is not set, all the organizational processes are taken into account.

The NOV-6c Operational Event-Trace Composition Hierarchy Chapter

This chapter shows the hierarchical structure of the organizational processes through their operations.

☛ *The selected organizational process mentioned within a structure are not reconsidered when starting a new structure (to avoid redundant listing).*

The tree structure displays the name and comment of each organizational process in a table.

The NOV-6c Operational Event-Trace Specialization Hierarchy Chapter

The NOV-6c operational event-trace specialization hierarchy chapter shows how the organizational processes of the architecture are specialized.

The organizational processes are presented in a table along with their comments and the library to which they belong.

If a variant of an organizational process is created, the varied organizational process is presented in the table along with its variant.

☛ *To create new specializations (variants), use the **New > Variant** command on the organizational process to be varied (specialized).*

The NOV-6c Operational Event-Trace Generalization Hierarchy Chapter

The NOV-6c operational event-trace generalization hierarchy chapter shows how the organizational processes of the architecture are generalized.

The organizational processes are presented in table form along with their comments and the library to which they belong.

If a variant of an organizational process exists, this variant is also presented in the table along with the varied organizational process (organizational process from which the variant was created).

☛ *To add a new generalization, use the **Connect > Variant Of** command of the specialized item (variant).*

The NOV-6c Operational Event-Trace Dictionary Chapter

This chapter lists all the organizational processes with their details and all the interaction scenarios that are examples of how the architecture behaves from an operational perspective. These items are listed with their name and comment.

A paragraph is added for each organizational process which has at least some components or a describing diagram. The title of the paragraph is the name of the organizational process. This paragraph contains the comment, the list of involved components and the (commented) diagram.

Paragraphs are similarly generated for each interaction scenario.

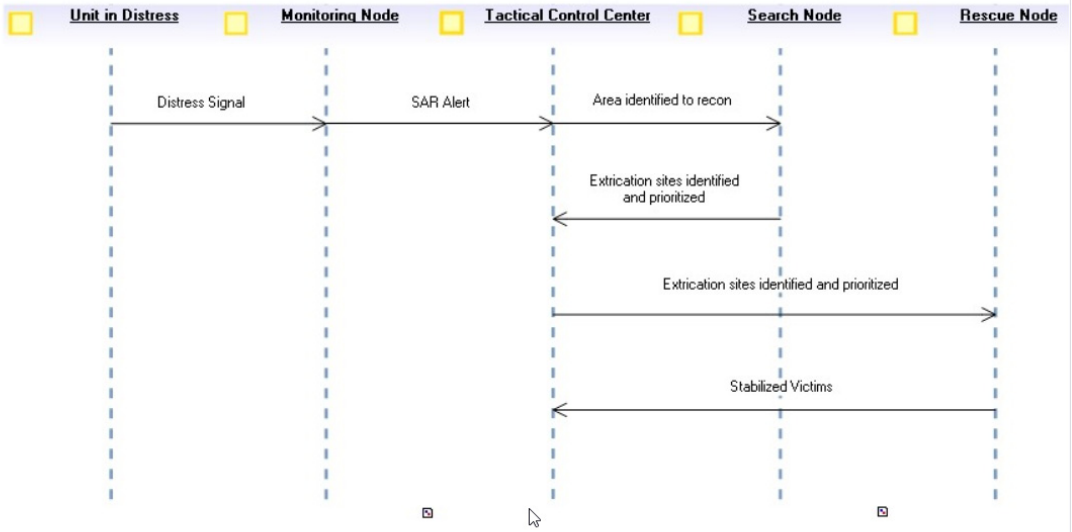
The NOV-6c Operational Event-Trace Exchange Chapter

This chapter is used to show the exchanges that occur between organizational processes and operations in the architecture.

It displays ***operational interaction scenario diagrams***:

5. NOV-6c - Operational Event-Trace Exchange

The following diagram shows the exchanges defined in the operational interaction scenarios.



Organizational Interaction Scenario Diagram

The exchanges are detailed in a table:

The following table lists the exchanges from or targeting organizational processes or operations with their information element.

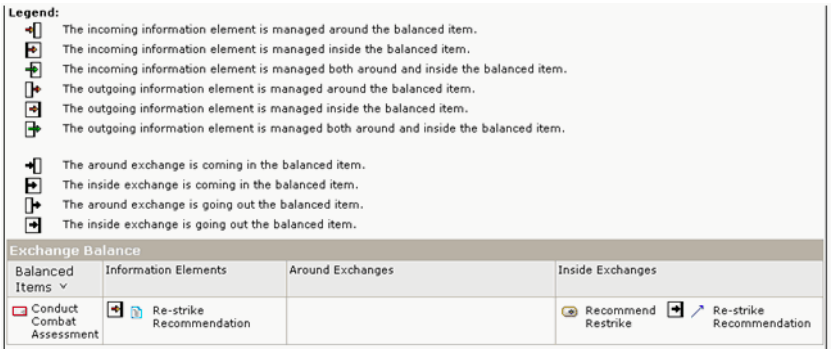
Exchanges

Name	Sources	Targets	Information Elements
Area identified to recon	Tactical Control Center	Search Node	Area identified to recon
Distress Signal	Unit in Distress	Monitoring Node	Distress Signal
Extrication sites identified and prioritized	Search Node	Tactical Control Center	Extrication sites identified and prioritized
Extrication sites identified and prioritized	Tactical Control Center	Rescue Node	Extrication sites identified and prioritized
SAR Alert	Monitoring Node	Tactical Control Center	SAR Alert
Stabilized Victims	Rescue Node	Tactical Control Center	Stabilized Victims

The NOV-6c Operational Event-Trace Exchange Balance Chapter

This chapter is used to show the balance between the exchanges defined within an organizational process and the exchanges it receives and/or sends out.

It lists all the internal and outside messages and interactions that detail the selected organizational process.



Example of an NOV-6c Operational Event Trace Exchange Balance Chapter

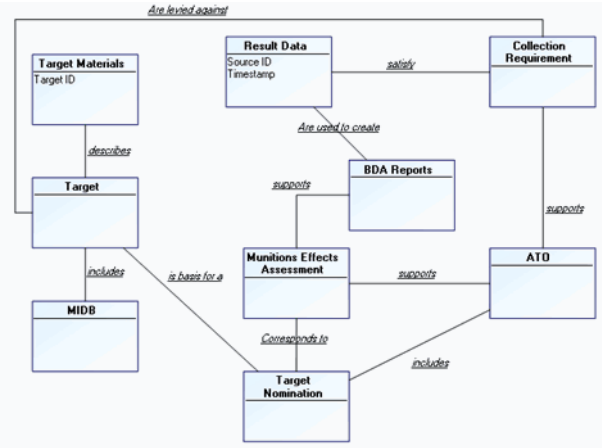
NOV-7 INFORMATION MODEL

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.

Creating Data Models

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.*

An information model should be based on the NAV-2 subview, which actually forms the operational domain object model and which contains the definitions of all the concepts that are relevant for the architecture effort.

☛ *See NAV-2 Integrated Dictionary.*

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.*

The NOV-7 Report Template

The NOV-7 report template is delivered with a report that describes data models.

The first parameter in this template is used to indicate the architecture to be analyzed.

The second parameter, **Data Model Subset**, is optional and is used to reduce the scope of the study to a subset of the data models of the architecture. If this parameter is not set, all the data models defined within the architecture are taken into account.

The NOV-7 Information Model Chapter

This chapter lists all the selected data models in a table with their short name, long name and comment.

Paragraphs specific to each data model are then displayed. These paragraphs contain the following items:

- The comment
- The describing diagrams
- The table of the entities with the attributes
- The list of associations linking the entities

The NOV-7 Information Model Hierarchy Chapter

This chapter presents in table form the hierarchy of all the data models of the architecture. A comment or definition is also displayed for the different data models.

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-3 Capability to Services Mapping
- ✓ NSOV-4a Service Constraints
- ✓ NSOV-4b Service State Model
- ✓ NSOV-4c Service Interaction Specification
- ✓ NSOV-5 Service Functionality
- ✓ NSOV-6 Service Composition

About Services

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 Managing Services.*

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.

Managing Services

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

See also: About Services.

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.*

The NSOV-1 Report Template

The NSOV-1 report template is delivered with a report that displays the service taxonomy represented as a hierarchy. This template is defined with two parameters:

- The **Architecture** parameter which indicates the architecture to be analyzed.
- The **Business Service** parameter, which is optional. This parameter is used to reduce the scope of the study to the specified business services of the architecture. If this parameter is not set, all the business services defined within the architecture are taken into account.

The NSOV-1 Service Specialization Hierarchy Chapter

This chapter shows how the different services of the architecture are specialized. In fact variants of certain services are created to add more detail and specifics to fit and support particular scenarios of the architecture.

To show these variants, the Specialization Hierarchy chapter presents the varied objects and their variants in a table. The libraries to which the services belong as well as the comments of these objects are also indicated.

☛ *To create new specializations (variants), use the **New > Variant** command on the service to be varied (specialized).*

The NSOV-1 Service Generalization Hierarchy Chapter

This chapter shows how the different services of the architecture are generalized.

The variants of services are presented in table form along with the varied services (service from which the variant was created). The comments and library to which the services belong are also presented.

☛ *To add a new generalization, use the **Connect > Variant Of** command of the specialized item (variant).*

The NSOV-1 Service Dictionary Chapter

The Service Dictionary chapter provides an alphabetical list of the services defined in the architecture.

It also specifies the service category (here, operational).

⊟ 3. NSOV-1 - Service Dictionary

The following table lists the services of the architecture.

Services (8)

Short Name	Name	Service Category	Comment
Land Search and Rescue	NAF Sample::SAR (Search and Rescue)::Land Search and Rescue	Operational	[Edit]
Maritime Rescue	NAF Sample::SAR (Search and Rescue)::Maritime Rescue	Operational	[Edit]
Maritime Search	NAF Sample::SAR (Search and Rescue)::Maritime Search	Operational	[Edit]
Maritime Search and Rescue	NAF Sample::SAR (Search and Rescue)::Maritime Search and Rescue	Operational	[Edit]
Rescue	NAF Sample::SAR (Search and Rescue)::Rescue	Operational	[Edit]
Search	NAF Sample::SAR (Search and Rescue)::Search	Operational	[Edit]
Search and Rescue	NAF Sample::SAR (Search and Rescue)::Search and Rescue	Operational	[Edit]
Search Sites	NAF Sample::SAR (Search and Rescue)::Search Sites	[Edit]	[Edit]

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.*

Creating Service Definitions

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.

The NSOV-2 Report Template

The NSOV-2 Report Template supplies a dictionary chapter that details the services defined to support the different operational activities of the architecture. It is defined with two parameters:

- The **Architecture** parameter, which specifies the architecture to be analyzed.
- **Service Definitions** parameter, which is optional. This parameter is set with service definitions (protocols) and is used to reduce the scope of the study. If no value is set for this parameter, all the service definitions are taken into account for the Report.

The NSOV-2 Service Definition Hierarchy Chapter

The Service Definition Hierarchy chapter presents a table with the structure of the service definitions in the architecture. The comments for these service definitions are also displayed.

NSOV-2 - Service Definition Hierarchy

The service definition hierarchy report details the structure of the service definitions

SAR (Search and Rescue)

The following table shows the hierarchy of the service definitions.

	Comment
Asset Delivery	[Edit]
Asset Provision Initialization	[Edit]
Conduct Search and Rescue Reconnaissance	[Edit]
Distress Beacon Activation	[Edit]
Distress Signalization	[Edit]
Extricate	[Edit]
IMO Distress Service	[Edit]
Initiate Rescue	[Edit]
Land Search and Rescue Service	[Edit]
Maritime Rescue	[Edit]
Reassurance (As in Maritime Rescue)	[Edit] **Interaction:** [Edit]
Recover Victim (As in Maritime Rescue)	[Edit] **Interaction:** [Edit]
Maritime Search	[Edit]
Material Support	[Edit]
Asset Delivery (As in Material Support)	[Edit] **Interaction:** [Edit]
Asset Provision Initialization (As in Material Support)	[Edit] **Interaction:** [Edit]
Medical Assistance	[Edit]
Piloting Service	[Edit]
Reassurance	[Edit]
Recover Victim	[Edit]

Example of an NSOV-2 Service Definition Hierarchy Chapter

The NSOV-2 Service Definitions Dictionary Chapter

The service definition dictionary report is an alphabetical list of the service definitions defined for the architecture.

