HOPEX NAF

User Guide

HOPEX V2

INTRODUCTION TO HOPEX NAF

Enterprise architecture (EA) is a term related to the practice of modeling the elements of business strategies, business processes and supporting technologies, and policies and infrastructures that make up an enterprise. Its role is to provide support for decision-making in the enterprise strategy. The architecture is also used as an analysis tool to develop new capabilities and structure organizations and to optimize processes and spending. This is typically what the HOPEX Modeling tool helps to do. Regardless of the modeling tool used, a set of frameworks have been designed to be used either as guidelines to implement enterprise architecture projects (for example, TOGAF) or to define the deliverables expected as a result of such projects. This is the case of NAF.

The NATO Architecture Framework (NAF) is not an enterprise architecture. 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 interoperable military capabilities. The framework ensures that architectures developed by NATO and its member countries can be compared and related across NATO and National boundaries. NATO common funded programs are to comply with the NAF to promote systems interoperability

Bearing all this in mind, 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.

The framework details all the views addressing the different aspects of enterprise architectures (the overall purpose, the operational perspective, the system perspective and the technical perspective).

The **HOPEX NAF** is designed to generate NAF deliverables. It is based on the standard features of the HOPEX Modeling tool, 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 **HOPEX NAF** is based on the NAF v.3 release.

The present guide details how all views defined in the NAF standard are implemented in the **HOPEX NAF** product.

✓ "Modeling Architectures in a Repository", page 14

MODELING ARCHITECTURES IN A REPOSITORY

The **HOPEX** Modeling suite is designed to facilitate modeling and analyzing enterprise architectures. When several architectures are to be designed, the issue of architecture management is raised straight away. Why model several architectures in a single repository? This may have to do with one of three things:

✓ "Architecture Composition", page 14
✓ "Evolution Traceability", page 14
✓ "Evolution Traceability", page 14

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**'s 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 the **HOPEX** Modeling tool to create NAF deliverables.

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 timeframe 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", page 19
✓ "NATO Capability View Subviews", page 20
✓ "NATO Operational View Subviews", page 21
✓ "NATO System-Oriented View Subviews", page 23
✓ "NATO System View Subviews", page 23
✓ "NATO Technical View Subviews", page 26
✓ "NATO Programme View Subviews", page 27
✓ "NATO All View Subviews", page 19
✓ "NATO Capability View Subviews", page 20
✓ "NATO Operational View Subviews", page 21
✓ "NATO System-Oriented View Subviews", page 23
✓ "NATO System View Subviews", page 23
✓ "NATO Technical View Subviews", page 26
✓ "NATO Programme View Subviews", page 27

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

Code	Subview Name	Comment
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
NCV-4	Capability Dependencies	Capabilities, capability functions, describes the dependencies between capabilities, defines logical groupings of capabilities

Code	Subview Name	Comment
NCV-5	Capability to Organisational 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
NCV-7	Capability to Services Mapping	Describes the mapping between capabilities and the services that these capabilities enable

NATO Operational View Subviews

Code	Subview Name	Comment
NOV-1	High Level Operational Concept Descriptiona	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	Organisational Relationships Chart	Organizational context, role or other relationships among organizations

Code	Subview Name	Comment
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
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	Services to Operational Activities Mapping	Identifies operational activities supported by services
NSOV-4	Service Orchestration	Identifies and describes how services are used in the execution of operational activities and how services are used to support operational processes.
NSOV-5	Service Behaviour	Specifies the function and behaviour of individual services

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.

Code	Subview Name	Comment
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
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 Activity 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

Code	Subview Name	Comment
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
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

THE HOPEX IMPLEMENTATION

As an enterprise architecture tool, the **HOPEX** Modeling application can be used to implement a NAF project. Many different frameworks have been designed to help implement enterprise architecture projects. Even though common concepts can be retrieved in each framework, vocabularies may differ and sometimes the same words can be used with slight variations. **HOPEX** has its own history and the vocabulary contained in the **HOPEX** Modeling tool metamodel reflects **HOPEX**'s vision. The tool also supplies a rich set of diagrams that are dedicated to modeling the different levels of enterprise architecture, however, because of the history, the diagrams are either not named or are not exactly equivalent to those described in the NAF standard.

NAF Subview Generation

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

Even though there are three kinds of reports, the **HOPEX** implementation of the NAF views is consistent; it always relies on the report platform that allows for the generating of subview contents in the **HOPEX** Modeling tool both as working and input tools, and in HTML and RTF formats.