2. NSOV-2 - Service Definition Dictionary

Service Definitions (32)

Short Name	Name	Comment
Architecture Governance	NAF Sample::SAR (Search and Rescue)::Architecture Governance	[Edit]
Asset Delivery	NAF Sample::SAR (Search and Rescue)::Asset Delivery	[Edit]
Asset Provision Initialization	NAF Sample::SAR (Search and Rescue)::Asset Provision Initialization	[Edit]
Cancel Search	NAF Sample::SAR (Search and Rescue)::Search::Cancel Search	[Edit]
Conduct Search and Rescue Reconnaissance	NAF Sample::SAR (Search and Rescue)::Conduct Search and Rescue Reconnaissance	[Edit]
Distress Beacon Activation	NAF Sample::SAR (Search and Rescue)::Distress Beacon Activation	[Edit]
Distress Signalization	NAF Sample::SAR (Search and Rescue)::Distress Signalization	[Edit]
Extricate	NAF Sample::SAR (Search and Rescue)::Extricate	[Edit]
IMO Distress Service	NAF Sample::SAR (Search and Rescue)::IMO Distress Service	[Edit]
Initiate Rescue	NAF Sample::SAR (Search and Rescue)::Initiate Rescue	[Edit]
Land Search and Rescue Service	NAF Sample::SAR (Search and Rescue)::Land Search and Rescue Service	[Edit]
Maritime Rescue	NAF Sample::SAR (Search and Rescue)::Maritime Rescue	[Edit]
Maritime Search	NAF Sample::SAR (Search and Rescue)::Maritime Search	[Edit]

NSOV-3 Capability to Services Mapping

Purpose

The NSOV-3 Product describes the mapping between the capabilities required by an Enterprise and services as defined for SOA.

Parameters

- **Architecture** (mandatory)
- **Capability subset** (optional)

  ☛ *The object types used for this parameter are Functional Activities and Functional Processes. As a result both functional activities and functional processes are displayed in the matrix in columns. This parameter must be set with a subset of all the functional activities and functional processes defined in the NOV-5 view.*
- **Service subset**

Contents

An NSOV-3 subview shows:

- which capabilities are required to be able to provide which services, or
- which services can be provided with a given set of capabilities.

Example of Report

NSOV-3 - Capability To Services Mapping

1. NSOV-3 - Capability To Services Mapping

This table shows the capabilities supporting the selected services.

	Assistance	Basic Maritime Search and Rescue	Basic Search and Rescue	Land SAR	Maritime SAR	Recovery	SAR (Search and Rescue)	Search
Land Search and Rescue							✓	
Maritime Rescue						✓		
Maritime Search								✓
Maritime Search and Rescue							✓	
Rescue						✓		
Search								✓
Search and Rescue							✓	

NSOV-4a Service Constraints

Purpose

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

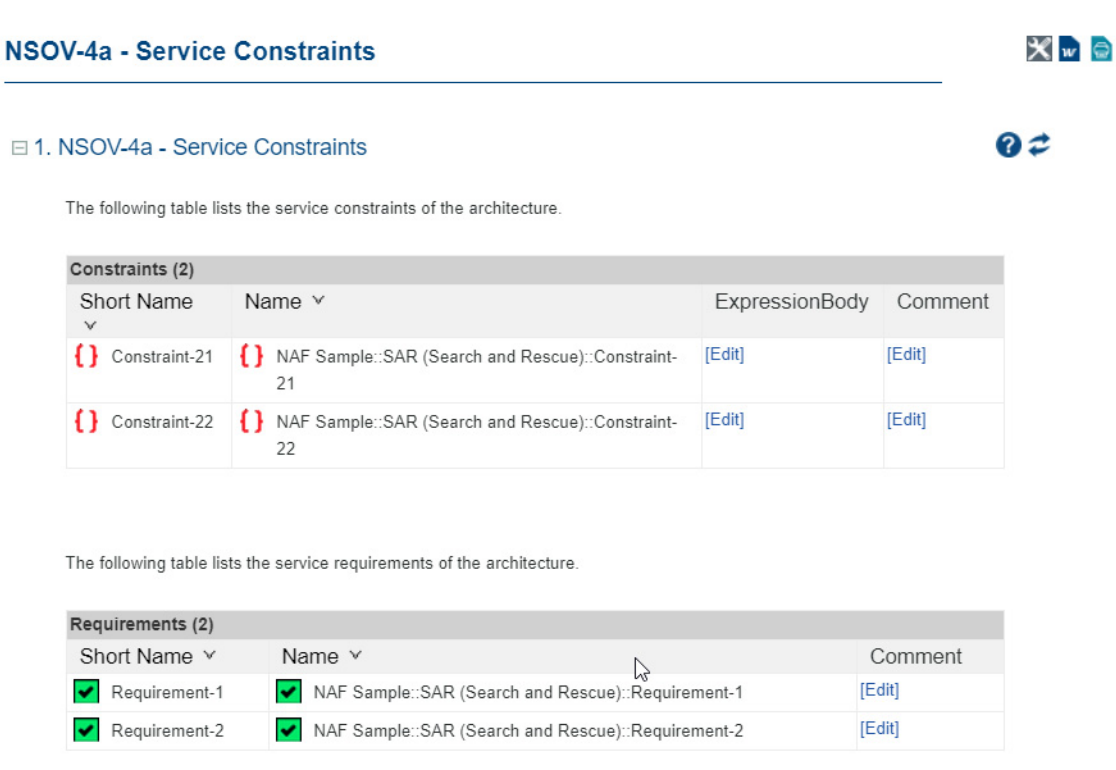
Contents

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.

Example of report



NSOV-4a - Service Constraints

1. NSOV-4a - Service Constraints

The following table lists the service constraints of the architecture.

Constraints (2)			
Short Name	Name	ExpressionBody	Comment
Constraint-21	NAF Sample::SAR (Search and Rescue)::Constraint-21	[Edit]	[Edit]
Constraint-22	NAF Sample::SAR (Search and Rescue)::Constraint-22	[Edit]	[Edit]

The following table lists the service requirements of the architecture.

Requirements (2)		
Short Name	Name	Comment
Requirement-1	NAF Sample::SAR (Search and Rescue)::Requirement-1	[Edit]
Requirement-2	NAF Sample::SAR (Search and Rescue)::Requirement-2	[Edit]

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

Purpose

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.

Parameter

- Architecture

Contents

This report details the service interaction scenarios that define use-cases of the service designed within the architecture.

A service interaction scenario is used to describe a service use case. It presents the sequence of message instances exchanged between service instances during the execution of the scenario.Example of report

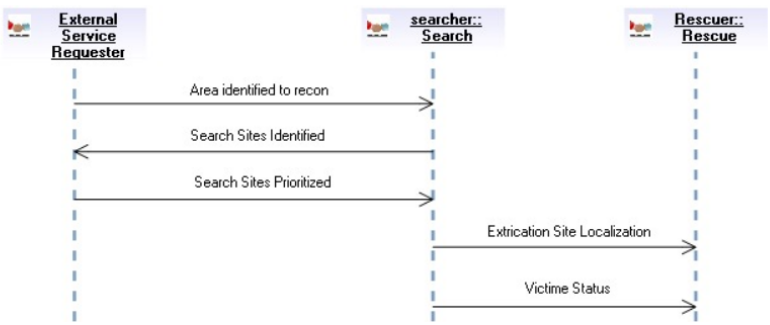
1. NSOV-4c - Service Interaction Specification

Interaction Scenarios (1)

Short Name	Name	NAF Architecture View Type	Comment
Service Interaction Scenario	NAF Sample::SAR (Search and Rescue)::Service Interaction Scenario	Service-Oriented Architecture View	[Edit]

Service Interaction Scenario

NAF Sample::SAR (Search and Rescue)::Service Interaction Scenario



NSOV-5 SERVICE FUNCTIONALITY

Parameters

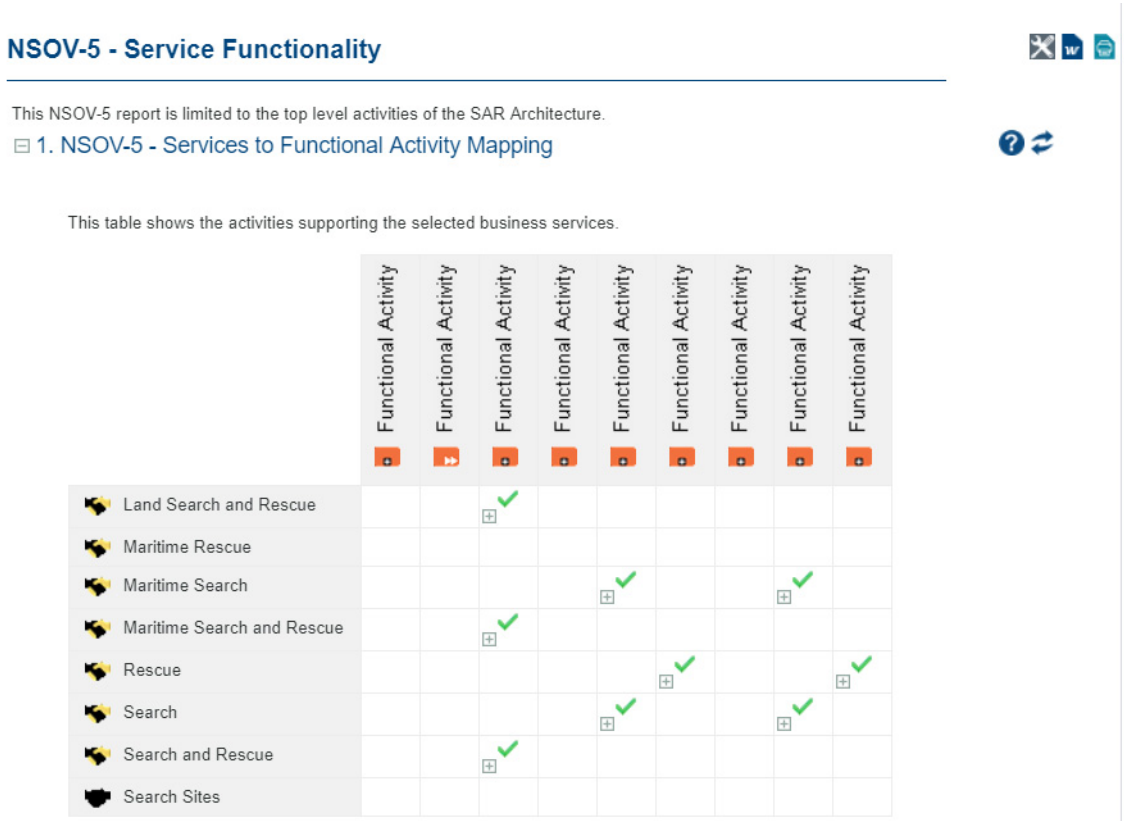
The possible parameters are as follows:

- Architecture (mandatory
- Functional activity subset
- Service subset

Services to Functional Activity Mapping chapter

The Services to Functional Activity Mapping chapter describes the mapping between services and the functional activities that those services support.

Contents



NSOV-6 SERVICE COMPOSITION

Purpose

This view specifies how services can be combined and sequenced to provide a higher level service.

Note that there might be several instances (products) of this view since there are generally several different possible service combinations.

> ☛ *It should be noted that no decision concerning implementation of the service is actually made in this view – i.e. the orchestration is a requirement specification rather than a design. In fact, none of the service subviews describe the implementation of the services - this is covered in the NSV-12 view. For more information see NSV-12 Service Provision.*

Parameters

The possible parameters are as follows:

- Architecture (mandatory
- Functional activity subset
- Service subset

Contents

The service composition report details the structure of the services.

NSOV-6 - Service Exchange Coverage Chapter

This mapping report details what operational activities deliver the information elements the services provide.

NSOV-6 - Detailed Service Exchange Coverage Chapter

This mapping report details what are the information elements the operational activities deliver that match those the services provide.

2. NSOV-6 - Detailed Service Exchange Coverage

	Functional Activity	Functional Activity	Functional Activity	Functional Activity	Functional Activity
Rescue				Stabilized Victims	Stabilized Victims
Search	Extrication sites identified and prioritized	Extrication sites identified and prioritized Victim not found	Victim not found		
Search and Rescue	Extrication sites identified and prioritized	Extrication sites identified and prioritized Victim not found	Victim not found	Stabilized Victims	Stabilized Victims
Search Sites	Extrication sites identified and prioritized	Extrication sites identified and prioritized Victim not found	Victim not found		

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-2a System Port Specification
- ✓ NSV-2b System to System Port Connectivity
- ✓ NSV-2c System Connectivity Clusters
- ✓ NSV-2d Systems Communication Quality Requirements
- ✓ NSV-3 Resources-Resources Matrix
- ✓ NSV-4 Systems Functionality Description
- ✓ NSV-5 Systems Function to Operational Activity Traceability 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-12 Service Provision

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).