Objectives and Requirements

The NOV-6a and NSV-10a subviews identify the business, operational and system rules that apply to meet the objectives of systems, system hardware/software items, and/or system functions, enterprises, missions, operations, businesses, and architectures. These rules can act as either constraints or requirements.

To ensure that objectives and requirements are displayed in the NOV-6a and NSV-10a subviews the **Objectives and requirements modeling** option must be activated in **HOPEX**. By default, this option is activated.

If by chance, however, you are not able to view your objectives and requirements activate the **Objectives and requirements modeling** option:

- Select **Tools > Option > Business Process and Architecture Modeling > Objectives and requirements modeling**.

Tagging Constraints, Contents and Requirements as Operational or System Items

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:

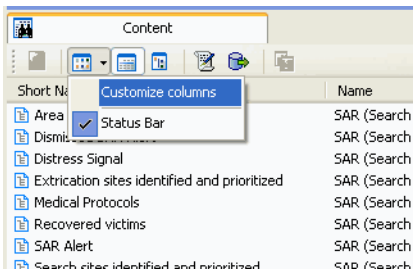
- NAF - All Operational Contents retrieved by linked objects
- NAF - All System Contents retrieved by linked objects
- NAF - All Operational Constraints retrieved by linked objects
- NAF - All System Constraints retrieved by linked objects
- NAF - All Operational Requirements retrieved by linked objects
- NAF - All System Requirements retrieved by linked objects

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 **NAF - All Operational Contents retrieved by linked objects** 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.

Variations

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.

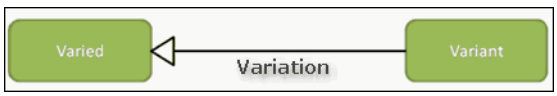
Variants inherit the elements of the object from which they derive. Inherited elements are displayed in the navigation tree with an inheritance arrow next to them.

For more information on variations, see the **HOPEX Common Features** user guide "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.*

Graphical visualization of variations

In diagrams, variations are displayed as inheritances. The inheritance arrow available in UML class diagrams is used to represent variations, and goes from the variant item to the varied item.



Activating variations

For architecture modeling using the **HOPEX NAF** product, users are encouraged to display variations. The option that allows users to display and use variations is deactivated by default.

To activate the variation option:

- In the **HOPEX** menu select **Tools > Option > Business Process and Architecture Modeling > Activate variations**.

IMPORTING SOLUTION PACKS

As a standard the **HOPEX NAF** is delivered with certain solution packs that can be imported to enhance the architecture descriptions and analyses.

Preparing Files for Import

The solution packs are delivered in compressed files and must be decompressed before they can be imported into a repository.

Three compressed files provide the NAF solution packs:

- **NAF** (NAF.exe): contains the "Metamodel Customization" (metamodel renaming), "Sample" and "Data" solution packs.
- **HOPEX Architecture** (Hopex.exe): contains the "HOPEX Architecture framework".
- **Military Terms** (Military Terms.exe): contains the Military Terms Dictionary.

To decompress a compressed file:

1. In your **HOPEX** installation folder open the **Utilities** then the **Solution Pack** or **Solution Pack.R3** folders.
2. Double-click the compressed file, for example **NAF.exe**.
3. In the dialog box that opens click **Extract**.

The dialog box closes and the solution pack files are extracted to the Solution Pack folder.

To import a solution pack, see **HOPEX Power Supervisor** > Managing Objects > Importing Reference Frameworks > Importing a Solution Pack.

Metamodel Customizations

The **HOPEX** Modeling tool comes with its own set of concepts and vocabulary. To minimize the differences between the names of concepts in this tool and in the **HOPEX NAF**, a, solution pack can be imported into **HOPEX** that renames certain concepts using the NAF vocabulary. This renaming is only done in English. To retrieve the list of renamed terms, refer to the paragraph "MEGA Metamodel Renaming", page 280.

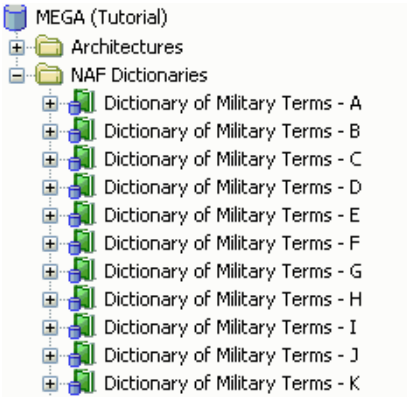
Using the Military Terms Dictionary

A dictionary is provided with the 2009 version of the **HOPEX** Modeling suite. The dictionary contains:

- Over 6000 terms with their definitions. These terms are all approved by the Department of Defense (DoD) and some of them (approx. 50%) are also approved by NATO.
- Over 6000 acronyms used within the context of DoD projects.