See:

- ✓ Creating Capability Configurations
- ✓ The NSV-1 Report Template
- ✓ The NSV-1 Application Composition Hierarchy Chapter
- ✓ The NSV-1 Application Specialization Hierarchy Chapter
- ✓ The NSV-1 Application Generalization Hierarchy Chapter
- ✓ The NSV-1 Application Dictionary Chapter
- ✓ The NSV-1 Application Exchange Balance Chapter
- ✓ The NSV-1 Artifact Composition Hierarchy Chapter
- ✓ The NSV-1 Artifact Specialization Hierarchy Chapter
- ✓ The NSV-1 Artifact Generalization Hierarchy Chapter
- ✓ The NSV-1 Artifact Dictionary Chapter
- ✓ The NSV-1 Artifact Exchange Balance Chapter
- ✓ The NSV-1 Resource Architecture Composition Hierarchy Chapter
- ✓ The NSV-1 Resource Architecture Specialization Hierarchy Chapter
- ✓ The NSV-1 Resource Architecture Generalization Hierarchy Chapter
- ✓ The NSV-1 Resource Architecture Dictionary Chapter
- ✓ The NSV-1 Resource Architecture Exchange Balance Chapter
- ✓ The NSV-1 System Exchange Chapter
- ✓ The NSV-1 System Exchange Compliance Chapter

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.*

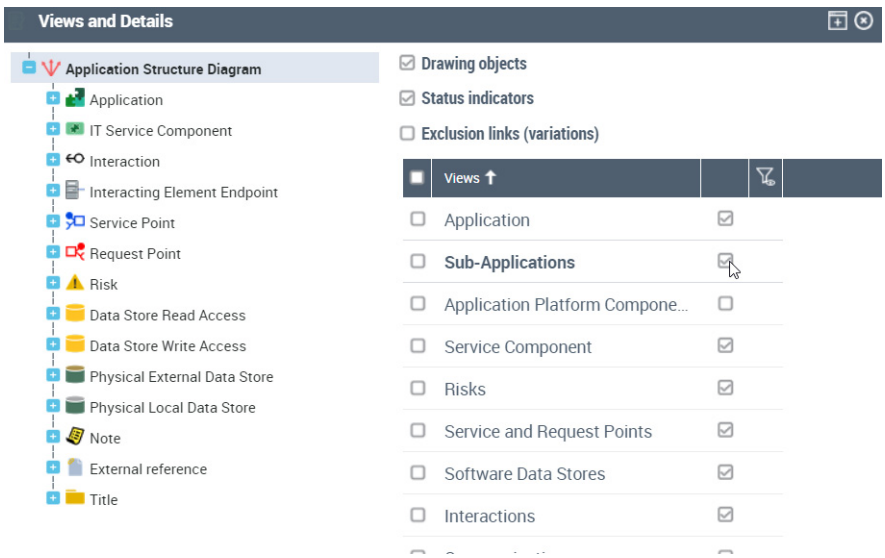
To create an application structure diagram:

1. In the NAF navigation tree, expand **System Views > NSV-1 System Interface Description > Applications**.
2. Right-click an application and select **New > 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.



The NSV-1 Report Template

The NSV-1 report template synthesizes information about the composition and interaction of systems. This template is composed of four parameters:

- The **Architecture** to be analyzed.
- The **Application Subset**. This parameter is optional and is used to reduce the scope of the study to a subset of the different applications used in the architecture.
- The **Artifact Subset**. This is also an optional parameter. This is used to reduce the scope of the study to a selection of artifacts. If no artifact is selected, all the artifacts of the architecture are included in the analysis.
- The **Resource Architecture Subset**, which is also optional. This parameter is set with a subset of the resource architectures defined in the architecture. If this parameter is not set, all the resource architectures of the architecture are taken into account in the analysis.

The NSV-1 Application Composition Hierarchy Chapter

This chapter gives a view of the application hierarchies. Application Tree Diagrams are displayed with the complete structure, taking into account the optional scope.

The NSV-1 Application Specialization Hierarchy Chapter

The NSV-1 application specialization hierarchy chapter shows how the applications of the architecture are specialized.

The applications are presented in a table along with their comments and the library to which they belong. If a variant of an application is created, the varied application is presented in the table along with its variant.

To create new specializations (variants), use the **New > Variant** command on the application to be varied (specialized).

The NSV-1 Application Generalization Hierarchy Chapter

The NSV-1 application generalization hierarchy chapter shows how the applications of the architecture are generalized.

The applications are presented in table form along with their comments and the library to which they belong. If a variant of an application exists, this variant is also presented in the table along with the varied application (application from which the variant was created).

To add a new generalization, use the **Connect > Variant Of** command of the specialized item (variant).

The NSV-1 Application Dictionary Chapter

The Application Dictionary chapter lists the applications defined in the architecture in an alphabetical table. This enables the retrieval of items for which the names are

known. It also displays the Application Structure diagram for the different applications.

The NSV-1 Application Exchange Balance Chapter

This chapter is used to verify that the exchanges defined between applications are correctly designed. All the incoming and outgoing exchanges of the applications are compared and examined to reveal missing as well as unnecessary information for the exchange design. The chapter lists all the internal and outside messages and interactions that detail the selected applications.

The NSV-1 Artifact Composition Hierarchy Chapter

This chapter gives a view of the artifact hierarchies. Artifact Tree Diagrams are displayed with the complete structure, taking into account the optional scope.

The NSV-1 Artifact Specialization Hierarchy Chapter

The NSV-1 artifact specialization hierarchy chapter shows how the artifacts of the architecture are specialized.

The artifacts are presented in a table along with their comments and the library to which they belong. If a variant of an artifact is created, the varied artifact is presented in the table along with its variant.

To create new specializations (variants), use the **New > Variant** command on the system process to be varied (specialized).

The NSV-1 Artifact Generalization Hierarchy Chapter

The NSV-1 artifact generalization hierarchy chapter shows how the artifacts of the architecture are generalized.

The artifacts are presented in table form along with their comments and the library to which they belong. If a variant of an artifact exists, this variant is also presented in the table along with the varied artifact (artifact from which the variant was created).

To add a new generalization, use the **Connect > Variant Of** command of the specialized item (variant).

The NSV-1 Artifact Dictionary Chapter

The Artifact Dictionary chapter lists the artifacts defined in the architecture in an alphabetical table. This enables the retrieval of items for which the names are known. It also displays the Artifact Structure diagram for the different artifacts.

The NSV-1 Artifact Exchange Balance Chapter

This chapter is used to verify that the exchanges defined between artifacts are correctly designed. All the incoming and outgoing exchanges of the artifacts are compared and examined to reveal missing as well as unnecessary information for

the exchange design. The chapter lists all the internal and outside messages and interactions that detail the selected artifacts.

The NSV-1 Resource Architecture Composition Hierarchy Chapter

This chapter gives a view of the resource architecture hierarchies. Resource Architecture Tree Diagrams are displayed with the complete structure, taking into account the optional scope.

The NSV-1 Resource Architecture Specialization Hierarchy Chapter

The NSV-1 resource architecture specialization hierarchy chapter shows how the resource architectures of the architecture are specialized.

The resource architectures are presented in a table along with their comments and the library to which they belong. If a variant of a resource architecture is created, the varied resource architecture is presented in the table along with its variant.

To create new specializations (variants), use the **New > Variant** command on the system process to be varied (specialized).

The NSV-1 Resource Architecture Generalization Hierarchy Chapter

The NSV-1 resource architecture generalization hierarchy chapter shows how the resource architectures of the architecture are generalized.

The resource architectures are presented in table form along with their comments and the library to which they belong. If a variant of a resource architecture exists, this variant is also presented in the table along with the varied resource architecture (resource architecture from which the variant was created).

To add a new generalization, use the **Connect > Variant Of** command of the specialized item (variant).

The NSV-1 Resource Architecture Dictionary Chapter

The Resource Architecture Dictionary chapter lists the resource architectures defined in the architecture in an alphabetical table. This enables the retrieval of items for which the names are known. It also displays the Resource Architecture Structure diagram for the different resource architectures.

The NSV-1 Resource Architecture Exchange Balance Chapter

This chapter is used to verify that the exchanges defined between resource architectures are correctly designed. All the incoming and outgoing exchanges of the resource architectures are compared and examined to reveal missing as well as unnecessary information for the exchange design. The chapter lists all the internal and outside messages and interactions that detail the selected resource architectures.

The NSV-1 System Exchange Chapter

This chapter shows the diagrams that display information exchanges between systems. It also details the information exchange between the selected systems. All

interactions connecting systems are displayed in a list with their name, the source, and target of the exchange and the information elements.

The NSV-1 System Exchange Compliance Chapter

This chapter indicates the level of compliance between the need to exchange information from one system to another and the information actually exchanged through the implemented missions.

Tasks supported by the systems actually exchange information within the context of designed missions. This information is compared with the need defined in the needlines/interactions. Missing or unexpected information is detected.

This chapter relies on the systems defined in the NSV-1 subview as well as the system processes that describe how missions are performed. These processes are described in detail in NSV-4, however, a brief description is necessary to aid in understanding the content of this chapter.

A system structure does not indicate how missions are performed. It only describes the systems involved in the mission and how they are structured. Through interactions, you can guess the potential information exchanges that occur between systems. To add a dynamic perspective to systems, a process must be described (see the NSV-4 subview). This process is composed of tasks that exchange information. Information is actually exchanged in the process while the system structure only describes the ability to perform such exchanges.

Tasks are performed under the control of resource architectures, applications and artifacts.

The System exchange compliance chapter identifies three states for the information exchange:

- **Information Exchange Correctly Designed**: an information item is defined at both the system and the process levels.
- **Missing Information**: an information item can potentially be exchanged between two systems, however, there is no process available to perform this exchange. The question is therefore, whether the interface described between the two systems is still useful.
- **Unexpected Information**: an information item is exchanged in a process between two activities (tasks), however, there is no interaction between the systems carrying out the activities (task). The question is therefore, should the information be exchanged in this manner in the process and must the interaction be reviewed to depict the ability to make such an exchange.

NSV-2a System Port Specification

The purpose of the NSV-2a subview is to provide specifications for how clients of a particular environment can connect to the systems in that environment. This subview therefore specifies the ports on a system, and the protocols used by these ports to communicate with other systems.

The NSV-2a Report Template

The NSV-2a report template synthesizes information about the connection ports used in systems of an architecture.

The parameters taken into account for the report are:

- The **Architecture** to be analyzed.
- The **Application Subset**. This parameter is optional and is used to reduce the scope of the study to a subset of the different applications used in the architecture.
- The **Artifact Subset**. This is also an optional parameter. This is used to reduce the scope of the study to a selection of artifacts. If no artifact is selected, all the artifacts of the architecture are included in the analysis.
- The **Resource Architecture Subset**, which is also optional. This parameter is set with a subset of the resource architectures defined in the architecture. If this parameter is not set, all the resource architectures of the architecture are taken into account in the analysis.

The NSV-2a System Ports Dictionary Chapter

This chapter lists the communication ports of the architecture in alphabetical order. This information is presented in a table. A paragraph dedicated to each of the retrieved communication ports is displayed in the chapter. This paragraph includes:

- The physical resource to which the communication port belongs with its name and comment
- The communication channel to which the port is attached with its name and comment

The NSV-2a System Ports by Application Dictionary Chapter

This chapter lists all the applications to which the communication ports retrieved during the study are linked.

A paragraph dedicated to each application is displayed with the name and comment of the communication ports and communication protocols that belong to it. The Application Structure Diagram for each application is also displayed.

The NSV-2a System Ports by Artifact Dictionary Chapter

This chapter lists all the artifacts of the architecture (or those defined in the analysis scope) linked to communication ports.

A paragraph dedicated to each artifact is displayed with the following:

- The Artifact Assembly Diagram for the artifact
- The "owned" communication ports of the artifact with their name and comment
- The communication protocols that the different communication ports use, with their name and comment

The NSV-2a System Ports by Resource Architecture Dictionary Chapter

This chapter lists all the resource architectures of the architecture (or those defined in the analysis scope) linked to communication ports.

A paragraph dedicated to each resource architecture is displayed with the following:

- The Resource Architecture Structure Diagram for the resource architecture concerned
- The "owned" communication ports of the resource architecture with their name and comment

The communication protocols that the different communication ports use, with their name and comment

NSV-2b System to System Port Connectivity

The NSV-2b subview identifies the protocols stack used by a connection between two ports. These ports may be on different systems. The NSV-2b subview is closely related to the NSV-2a subview which specifies the available protocols on each port. The connections specified in the NSV-2b subview conform to the protocols specified on the corresponding port definitions in the NSV-2a subview.

The NSV-2b Report Template

The NSV-2b report template synthesizes information about the communication channels considered in the connectivity of systems of the architecture.

The parameters taken into account for the report are:

- The **Architecture** to be analyzed.
- The **Application Subset**. This parameter is optional and is used to reduce the scope of the study to a subset of the different applications used in the architecture.
- The **Artifact Subset**. This is also an optional parameter. This is used to reduce the scope of the study to a selection of artifacts. If no artifact is selected, all the artifacts of the architecture are included in the analysis.
- The **Resource Architecture Subset**, which is also optional. This parameter is set with a subset of the resource architectures defined in the architecture. If this parameter is not set, all the resource architectures of the architecture are taken into account in the analysis.

The NSV-2b Communication Channels Dictionary Chapter

This chapter provides a list of the communication channels of the architecture with their "End points" (artifact component, communication channel, etc). This

information is presented in a table. A paragraph dedicated to each of the retrieved communication channels is displayed in the chapter. This paragraph includes:

- The physical resource to which the communication channel belongs, with its name and comment
- The communication ports that use the communication channel, with their name and comment
- The communicating elements, e.g. artifact, linked to the communication channel
- The communication protocols used by the communication ports, with their name and comment

The NSV-2b Communication Channels by Application Dictionary Chapter

This chapter lists all the applications to which the communication channels retrieved during the study are linked.

A paragraph dedicated to each application is displayed with the name and comment of the communication channel and communication protocols that belong to it. The Application Structure Diagram for each application is also displayed.

The NSV-2b Communication Channels by Artifact Dictionary Chapter

This chapter lists all the artifacts of the architecture (or those defined in the analysis scope) linked to communication channels.

A paragraph dedicated to each artifact is displayed with the following:

- The Artifact Assembly Diagram for the artifact
- The "owned" communication channels of the artifact with their End points
- The communication protocols used, with their name and comment

The NSV-2b Communication Channels by Resource Architecture Dictionary Chapter

This chapter lists all the resource architectures of the architecture (or those defined in the analysis scope) linked to communication channels.

A paragraph dedicated to each resource architecture is displayed with the following:

- The Resource Architecture Structure Diagram for the resource architecture
- The "owned" communication channels of the resource architecture with their End points (e.g. Architecture use, physical asset, etc.)
- The communication protocols used, with their name and comment

NSV-2c System Connectivity Clusters

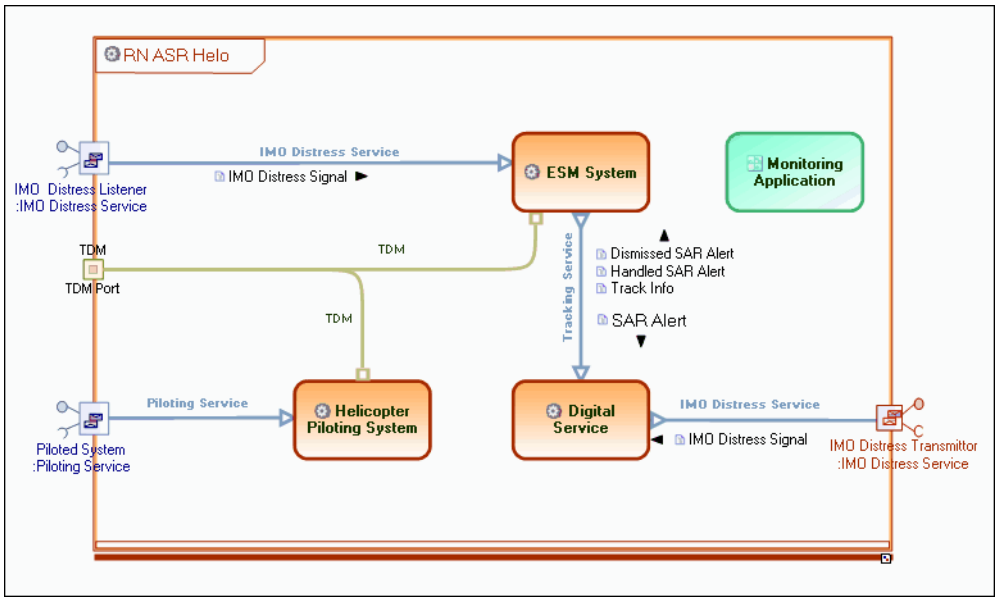
The purpose of the System Connectivity Clusters subview is to define the connectivity requirements between nodes. This subview is then used to estimate requirements for physical routing and bandwidth.

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.

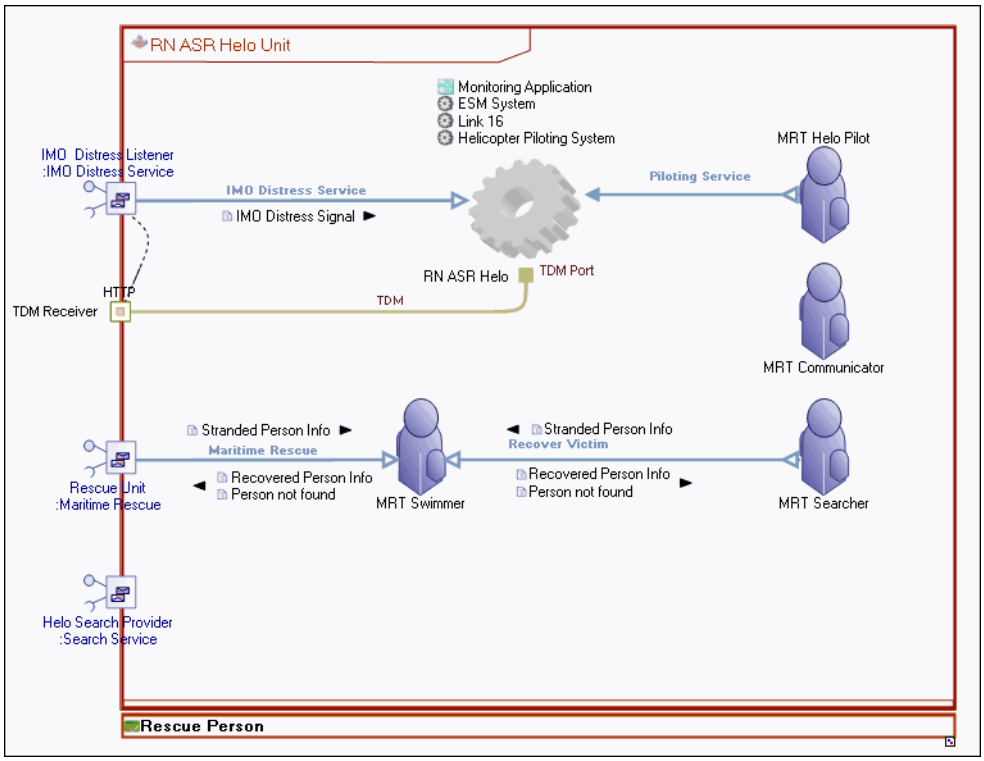


At the upper level, the RN ASR Helo system is used in the RN ASR Helo Unit system architecture.

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.



The NSV-2c Report Template

The NSV-2c report template synthesizes information about the communication channels considered as connectivity clusters.

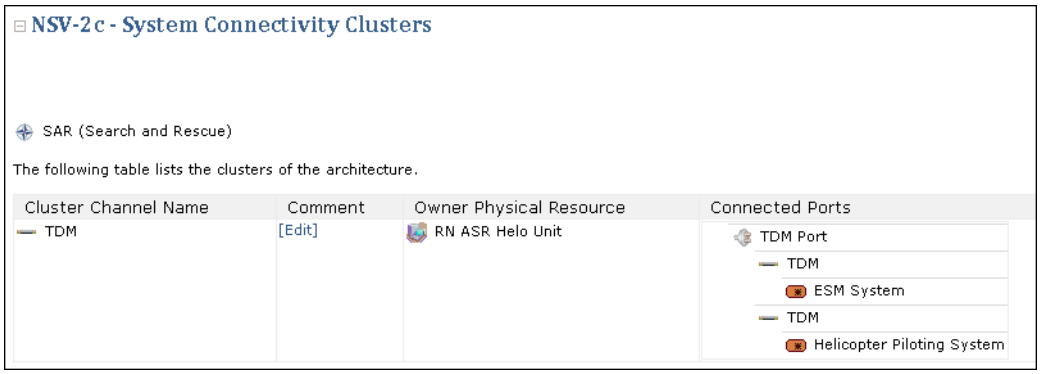
The parameters taken into account for the report are:

- The Architecture to be analyzed.
- The Subset parameter. This parameter is optional and can be set with a subsets of the applications, artifacts, communication channels and resource architectures used in the architecture. If this parameter is not set, all these aforementioned architecture objects are taken into account in the analysis.

The NSV-2c System Connectivity Clusters Chapter

The system Connectivity Clusters chapter displays a list of all the communication channels that are considered as connectivity clusters in the architecture.

For each cluster, the name, the comment, the system owning the cluster and the connected sub-systems are listed.



Example of an NSV-2c System Connectivity Clusters Chapter

NSV-2d Systems Communication Quality Requirements

The NSV-2d Systems Communication Quality Requirements subview identifies the specific quality requirements that apply to communications between systems.

It focuses on specific categories of quality requirements for systems communication. This focus is available to offer separate attention to certain communication aspects, other than those already specified in the other NSV-2 subviews.

Currently, the only category supported by this subview is the "Electromagnetic Spectrum and Bandwidth" category.

In addition to NSV-7 that specifies the quality requirements for systems as a whole, NSV-2d specifies the quality requirements for specific system communication aspects.

The NSV-2d Report Template

The NSV-2d report template synthesizes information about the quality requirements for the communication between systems.

The parameters taken into account for the report are:

- The **Architecture** to be analyzed.
- The **Subset** parameter. This parameter is optional and can be set with a subsets of the applications, artifacts, communication channels and resource architectures used in the architecture. If this parameter is not set, all these aforementioned architecture objects are taken into account in the analysis.

The NSV-2d Systems Communication Quality Requirements Chapter

This chapter presents a list of the communication channels that support the communication protocols that define the "spectrum allocation" standard.

NSV-3 RESOURCES-RESOURCES MATRIX

NSV-3 provides details about the interface characteristics described in NSV-1 for the architecture, arranged in matrix form.

NSV-3 gives a quick overview of all the interface characteristics presented in multiple NSV-1 diagrams. The matrix form facilitates making rapid assessments of the potential commonalities and redundancies (or, if fault-tolerance is desired, the lack of redundancies).

NSV-3 is a useful tool for managing the evolution of systems and system infrastructures, the insertion of new technologies/functionalities, and the redistribution of systems and processes with evolving operational requirements.

Users can easily access NSV-3 reports in the NAF navigation tree. No other objects are displayed since the aim is to synthesize information already designed in the NSV-1 section.

The NSV-3 Report Template

The NSV-3 Report template synthesizes information about the system interfaces in a matrix.

The first parameter is used to indicate the architecture to be analyzed.

The second parameter, the System Subset is optional. This is used to reduce the scope of the study to a subset of the systems defined in the architecture. If this parameter is not set, all the systems defined in the architecture are taken into account. The subset can be defined by setting systems.

The NSV-3 Resources to Resources Channel Matrix Chapter

This chapter displays a squared matrix of systems which indicates if systems are connected to other systems through communication channels. The systems are organized in rows and columns.

Each cell in the matrix represents the communication channel between a system in row and a system in a column. If a communication channel exists between two systems, the corresponding cell contains a black dot.

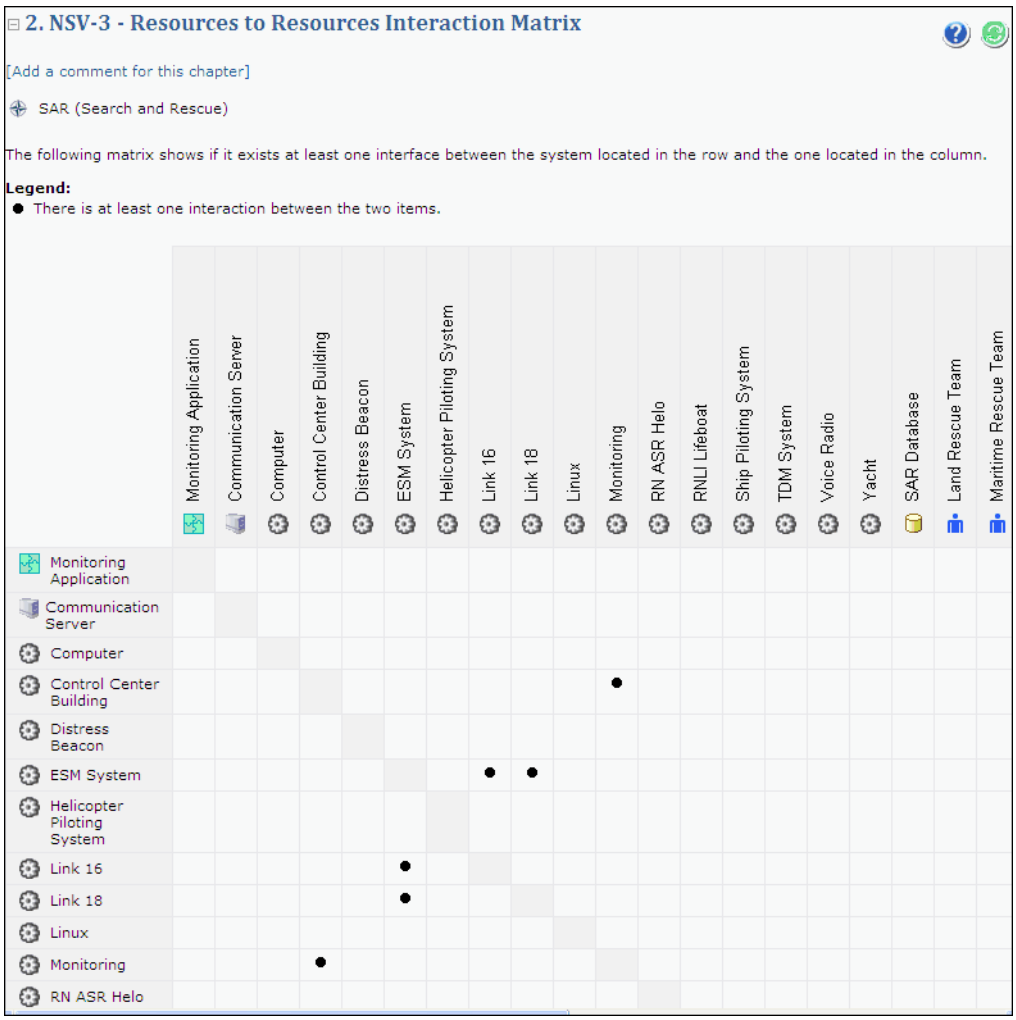
Diagonal cells dealing with the same source and target are grayed.