Before you can access this dictionary from the **NAF** navigation window you must first import it into the **HOPEX** interface from the **HOPEX Administration** application.

After import, the DoD dictionary is instantly available in the NAF navigation window of the **HOPEX** application.



HOPEX Architecture Framework

HOPEX Architecture framework has been created with a list of pre-existing artifacts and provided as an add-on.

> *An artifact is any material item of a system that is neither software or an organizational unit (where organizations are people). Depending on the context and user preferences, artifacts can be represented as platforms, components and any physical item that occupies space and has attributes.*

Artifacts are normally used within the context of architecture modeling to represent all the non-human components of an architecture. This concept is usually delivered with the **HOPEX Architecture** product, however, it can also be useful for NAF which deals to some extent with infrastructure modeling.

The HOPEX Architecture framework also gives access to communication protocols and resource architectures.

> *A resource architecture is a set of material and organizational resources required for a system. It describes the interactions between components and their means of communication. It also indicates the*

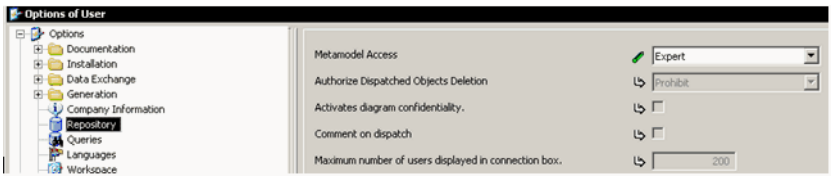
service ports necessary for allowing users to connect and integrate with it. This architecture can be modeled in a resource architecture diargam and usually contains sub-architectures, equipment and human resources.

To access these elements in your working repository you must import the **HOPEX Architecture** framework.

Viewing artifacts and resource architectures

If after importing the HOPEX Architecture framework you are unable to access artifacts and resource architectures, you will need to activate the "Business Process and Architecture Modeling / Infrastructure Modeling post-2009 SP2" option in the HOPEX Options.

You should also ensure that you have Expert level access to the metamodel (**Tools > Options > Repository > Metamodel Acess > Expert**)



OVERVIEW OF HOPEX NAF

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

- "The NAF Start Page", page 36
- "The NAF Navigation Tree", page 37

See also:

- "Creating NAF Architectures", page 40

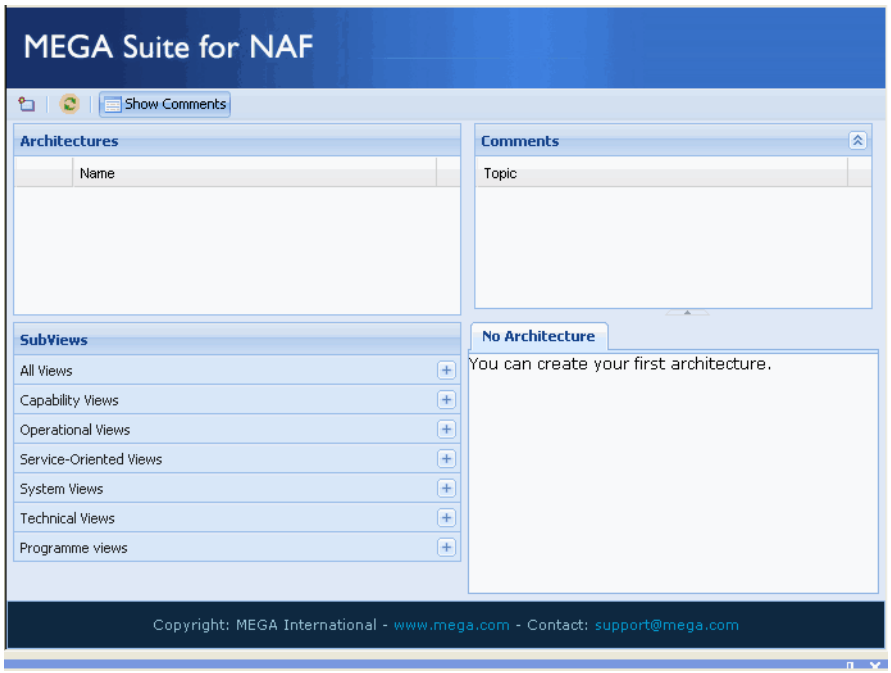
THE NAF START PAGE

The first tool is a dedicated start page that shows all the NAF Architectures designed and the related subviews. This page allows the user to access all the main items corresponding to the subviews defined in the NAF standard. It also facilitates creating these main items in the appropriate way to enable NAF subview generation (this will be described in later sections).