The NSV-3 Resources to Resources Interaction Matrix Chapter

This chapter displays the system components that should be able to interact with each other. The system interaction information is displayed in a matrix. The selected systems are organized in rows and columns.

Each cell in the matrix represents the interface between a system in row and a system in a column. If an interface exists, the corresponding cell contains a black dot.

Diagonal cells dealing with the same source and target are grayed.



Example of an NSV-3 Resources to Resources Interaction Matrix

NSV-4 Systems Functionality Description

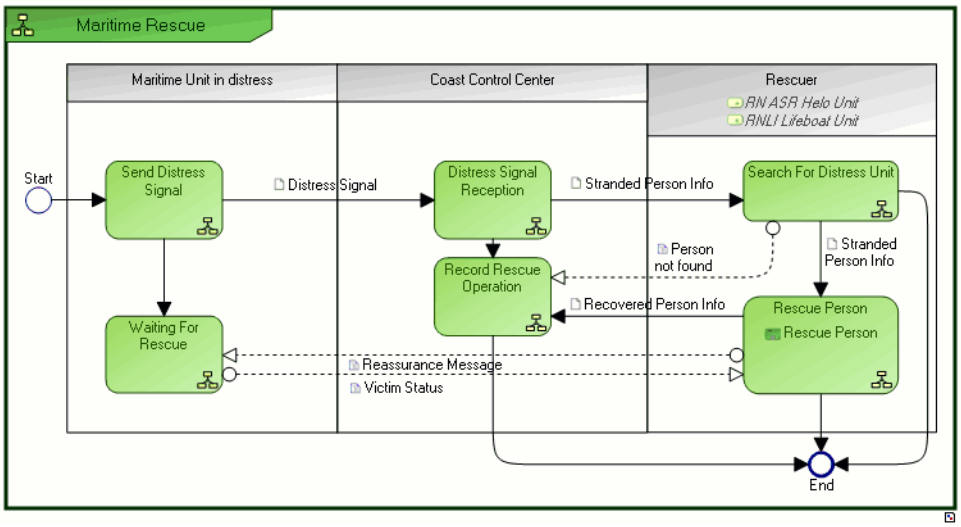
The NSV-4 subview documents system functional hierarchies and system functions, and the system data flows between them. While NSV-1 products describe the available interface between systems, they do not explain how these interfaces are used to exchange information or how data flows between the systems. 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.

About System Processes and System Functions

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.*

See:

- ✓ The NSV-4 Report Template
- ✓ The NSV-4 System Function Composition Hierarchy Chapter
- ✓ The NSV-4 System Function Dictionary Chapter
- ✓ The NSV-4 System Function Exchange Chapter
- ✓ The NSV-4 System Function Exchange Balance Chapter

The NSV-4 Report Template

The NSV-4 report template synthesizes information about the system functions and the data flowing between them.

The first parameter indicates the architecture to be analyzed.

The second parameter, called a System Process Subset, is optional. This is used to reduce the scope of the study to a subset of the system processes defined in the architecture. If this parameter is not set, all the system processes defined in the architecture are taken into account.

The NSV-4 System Function Composition Hierarchy Chapter

This chapter expands the hierarchy of system functions. It details the hierarchical structure of the system processes and their assigned tasks. The performance of system functions can be delegated to other system processes. This results in a system functions tree where system functions are indirectly connected by sub-system processes.

The chapter starts from (explicitly or implicitly) selected system processes. For each system process a tree structure of the system function is displayed with its name and comment. The tasks and performing system processes are also included in the tree.

The NSV-4 System Function Specialization Hierarchy Chapter

The NSV-4 system function specialization hierarchy chapter shows how the system processes of the architecture are specialized.

The system processes are presented in a table along with their comments and the library to which they belong. If a variant of a system process is created, the varied system process is presented in the table along with its variant.

To create new specializations (variants), use the **New > Variant** command on the system process to be varied (specialized).

The NSV-4 System Function Generalization Hierarchy Chapter

The NSV-4 system function generalization hierarchy chapter shows how the system processes of the architecture are generalized.

The system processes are presented in table form along with their comments and the library to which they belong. If a variant of a system process exists, this variant is also presented in the table along with the varied system process (system process from which the variant was created).

To add a new generalization, use the **Connect > Variant Of** command of the specialized item (variant).

The NSV-4 System Function Dictionary Chapter

This chapter lists the flows of data exchanged by the system function.

A list of the system processes retrieved during the analysis is presented.

If all the system processes of the architecture are retrieved (no value for the sub-system process parameter), the system process diagrams of the root system process are displayed in the chapter.

A paragraph dedicated to each of the selected or retrieved system processes is displayed in the chapter. This paragraph includes:

- The system process with its name and comment
- The system process diagrams which describe the system process
- The list of tasks organized by the system process with the name and comment of each task
- The list of all the tasks retrieved in the architecture

The NSV-4 System Function Exchange Chapter

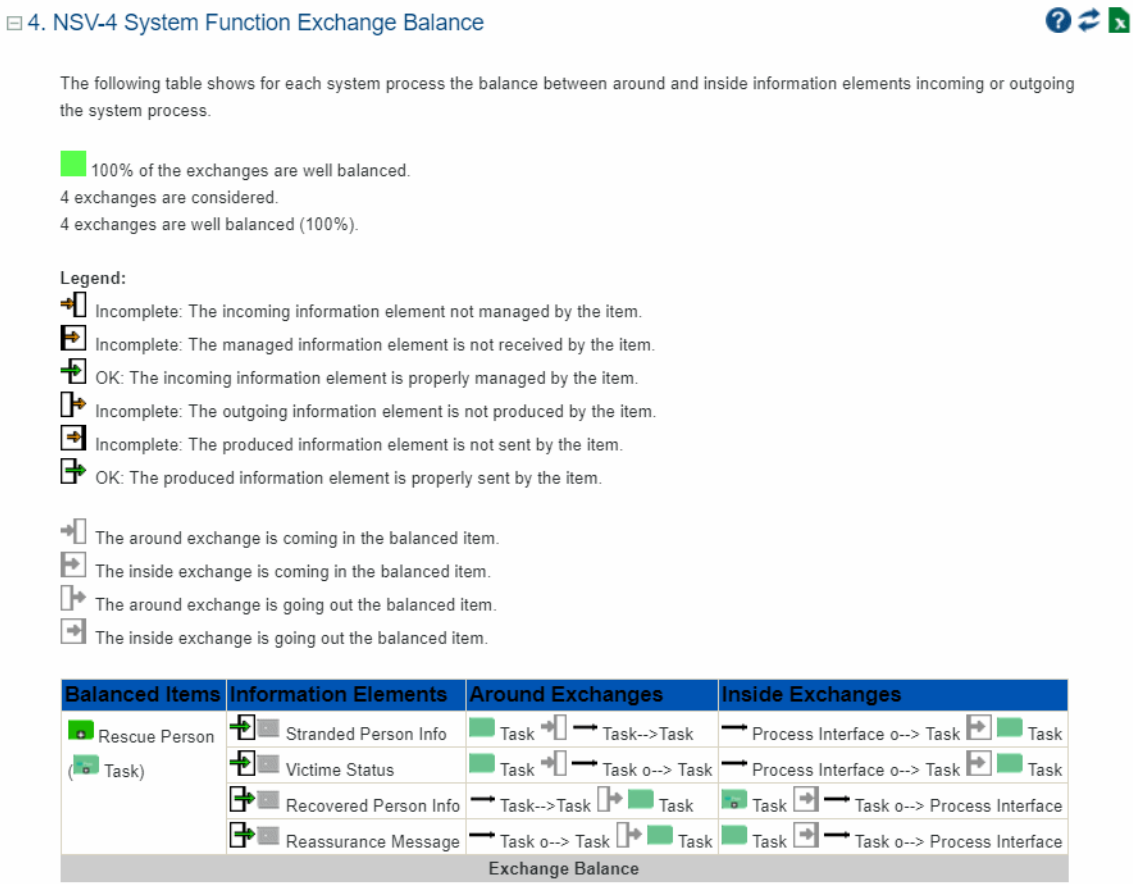
The NSV-4 System Function Exchange chapter lists the interactions that are set between the tasks of system processes. It shows the diagrams that display information exchanges between system processes and system functions. It also details the information exchange between the selected tasks. All messages or interactions connecting tasks are displayed in a list with their name, comment and source and target and the information element that is exchanged.

The NSV-4 System Function Exchange Balance Chapter

This chapter helps to define the compliance of the exchanges defined within system processes and the exchanges received and /or sent by these system processes. The chapter lists all the exchanges modeled within and outside each system process between the contained and detailing tasks.

All incoming and outgoing exchanges defined inside and outside of the system process are compared. This comparison enables the user to determine missing as well as unnecessary information for the exchange design.

Different icons with arrows are used to give valid information relating to each message and interaction. These arrows aid in the comparison making process.



4. NSV-4 System Function Exchange Balance

The following table shows for each system process the balance between around and inside information elements incoming or outgoing the system process.

100% of the exchanges are well balanced.
4 exchanges are considered.
4 exchanges are well balanced (100%).

Legend:

Incomplete: The incoming information element not managed by the item.
Incomplete: The managed information element is not received by the item.
OK: The incoming information element is properly managed by the item.
Incomplete: The outgoing information element is not produced by the item.
Incomplete: The produced information element is not sent by the item.
OK: The produced information element is properly sent by the item.

The around exchange is coming in the balanced item.
The inside exchange is coming in the balanced item.
The around exchange is going out the balanced item.
The inside exchange is going out the balanced item.

Balanced Items	Information Elements	Around Exchanges	Inside Exchanges
Rescue Person (Task)	Stranded Person Info	Task Task-->Task	Process Interface o--> Task Task
	Victime Status	Task Task o--> Task	Process Interface o--> Task Task
	Recovered Person Info	Task-->Task Task	Task Task o--> Process Interface
	Reassurance Message	Task o--> Task Task	Task Task o--> Process Interface

Exchange Balance

Example of an NSV-4 System Function Exchange Balance Chapter

In the above example, two of the exchanges were not well balanced. Upon investigation it was revealed that they were not accounted for in any of the system capability configurations defined in NCV-3.

Retrieving Interesting Information in the NSV-4 Report Chapters

To ensure that interesting information appears in the report chapters, and especially the NSV-4 System Function Exchange and System Function Exchange Balance chapters, the modeler should ensure that the system processes to be studied and / or included in the report at least have the following data:

- There should be tasks that "call" other system processes
- There should be messages with content between at least two tasks of a system process

It can be argued that the most interesting section of the NSV-4 System Function Exchange chapter is the table that lists the different exchanges that are sent from or to system processes and tasks with the information element. If there are no messages with content between tasks, there will be no exchange to show in the chapter.

If there are no called system processes for tasks, and if the called system processes have not been described or modeled, in other words are empty, the exchanges concerning these system processes are excluded from the NSV-4 System Function Exchange Balance chapter. Likewise if there are no incoming or outgoing exchanges to study there will be nothing to display in this chapter.

NSV-5 Systems Function to Operational Activity Traceability Matrix

The NSV-5 Systems Function to Operational Activity Traceability 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.

The NSV-5 Report Template

The NSV-5 report template generates a matrix of the mapping between a group of system functions and activities.

- The first parameter indicates the Architecture to be analyzed.
- The second parameter, the Operational Activity Subset, is optional. It restricts the study to a subset of the operational activities. If this parameter is not set, all the activities defined in the architecture are taken into account. This can result in a large matrix. To focus on a specific part of the architecture, the parameter value is defined using either activities or business capabilities (business processes).
- The third parameter, the System Task Subset, which is also optional, is used to reduce the scope of the study to a subset of the system functions defined in the architecture. If this parameter is not set, all the system functions defined in the architecture are taken into account. The parameter value is defined by setting system capabilities as shortcuts to all the contained system functions.

The NSV-5 Systems Function to Operational Activity Traceability Matrix Chapter

This chapter displays a matrix containing:

- the selected system processes in rows

  A system process is the executable representation of a process. It consists of tasks.

- the functional processes and activities in columns.

  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.

Mappings between system functions and functional activities can be indicated in each cell of the matrix. A checkmark is displayed if a mapping exists.

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.

The NSV-6 Report Template

The report template used to generate the NSV-6 report chapters is composed of three parameters.

- The first parameter is the Architecture to be analyzed.
- The second parameter is an optional value that is used to reduce the scope of the study. This parameter is set with interfaces described in the architecture. In this case, the NSV 6 report focuses on this set of interfaces and on the information exchanges contained in these interfaces. The **HOPEX** objects used to set this parameter are interactions.
- The third parameter, which is also optional is set with the exchange properties to be considered during the analysis. The objects used for these properties are of the **HOPEX** MetaAttribute and MetaAssociationEnd types, which are applicable to content and message flows. During the report generation, the value for the parameter is verified and for each correct value encountered, a column is added to the table that displays the value of the attribute for the content or the message flow, or the list of the objects linked to the association.

The NSV-6 Systems Data Exchange Dictionary Chapter

The Systems Data Exchange Dictionary is similar to the OV-3 Operational Information Exchange Dictionary, however, columns characterize the system properties of the exchange.

This Systems Data Exchange Dictionary chapter lists in alphabetical order all the information elements (contents) of the architecture, defined at the system level.

Each information element is then described in detail with a list of all the information exchanges in which it is involved. The source and target of the information is also included.

NSV-6 Systems Data Exchange Dictionary

SAR (Search and Rescue)

The following table lists all the system information elements retrieved in the architecture.

Information Elements (13)

Short Name	Name	Comment
Control	SAR (Search and Rescue)::Control	[Edit]
Distress Signal	SAR (Search and Rescue)::Distress Signal	[Edit]
Missing Person Info	SAR (Search and Rescue)::Missing Person Info	[Edit]
Reassurance Message	SAR (Search and Rescue)::Reassurance Message	[Edit]
Search Cancel Request	SAR (Search and Rescue)::Search Cancel Request	[Edit]
Search Case ID	SAR (Search and Rescue)::Search Case ID	[Edit]
Search Request	SAR (Search and Rescue)::Search Request	[Edit]
Stranded Person Info	SAR (Search and Rescue)::Stranded Person Info	[Edit]
Track Info	SAR (Search and Rescue)::Track Info	[Edit]
Victime Status	SAR (Search and Rescue)::Victime Status	[Edit]
Warning Order	SAR (Search and Rescue)::Warning Order	[Edit]

Control

SAR (Search and Rescue)::Control

The following table lists the exchanges the current information element is involved in.

Exchanges (5)

Short Name	Name	Sources	Targets	Comment
Controlling Service	SAR (Search and Rescue)::SAR Operational Architecture (Search and Rescue)::Controlling Service	Tactical C2 Rescue Node	Tactical C2 Rescue Node	[Edit]
Controlling Service	SAR (Search and Rescue)::SAR Operational Architecture (Search and Rescue)::Controlling Service	Tactical C2 Search Node	Tactical C2 Search Node	[Edit]
Controlling Service	SAR (Search and Rescue)::Coast Control Center::Controlling Service	Control Center Building	Control Center Building	[Edit]

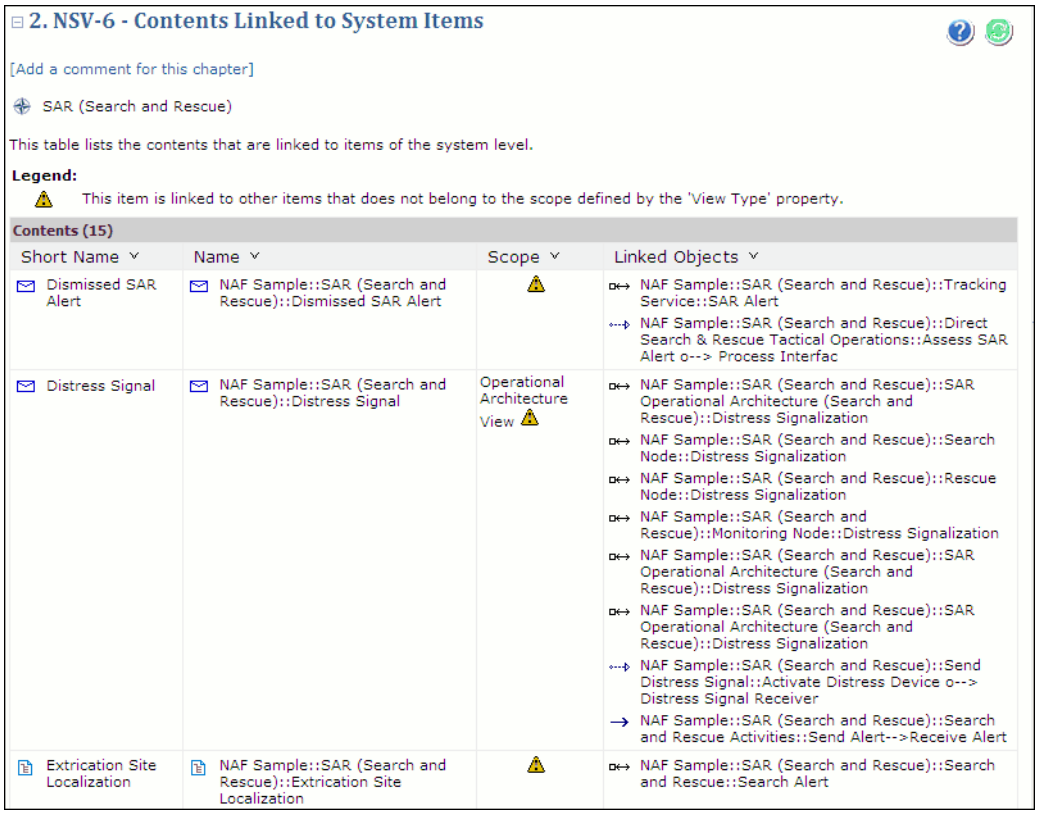
Example of an NSV-6 Systems Data Exchange Dictionary

The NSV-6 Systems Data Exchange Matrix Chapter

This chapter identifies and displays the information elements of the architecture systems and the relevant attributes of their information exchanges. These exchanges are associated to their producer and consumer systems and tasks and to the interactions that they satisfy.

The NSV-6 Contents Linked to System Items Chapter

This chapter presents a table which lists the contents of the architecture that are linked to system items. This helps the user determine if the type of contents linked are defined according to the scope defined for this system architecture. This definition is based on the "View Type" property.



2. NSV-6 - Contents Linked to System Items

[Add a comment for this chapter]

SAR (Search and Rescue)

This table lists the contents that are linked to items of the system level.

Legend:

⚠ This item is linked to other items that does not belong to the scope defined by the 'View Type' property.

Contents (15)

Short Name	Name	Scope	Linked Objects
Dismissed SAR Alert	NAF Sample::SAR (Search and Rescue)::Dismissed SAR Alert	⚠	NAF Sample::SAR (Search and Rescue)::Tracking Service::SAR Alert NAF Sample::SAR (Search and Rescue)::Direct Search & Rescue Tactical Operations::Assess SAR Alert o--> Process Interfac
Distress Signal	NAF Sample::SAR (Search and Rescue)::Distress Signal	Operational Architecture View ⚠	NAF Sample::SAR (Search and Rescue)::SAR Operational Architecture (Search and Rescue)::Distress Signalization NAF Sample::SAR (Search and Rescue)::Search Node::Distress Signalization NAF Sample::SAR (Search and Rescue)::Rescue Node::Distress Signalization NAF Sample::SAR (Search and Rescue)::Monitoring Node::Distress Signalization NAF Sample::SAR (Search and Rescue)::SAR Operational Architecture (Search and Rescue)::Distress Signalization NAF Sample::SAR (Search and Rescue)::SAR Operational Architecture (Search and Rescue)::Distress Signalization NAF Sample::SAR (Search and Rescue)::Send Distress Signal::Activate Distress Device o--> Distress Signal Receiver NAF Sample::SAR (Search and Rescue)::Search and Rescue Activities::Send Alert-->Receive Alert
Extrication Site Localization	NAF Sample::SAR (Search and Rescue)::Extrication Site Localization	⚠	NAF Sample::SAR (Search and Rescue)::Search and Rescue::Search Alert

Example of a Contents Linked to System Items Chapter

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).

About Quality Requirements

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".

The NSV-7 Report Template

The SV-7 report template is supplied with one parameter: the architecture to be analyzed. There are no additional parameters.

The NSV-7 System Quality Requirements Description Chapter

This chapter lists the quality requirements of the architecture with their name and comment. A paragraph dedicated to each requirement is displayed with the sub-requirements of the requirement. It also displays the objects constrained by the requirement.

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.

See:

- ✓ Creating a Solution Master Plan
- ✓ Displaying the Gantt Chart
- ✓ The NSV-8 Report Template
- ✓ The NSV-8 Gantt chart Chapter
- ✓ The NSV-8 Systems Evolution Description Chapter
- ✓ The NSV-8 System Evolution Comparison Chapter

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 window expand the **System Views** and **NSV-8 Systems Configuration Management** folders.
2. Right-click the **All System Master Plans** folder and select **New > Master Plan**.
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**.
   The new Master Plan appears in the tree under the **All System Master Plans** folder.

A **Planned Item Periods** folder is automatically created with the Master Plan. Three basic milestones are created with the "milestone-oriented" Master Plans:

- As-Is,
- To-Be,
- Distant Future.

Displaying the Gantt Chart

The Gantt chart is automatically created with the master plan.

To view the Gantt chart of a master plan:

1. From the NAF navigation tree expland **System Views > NSV-8 > All master plans**.
2. Right-click the newly created master plan and select **Open 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:

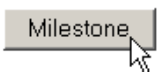
- Create milestones
- Create planned configuration capabilities
- Add the time periods of objects to be planned.
- Indicate the life-cycle status of the planned objects.

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 objects 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.

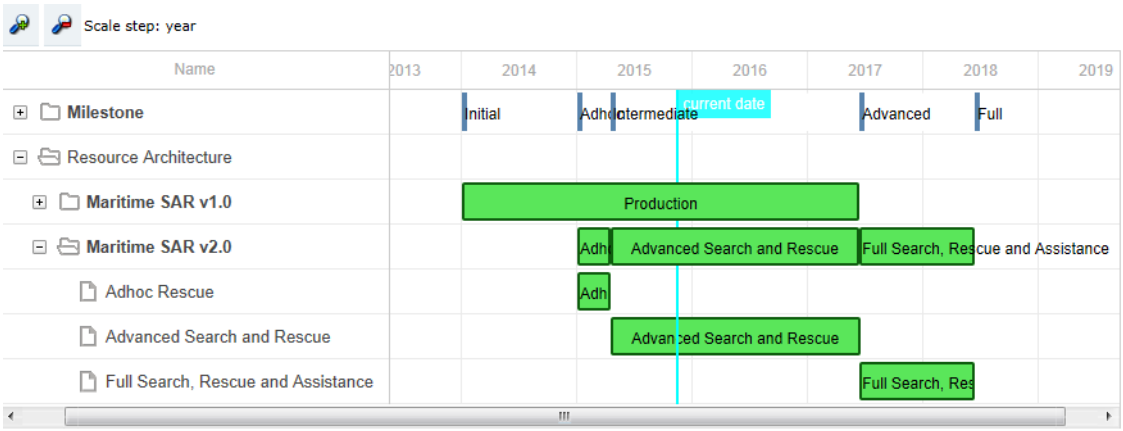
To add objects:

1. Above the master plan table, click the **Add Object** 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

The NSV-8 Report Template

The NSV-8 report template gives a graphical understanding of the evolution planned for the system.

The first parameter indicates the architecture to be analyzed.

The second parameter, Master Plan Subset, which is optional, indicates the master plans to be analyzed. Groups of master plans are considered in report chapters to enable plan comparisons (see the chapter below). If no master plan is set, all the master plans of the architecture are taken into account and the collected master plans are compared individually.

A similar report is also displayed in the property pages of the master plan. It can be used as an input tool to create and modify the milestones and the planned items. This is known as the Gantt Chart Chapter.

The NSV-8 Gantt chart Chapter

This chapter gives an automatic graphical representation of the analyzed master plans. A Gantt chart is displayed for each master plan.

The NSV-8 Systems Evolution Description Chapter

This chapter lists all the systems planned within the context of one or more master plans. The purpose of this chapter is to summarize planned items in relation the scenarios involved (What are the systems added? What is removed? What is maintained?).

The NSV-8 System Evolution Comparison Chapter

This chapter lists all the systems planned within the context of one or more master plans. The purpose of this chapter is to enable a comparison of the planned items in relation to the scenarios involved (What are the systems added? What is removed? What is maintained?). It also displays conflicts that exist among planned time periods.

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.

The NSV-9 Report Template

The NSV-9 report template shows the impact on technology of system evolution.

- The first parameter indicates the architecture to be analyzed.
- The second parameter, Master Plan Subset, which is optional, indicates the master plans to be analyzed. Groups of master plans are considered in the report to enable plan comparisons. If no master plan is set, all the master plans of the architecture are taken into account and the collected master plans are compared individually.

The NSV-9 Technology Forecast Chapter

This chapter gives a graphical representation of the impact of system evolution on the technology proposed in different master plans. A Gantt chart is displayed for each master plan.

The NSV-9 Technology Forecast Description Chapter

This chapter shows the impact of the system evolution proposed in different master plans on the technology. Its purpose is to enable a comparison of the planned items in relation to the scenarios involved (What are the systems added? What is removed? What is maintained?). It also displays conflicts that exist among planned time periods.

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.).

The NSV-10a Report Template

The NSV-10a report template is supplied with one parameter: the architecture on which the report is based. There are no additional parameters involved. The template comes with two chapters: The NSV-10a Resources Constraints Specification chapter and the NSV-10a Rules Linked to System Items chapter.