To launch the **HOPEX NAF** Start page:

1. In the **HOPEX Help** menu select **Start Page**.
2. On the **HOPEX** Start page click **Frameworks EA > NAF**.

The **NAF** Start page appears.



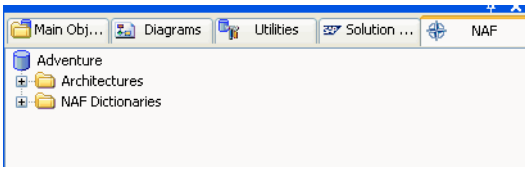
THE NAF NAVIGATION TREE

The second tool is a dedicated navigation tree. This tree shows all the architectures contained in the **HOPEX** repository. This is a more common tool for the **HOPEX** Modeling users since a large set of navigation trees are already available.

The **NAF** navigation tree starts navigation from two points: the architectures and the NAF dictionaries.

To launch the **NAF** navigation tree:

- In the **HOPEX** menu bar, click **View > Navigation Windows > NAF**.
  The NAF navigation window appears in the desktop.



By default, the NAF navigation window displays the repository in which the NAF architectures are located. It also contains an "Architectures" folder in which all the architectures of the current repository are located as well as a "NAF Dictionaries" folder which contains the official DoDAF terms and acronyms, if imported, approximately 50% of which are approved by NATO.

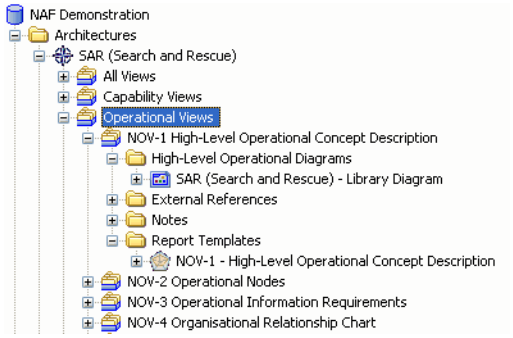
The Architecture Tree

In the NAF navigation window, the NAF architectures are found in the "Architectures" folder. Each architecture has seven main folders each of which corresponds to one of the seven NAF views:

- All Views
- Capability Views
- Operational Views
- Service-Oriented Views
- System Views
- Technical Views
- Programme 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 in question. 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.



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.

The **HOPEX NAF** is supplied with a **HOPEX** dictionary of the official DoDAF definitions. It contains more than 6000 terms as well as over 6000 acronyms and abbreviations, most of which have been approved by NATO and are therefore relevant in the context of NAF Projects with military orientation.

Viewing the terms of the NAF Dictionary

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

- In the NAF navigation window, 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.

Viewing the terms associated to an architecture

To view the terms associated to a particular architecture:

1. In the NAF navigation window, expand the architecture concerned.
2. Expand **All Views > NAV-2 Integrated Dictionary** and double click the **NAV-2 Integrated Dictionary** report icon.
3. In the window that opens in the desktop to the right of the navigation window, click **NAV-2 - Complete Integrated Dictionary**.
4. Scroll down the list to see all the terms associated to the architecture with their accompanying definitions and/or comments, if they exist.

An open book icon indicates the terms of the dictionary.

Linking terms to an architecture

To link a term to an architecture:

1. In the NAF navigation window, select the architecture concerned.
2. In the **HOPEX** menu bar, select **View > Edit Windows > Properties**. The properties dialog box of the architecture appears in the frame on the right of the desktop.
3. Select the **General** tab **> Definitions** sub-tab.
4. In the NAF navigation window, expand the folder in which the term to be attached is located.
5. Drag and drop the term in the **Properties** window of the architecture.

> ☛ *The terms in the "Definition" frame are the terms that define the architecture. The terms in the "Related Definition" frame are the terms that relate to the architecture.*

Ideally, you should link terms to the different items of the architecture, for example, an operational node, service, etc.).

The procedure is the same as for adding a term to an architecture except that you will instead add the term in the properties dialog box of the item concerned.

CREATING NAF ARCHITECTURES

A NAF Architecture can be created from the different tools previously discussed (the NAF navigation tree or the NAF Start page).

To create a NAF architecture from the NAF Start page:

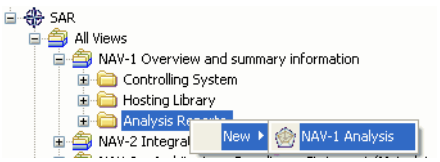
1. On the NAF start page, click the **New** icon.

2. In the dialog box that appears, enter the name of the new architecture and select the Modeling method to be used to create the architecture.
3. Click **Finish**.

The new architecture appears in the list of architectures on the start page and in the "Architectures" folder in the NAF navigation tree.

When you create a NAF architecture, a set of related objects are also created. These objects are contained in a new **HOPEX** library. The architecture itself is also contained in this library which is used as a container for all the architecture items. All these new objects are defined to represent a given perspective of the architecture, for example the operational or the system perspective.

Reports are not created with each new NAF architecture. A command is available on the architecture which allows you to create the desired reports for the different NAF Views.



All the objects created are transparent to the NAF user. They are used by the system to group all the main items created within a given architecture. The navigation tree and the NAF start page give direct access to these objects.

Once an architecture is created, the Views and subviews can be accessed via the NAF navigation tree or the NAF start page. Detailed information is provided for each subview later on in this document.

You can also set an architecture as the default architecture. In so doing, all the objects created will be created within the context of this architecture and consequently are contained in the library of the architecture.

The Default Architecture

When an architecture is created, several other objects are also created. To ensure that all these objects are taken into account in reports, they must all be enclosed within the architecture scope.

Items created within the framework of a hierarchy are automatically retrieved from the starting point of the hierarchy. For example, in an organizational chart, all org-units belonging to a hierarchy are retrieved from the top level org-unit. However, many other objects are not created within the framework of hierarchies. For example, data flows between two operational nodes do not belong to any tree. In this case, the **HOPEX** Modeling tool ensures that these objects can be safely retrieved by placing them in a **HOPEX** library. **HOPEX** libraries define a partition of the **HOPEX** repository and can be seen as boundaries for given modeling projects. If a library has been set as the default library, all new objects are created within its boundary.

The **HOPEX NAF** functions similarly, however, for the NAF expert the main item is the architecture not the library. So, when creating a new architecture a new library is automatically created which will receive all new objects. When you set a default architecture, the corresponding library is automatically set as the default library and so all new created objects are contained in the library of the NAF architecture.

Setting the default architecture

To set an architecture as the default architecture:

1. In the NAF navigation window, right-click the architecture you wish to set as the default architecture.
2. In the pop-up menu that appears, select **Set as default**.

A green check mark appears next to the name of the architecture concerned.

> ☛ *You can also set the default architecture from the **Home** navigation window. In this case, right-click the library in which all data pertaining to your architecture is based and select **Set as Default**. The results are the same as above.*

GENERATING NAF SUBVIEW DOCUMENTATION

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 the user can generate documentation in relation to the expected format.

The different types of documentation that can be generated are:

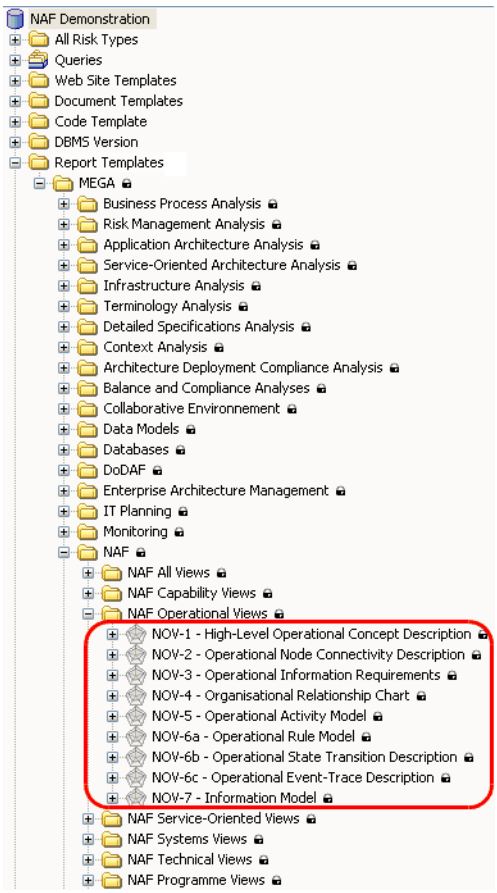
REPORTS

All the documentation generated is based on the report template that corresponds to the different NAF subviews. These report templates are located in the **Utilities** navigation window of the **HOPEX** desktop.