The NSV-10a Resources Constraints Specification Chapter

This chapter lists the constraints and requirements attached to system items of the architecture. This is an alphabetical list with an additional column for the comments. A paragraph is added for each listed constraint and requirement to explain their constrained objects.

If constraints and requirements are created from objects in Operational node structure diagrams (NOV-2), these constraints and requirements can be retrieved in the Systems Rule Model report provided the "View Type" property of the constraint is set to System. The constraints are then retrieved in the **System**

Constraints folder while the requirements are retrieved in the **System Requirements** folder of the NSV-10a navigational tree.

The NSV-10a Rules Linked to System Items Chapter

It is also possible to generate a chapter that lists all the potential constraints linked to a system item.

This chapter displays a table with an alphabetical list of the constraints that are linked to system items. This helps the user determine if the type of constraints linked are defined according to the scope defined for this system architecture. This definition is based on the "View Type" property.

If a constraint is not typed as System, a warning icon is displayed for this constraint in the **Scope** column. Although not typed as system, the constraint appears in the table because it is linked to a system item.

This happens if the constraint was not created from the NAF navigation tree but from a diagram and then attached to a system item of this diagram.

The " View Type" property for rule with a warning can, however, be changed from the table (right click the constraint **> Properties > NAF > NAF State Level > NAF Architecture Type >** etc.).

- Refresh the chapter to show the changes.

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.

NSV-10b Implementation

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-10b Template

The NSV-10b report template comes with the following parameter: the **Architecture** parameter that indicates the analyzed architecture.

The NSV-10b Resources State Transition Description Chapter

This chapter lists all the state machines attached to tasks (activities). The chapter starts with a simple table of the state machines (short name, long name, comment) and the linked tasks.

For each state machine with a describing diagram or at least one event, a dedicated paragraph is generated which has the name of the state machine as the title and which displays the comment and the (commented) describing diagrams. The

paragraph lists the events (name and comment) in a table. The successions are listed with their name, comment and the source and target events.

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.

The NSV-10c Report Template

The NSV-10c report template provides a report of the interaction scenarios of the architecture.

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

The parameter used for the report is the Architecture to be analyzed.

The NSV-10c Resources Event-Trace Description Chapter

This chapter lists all the interaction scenarios in a table. The table displays the short name, the long name and the comment of the interaction scenarios.

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

A specific paragraph is added for each interaction scenario which has detailed information. This paragraph contains:

- The diagrams describing the interaction scenario
- The list of participating system or human role instances (name and comment)
- The list of message instances

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.

The NSV-11a Report Template

The NSV-11a report template uses two parameters:

- The Architecture parameter, which indicates the architecture to be analyzed.
- The Data Model Subset parameter, which is optional. This parameter is used to reduce the scope of the study to a subset of the data models defined in the architecture. If this parameter is not set, all the data models defined in the architecture are taken into account.

The NSV-11a Logical Data Model Chapter

The NSV-11a logical data model chapter lists the operational data models of the architecture. Each data model is listed with its entities and attributes and the associations between the entities.

The NSV-11a Logical Data Model Hierarchy Chapter

The Logical Data Model Hierarchy chapter provides the hierarchy structure of the logical data models of the architecture.

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.

The NSV-11b Report Template

The NSV-11b report template uses two parameters:

- The Architecture parameter, which identifies the architecture to be analyzed.
- The Database Subset parameter, which is optional. This parameter is used to reduce the scope of the study to a subset of the databases defined in the architecture. If this parameter is not set, all the databases defined in the architecture are taken into account.

The NSV-11b Physical Data Model Chapter

The Physical Data Model chapter lists the databases modeled to support the storage of data with their tables and column details.

A database table is first displayed with the short name, long name and comment of each database.

- A paragraph is dedicated to each database. This paragraph includes:
- The comment of the database.
- The diagrams describing the databases.
- The associated data models.
- The list of tables contained in the databases. This list is a table with the short name, the long name and the comment of each table contained in the database.

The database paragraphs are followed by the table paragraphs. Each table which has one of the following items is inserted in a dedicated paragraph with:

- The comment of the table.
- The list of columns (name and comment).
- The list of indexes (name and comment).
- The list of keys (name and comment).

NSV-12 Service Provision

The Service Provision subview (NSV-12) is designed to illustrate which systems contribute to the provision of which services.

This subview the result of a mapping of systems, identified in NSV-1, to services defined in NSOV-2. If more detail is required beyond the system level, system functions, defined in NSV-4, can be mapped to services as well. This mapping can be either direct or through an intermediary mapping from NSV-4 system functions to NSV-1 systems. If the concept of a system is used to mean the more general concept of a type of resource, it is allowed to include concepts reflecting other types of resources, such as capability configurations, physical assets and roles.

The NSV-12 Report Template

The NSV-12 report template uses three parameters:

- The Architecture parameter, which identifies the architecture to be analyzed
- The Physical Resource Subset parameter, which is optional. This parameter is used to reduce the scope of the study to a subset of the physical resources defined in the architecture. If this parameter is not set, all the physical resources defined in the architecture are taken into account.
- The Service Subset parameter, which is also optional, indicates the services to be analyzed. If this parameter is not set, all the services defined in the architecture are taken into account.

The NSV-12 Services to Applications Traceability Matrix Chapter

The Services to Applications Traceability Matrix chapter displays a matrix with the applications that support the different services of the architecture.

The NSV-12 Services to Artifacts Traceability Matrix Chapter

The Services to Artifacts Traceability Matrix chapter displays a matrix with the artifacts that support the different services of the architecture.

The NSV-12 Services to Resource Architectures Traceability Matrix Chapter

This chapter displays a matrix which shows the resource architetures that support the selected services

You can add or remove links by clicking in the cells of the matrix.

The NSV-12 Services to Org-Units Traceability Matrix Chapter

This chapter displays in matrix form the org-units that support the selected services of the architecture.

The org-units are displayed in columns while the services are displayed in rows.

You can add and remove links by clicking in the cells of the matrix.

NAF Programme Views Subviews

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

- ✓ NPV-1 - Programme Portfolio Relationships
- ✓ NPV-2 - Programme to Capability Mapping

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.

All this information can be found in the different reports of the NPV-1 subview.

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

- ✓ The NPV-1 Report Template
- ✓ The NPV-1 Project Hierarchy Chapter
- ✓ The NPV-1 Project Dictionary Chapter
- ✓ The NPV-1 Hierarchical Project Dependencies Chapter
- ✓ The NPV-1 Project Dependencies Chapter

Managing Projects and Project Diagrams

Creating Projects

To create a project:

1. In the NAF navigation tree, expand the **Programme Views > NPV-1 Programme Portfolio Relationships** folders.
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 > Project Composition Diagram**.
2. In the Insert toolbar, click the **Project** button and click in the project frame to create several projects.
3. Create **Project Dependencies** between the created projects if needed.
4. 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.*

The NPV-1 Report Template

Report Parameters

The NPV-1 report template uses three parameters:

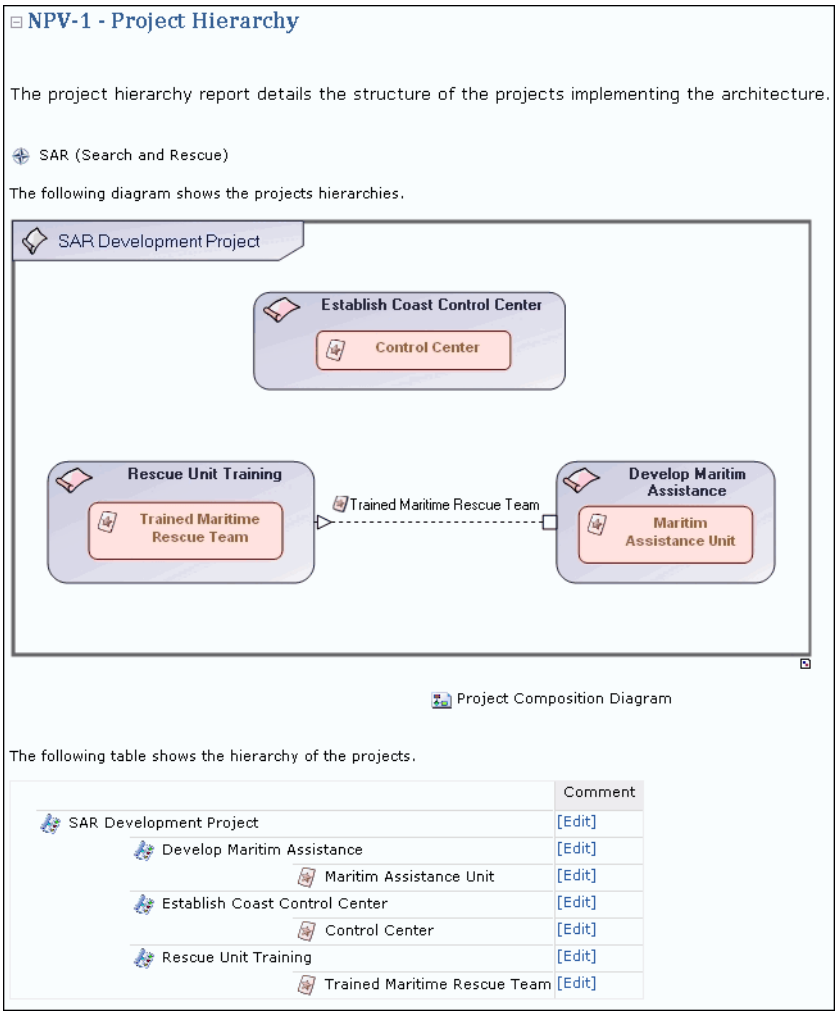
- The **Architecture** parameter, which is the architecture to be analyzed. This is the sole parameter that enables the retrieval of all the root projects of the architecture.
- The **Projects** parameter, which is optional. This parameter is used to reduce the scope of the study. The selected projects must of course belong to the architecture being analyzed. If no project has been selected, all the projects of the architecture are taken into account.
- The **Column Projects** parameter. This is also optional parameter. This parameter enables the creation of a table with different sets of projects for rows and for columns.

The NPV-1 Project Hierarchy Chapter

The project hierarchy chapter shows the structure of the projects implemented in the analyzed architecture.

The Project composition diagram is displayed to show the structure of the projects. This diagram is followed by a table of the project hierarchy. The Root project is

displayed with its comment and its sub-projects. The deliverables and comments of the sub-projects are also displayed.



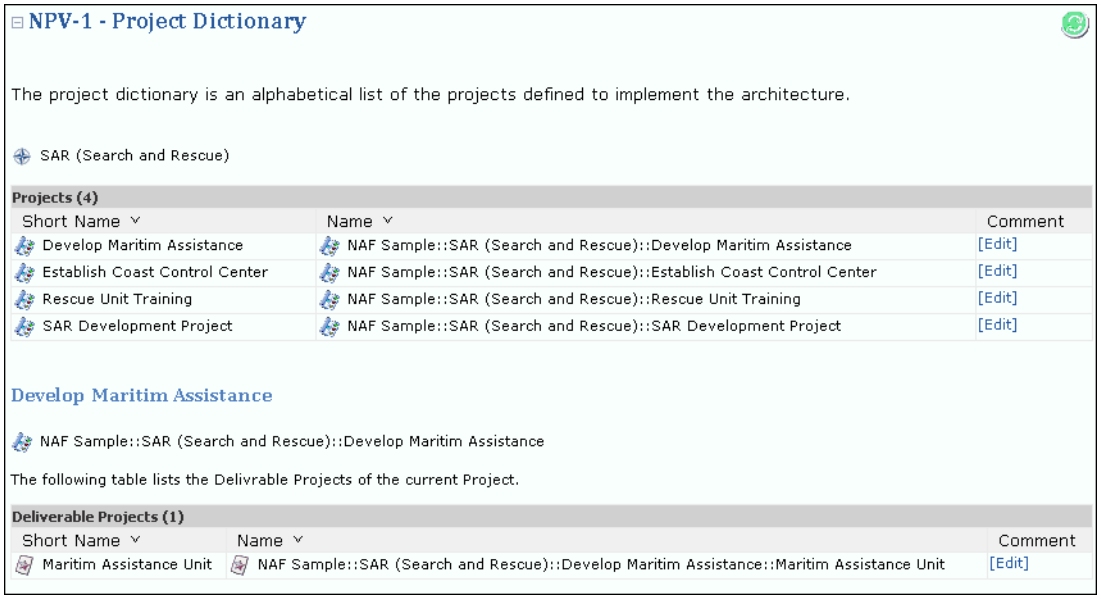
Example of a Project Hierarchy Report

The NPV-1 Project Dictionary Chapter

The project dictionary report provides a list of all the projects of the architecture in alphabetical order.

This list is presented in table with the short name, long name and comment of each project.

A paragraph is then dedicated to each project. In this paragraph, the deliverables expected of the project are listed. The project composition diagram is also displayed.



NPV-1 - Project Dictionary

The project dictionary is an alphabetical list of the projects defined to implement the architecture.