To view these templates:

- Expand the **Report Templates > HOPEX > NAF** folders.

The templates for the different subviews along with their report generators, parameters and so on are located in the different Views folders.

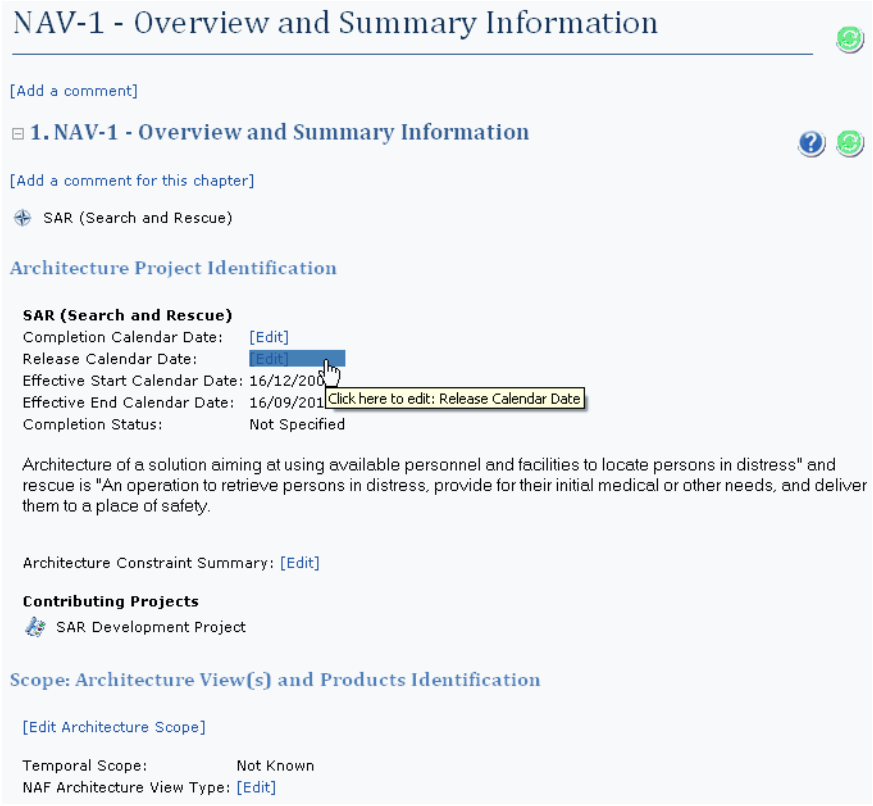


To create a new report for a given architecture, the user creates a new report instance based on the expected subview template. For example, to create an NOV-1 document, a report is created from the NAF architecture based on the NOV-1 Report Template.

Depending on the subview, the report template supplies one or more chapters. Each chapter addresses a part of the expected subview. The idea is to be able to build reports with the essential information.

Several reports can be created for a given subview. Each report addresses a different subset of the architecture. This allows for generating smaller documents which are easier to read.

From the **HOPEX** Modeling tool desktop, reports can be used for understanding architecture designs but also for designing architectures. In the second perspective, the user can change the properties of the objects cited in the document since the contextual menu is always accessible. Additional accesses are also provided depending on the generated content. One such access is 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.



It is also possible to specify two distinct reports:

- A report for the reader and
- A report as a working tool

For example, in the NAV-2 template a report generates the dictionary of the commented items of a given architecture. A second report lists all the items of the architecture and so the designer will be able to identify the uncommented items and add the missing information.

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.

To create a report:

1. From the **NAF** 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.

   ☛ *If you left the **Open report at validation** option checked, the report automatically opens when creation is complete.*

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 **NAF** 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.
3. Proceed in like manner to create all the reports for the other Views of the architecture.

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 NAF navigation window of the **HOPEX** tool, right-click the report to be converted and select **New > RTF** (or **PDF**).
   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 Different Versions

Converting reports into MS Word reports is quite 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 or PDF 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 external references or business documents allowing you to easily access them from the **HOPEX** desktop.

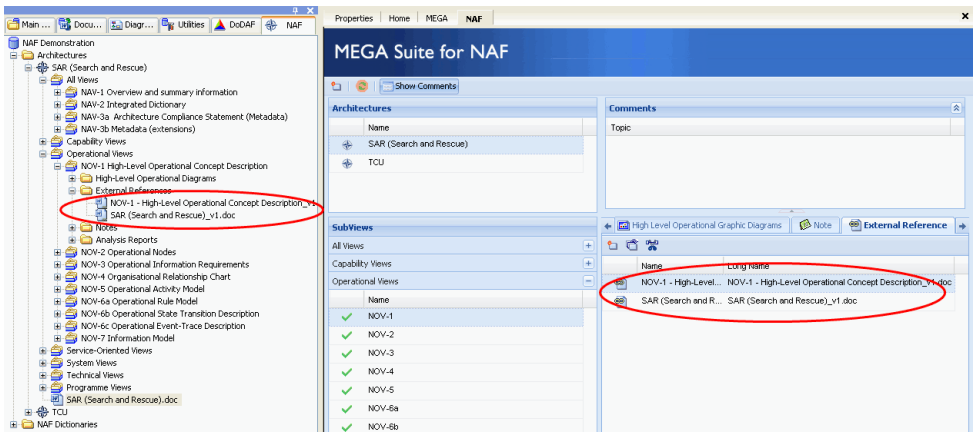
Adding external references

To attach a document as an external reference:

1. Right-click the element (report, architecture) to which the external reference is to be added and in the pop-up menu that appears, select **Properties**.

   ☛ *When adding an external reference to the overall architecture itself, the pop-up menu is accessible immediately after right-clicking the architecture.*

2. In the **Properties** dialog box that appears, select the **General** tab and the **References** sub-tab.
3. Click the **New** icon and in the dialog box that appears select **File > Next**.
4. Click the arrow in the "File" field and select **Browse** to specify the location of the document to be added as an external reference.
5. Click **Next** when you have specified this location.
   The name of the document and the complete address of the document are recapitulated in the dialog box that appears.
6. If you are satisfied with the information in the dialog box, click **Finish**.
   The document appears in the **References** sub-tab in the list of external references. If the external reference was added to the architecture itself, this reference also appears in the NAF navigation window under the NOV-1 subview as well as in the NAF start page for this same subview.



Example of external references added to an architecture

Accessing external references

You can access versions of documents saved as external references from the **Properties** dialog box of the corresponding element.

To access external references:

1. In the **NAF** navigation window, select the architecture and right-click the element concerned.
2. In the pop-up menu that appears select **Properties**.

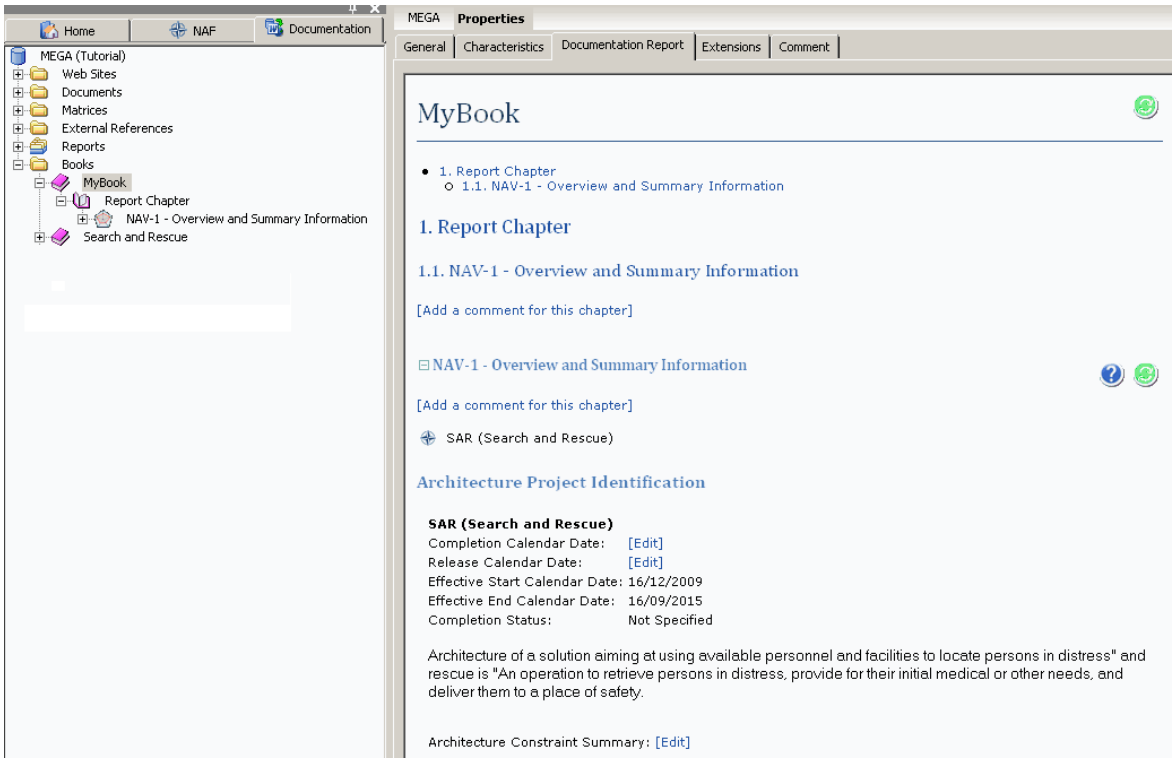
3. In the **General** tab, **Reference** sub-tab, double-click the external reference concerned to open it.