SAR (Search and Rescue)

Projects (4)

Short Name	Name	Comment
Develop Maritim Assistance	NAF Sample::SAR (Search and Rescue)::Develop Maritim Assistance	[Edit]
Establish Coast Control Center	NAF Sample::SAR (Search and Rescue)::Establish Coast Control Center	[Edit]
Rescue Unit Training	NAF Sample::SAR (Search and Rescue)::Rescue Unit Training	[Edit]
SAR Development Project	NAF Sample::SAR (Search and Rescue)::SAR Development Project	[Edit]

Develop Maritim Assistance

NAF Sample::SAR (Search and Rescue)::Develop Maritim Assistance

The following table lists the Delivrable Projects of the current Project.

Deliverable Projects (1)

Short Name	Name	Comment
Maritim Assistance Unit	NAF Sample::SAR (Search and Rescue)::Develop Maritim Assistance::Maritim Assistance Unit	[Edit]

Example of a Project Dictionary Chapter

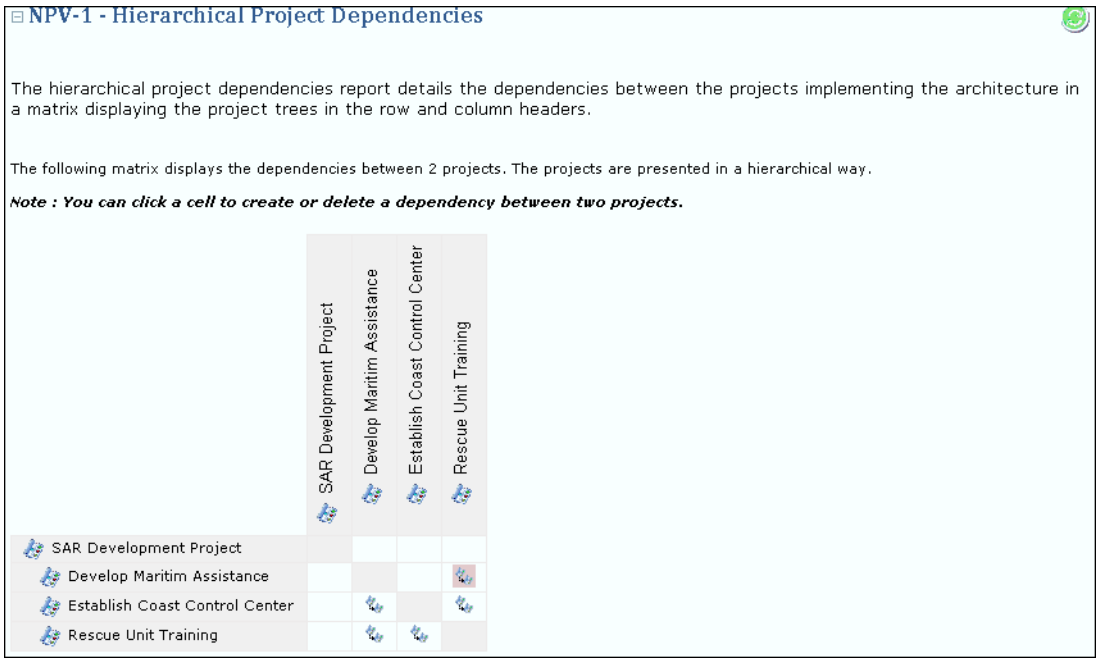
The NPV-1 Hierarchical Project Dependencies Chapter

Projects are sometimes dependent.

The Hierarchical Project Dependencies chapter shows the dependencies of projects in the architecture. This report allows you to use the subset parameters to generate appropriate sized tables. The projects are displayed according to the defined project hierarchy.

The report information is displayed in a matrix with project trees as rows and column headers.

Dependencies between two projects are represented by icons. An icon appears in the matrix between projects linked by a dependency. You can add or remove dependencies by clicking in the different cells of the matrix.



Example of a Hierarchical Project Dependencies Chapter

The NPV-1 Project Dependencies Chapter

The Project Dependencies chapter also shows the dependencies of projects in the architecture, however, project hierarchy is not displayed.

The dependencies are represented as icons in a matrix with projects as rows and column headers. An icon appears in the matrix between projects linked by a dependency. You can add or remove dependencies by clicking in the different cells of the matrix.

NPV-2 - Programme to Capability Mapping

The purpose of the Programme to Capability Mapping (NPV-2) subview is to primarily support the acquisition and fielding processes, including the management of dependencies between projects and the integration of all relevant project and programme elements to achieve a capability as defined in NATO capability packages (CP).

In NPV-2, programmes and projects are mapped to capabilities to show how the specific projects and programme elements help to achieve a NATO capability, as defined in a CP. Projects are mapped to the capability for a particular time period. Projects may contribute to multiple capabilities and may mature across time periods.

The NPV-2 subview analysis can be used to identify capability redundancies and shortfalls, highlight programme phasing issues, expose organizational or system interoperability problems, and support programme decisions, such as when to phase out a legacy system.

The NPV-2 Report

Report presentation

The Project Deliverables x Capability Increments report of NPV-2 shows the relationship between the capability configuration states defined in NSV-8 and the deliverables produced by the projects that enable reaching the expected states.

This report consists of a matrix with checkmarks that are used to represent mappings between capability configurations (resource architectures) and project deliverables.

You can add or remove mappings by clicking in the different boxes between the capability configurations and the project deliverables.

Report parameters

The NPV-2 report template uses two parameters:

- The **Architecture** parameter, which identifies the architecture to be analyzed.
- The **Projects Subset** parameter, which is optional. This parameter reduces the scope of the study to a subset of the projects selected. The selected projects must belong to the architecture. If no project is selected, all the projects of the architecture are taken into account.
- The **Capability Increment** parameter, which is also optional. this parameter is set with the time periods that you wish to include in the analysis. If no time period is selected, all the time periods of the architecture are taken into account.

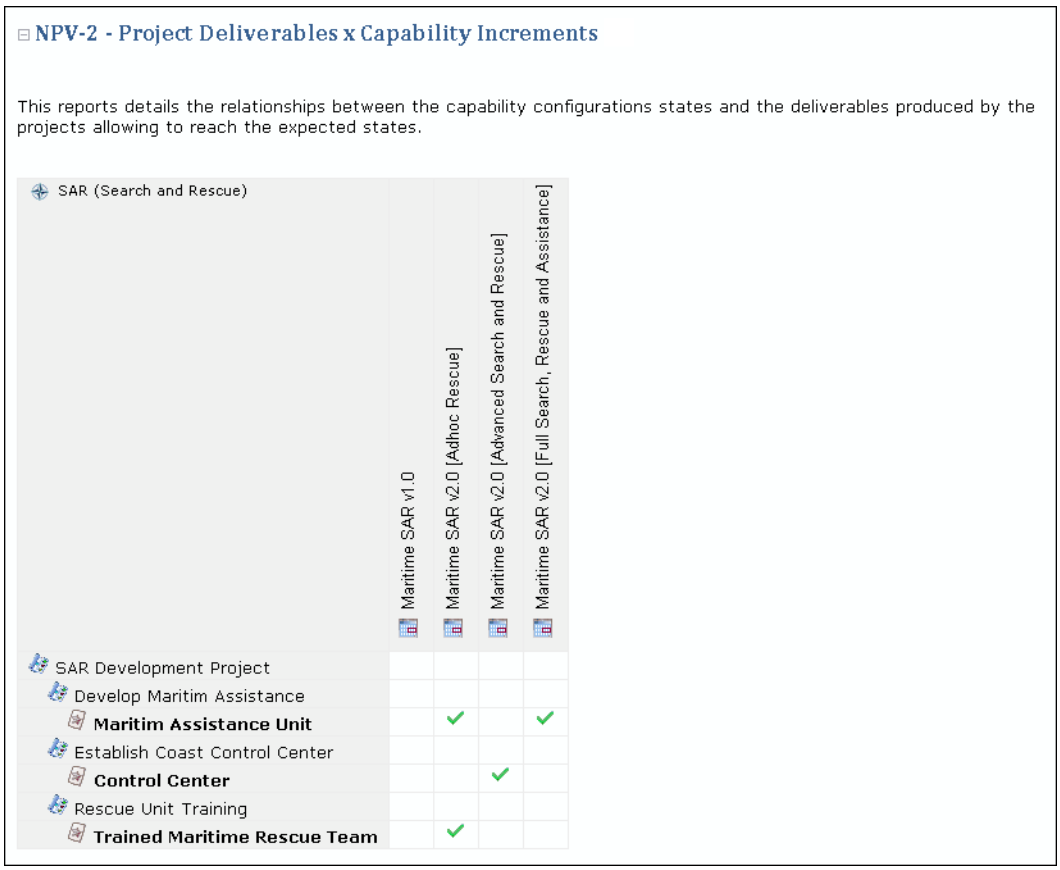
The NPV-2 Project Deliverables X Capability Increments Chapter

This chapter shows the relationship between the states of capability configurations and the deliverables produced by the projects implemented to attain the expected states.

This relationship is presented in a matrix with:

- a project tree with deliverables as rows
- time periods as columns.

Checkmarks appear in the matrix to show where a particular deliverables is expected, to ensure that a state of a system part is available. The deliverables are not directly connected to the state but to the time periods of the items that are linked to the states. The time periods are the intermediate notions that group both the states and the items.



NPV-2 - Project Deliverables x Capability Increments

This reports details the relationships between the capability configurations states and the deliverables produced by the projects allowing to reach the expected states.

SAR (Search and Rescue)	Maritime SAR v1.0	Maritime SAR v2.0 [Adhoc Rescue]	Maritime SAR v2.0 [Advanced Search and Rescue]	Maritime SAR v2.0 [Full Search, Rescue and Assistance]
SAR Development Project				
Develop Maritim Assistance				
Maritim Assistance Unit		✓		✓
Establish Coast Control Center				
Control Center			✓	
Rescue Unit Training				
Trained Maritime Rescue Team		✓		

Example of a Project Deliverables X Capability Increments Chapter

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.

✓ Defining the Standards
✓ The NTV-1 Template Report
✓ The NTV-1 Standards Profile Chapter

Defining the Standards

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.
```

The NTV-1 Template Report

The NTV-1 report template produces a report that gives a graphical understanding of the evolution of the standards in relation to planned systems of the architecture.

The template uses two parameters to generate the report chapters:

- The first parameter is the **Architecture** which indicates the architecture to be analyzed.
- The **Standard Subset** parameter. This is an optional value that is used to reduce the scope of the study. This parameter is set with the standards described in the architecture. In this case, the NTV-1 report focuses on this set of standards. The **HOPEX** objects used to set this parameter are standards.

The NTV-1 Standards Profile Chapter

This chapter presents the standards and guidelines that apply to the analyzed architecture(s). Paragraphs NSV-1 to each standard follow, listing the elements that use the standard in question.

NTV-2 Standards Forecast

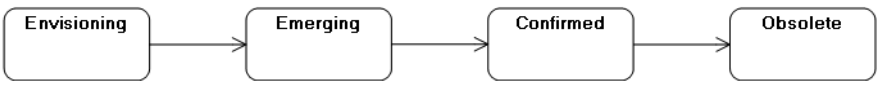
The purpose of the Standards Forecast subview (NTV-2) is to:

- 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
✓ Customization
✓ The NTV-2 Report Template
✓ The NTV-2 Standards Forecast Chapter
✓ The NTV-2 Standard Forecast Description Chapter

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. For more information, see the **HOPEX Planning** user manual, "Describing a Master Plan" chapter, "Object life cycle status" paragraph.*

The NTV-2 Report Template

The NTV-2 report template produces a report with chapters that give a graphical understanding of the evolution of the standards in relation to planned systems of the architecture.

The template uses two parameters to generate the report chapters:

- The first parameter is the **Architecture** which indicates the architecture to be analyzed.
- The second parameter, **Master Plan Subset**, is optional. This parameter indicates the master plans to be analyzed. Groups of master plans are considered in reports to enable plan comparisons. If no master plan is set, all the master plans of the architecture are taken into account and the collected master plans are compared individually.

The NTV-2 Standards Forecast Chapter

This chapter presents a view of the standards in a Gantt chart.

This chart provides a detailed representation of the possible conflicts that may occur during the evolution of the standards and the systems planned in the NSV-8 subview or the technologies defined in the NSV-9 subview.

The NTV-2 Standard Forecast Description Chapter

The technical standard forecast description chapter displays a table with milestones as column headers and Standards in rows. The states occupy the cells between the standards and the milestones.

This chapter is particularly useful when the NTV-2 master plan is also an aggregation of a sub-master plan. In this case the table displays the standards as they are planned in the sub-master plans.

It is also very useful when it shows the potential availability conflicts between standards and other dependent items.

NTV-3 - STANDARD CONFIGURATIONS

Report Parameters

The report template used to generate the NTV-3 report is composed of two parameters.

- The **Architecture** parameter, which specifies the architecture to be analyzed.
- The **Standard Subset** parameter. This is an optional value that is used to reduce the scope of the study. This parameter is set with the standards described in the architecture. In this case, the NTV-3 report focuses on this set of standards. The **HOPEX** objects used to set this parameter are standards.

The NTV-3 Standard Configurations Chapter

This chapter lists the standards planned in the NTV-2 view that are defined via the resource architectures. When these resources are linked to capabilities they describe capability configurations and then the related standards match up with the configurations that have been agreed on.