Using the BOOK to Organize Document Display

HOPEX has introduced the BOOK concept which facilitates structuring the way information is pesented.

This concept can be used to organize your documents. For example, you could decide to create a book which presents reports, in a particular order, in chapters with diagrams, etc.

For more information on books, see the **HOPEX Common Features** user guide, "Using Books" chapter.



Example of a Book

STATIC HTML DOCUMENTATION

The complete set of reports attached to a NAF architecture can be exported in a static HTML file. The advantage of this format is that it contains all the information regarding the architecture and the hyperlink system enables navigation from item to item. Since the documentation is in a static format, it can be compressed and sent to other persons. A CHM file can also be generated.

The static HTML documentation is based on a web site template (called NAF). The starting point is the NAF Architecture. All architecture items are inserted from this starting point.

✓ "Generating HTML Documentation", page 50
✓ "Contents of the Generated Web Site", page 51

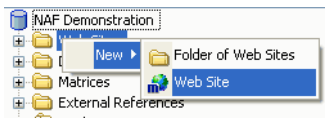
Generating HTML Documentation

To create a HTML documentation from a Web site template:

1. In the **HOPEX** desktop, select the **Documentation** navigation window tab.

   ☛ *If the Documentation navigation window is not visible in the desktop, in the **HOPEX** menu bar select **View > Navigation Windows > Documentation**.*

2. In the **Documentation** navigation tree, right-click the "Web Sites" folder and select **New > Web Site**.



   The **Create Web Site** dialog box appears.

   By default, the Web site is generated in the "Intranet" folder. You can select an alternative folder by clicking the ... button. It is recommended that you create a sub-folder for each Web site generated, for example by creating folders with the same name as your Web site in the "Intranet" folder.

   ☛ *The content of the "Intranet" folder is deleted at each generation. As such, when you have obtained a site with which you are satisfied and want to keep, remember to copy it to another folder.*

3. In the **From Web site template** list box, click the button.
4. Select a Web site template, in this case "NAF" and click **Next**.
5. Select the architecture on which the web site is based by clicking next to the term "Architectural Description" then on the arrow that appears.
6. Select the word list to view all the libraries that exist in your environment.

7. Select the correct architecture from the list displayed and click **OK**.
   The name of the architecture appears in the web site creation dialog box.
8. Click **Finish**.

The web site is created but not generated. You must generate the web site for visualization.

To generate the Web site:

1. In the **Documentation** navigation window, right-click the site concerned.
2. In the pop-up menu that appears, select **Generate**.
   A dialog boxes appear, indicating the progress of the Web site generation.

When the Web site has been generated, a dialog box proposes opening the site. This Web site is static.

You can access the generated Web site at any time by double clicking it in the **Documentation** navigation window, **Web Sites** folder.

Contents of the Generated Web Site

The generated Web site contains different information. It displays the NAF architecture to which the Web site corresponds. There are also different tabs which correspond to the different Viewpoints of the architecture. These tabs give access to the reports of each View, as well as to the notes and external references. There is also a generic page which displays each item of the architecture and indexes. Access to the generic page is from the "Site Map" link.

For more information on generating Web sites, see the **HOPEX Power Studio - Publisher** technique article, "Creating a Web site" chapter.

NAF All Views Subviews

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

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. NAV-1 includes assumptions, constraints, and limitations that may affect high-level decision processes involving the architecture.

This information included in this subview also includes the Eenterrpise Phases of the architecture.

The information included in this chapter also includes Libraries (see "The NAV-1 Library Organization Chapter", page 58) and 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 **HOPEX Planning** user guide, "Describing a Master Plan" chapter, "Using Timelines" paragraph.

The NAV-1 Overview and Summary Information Chapter

This chapter lists all the overview information relating 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.

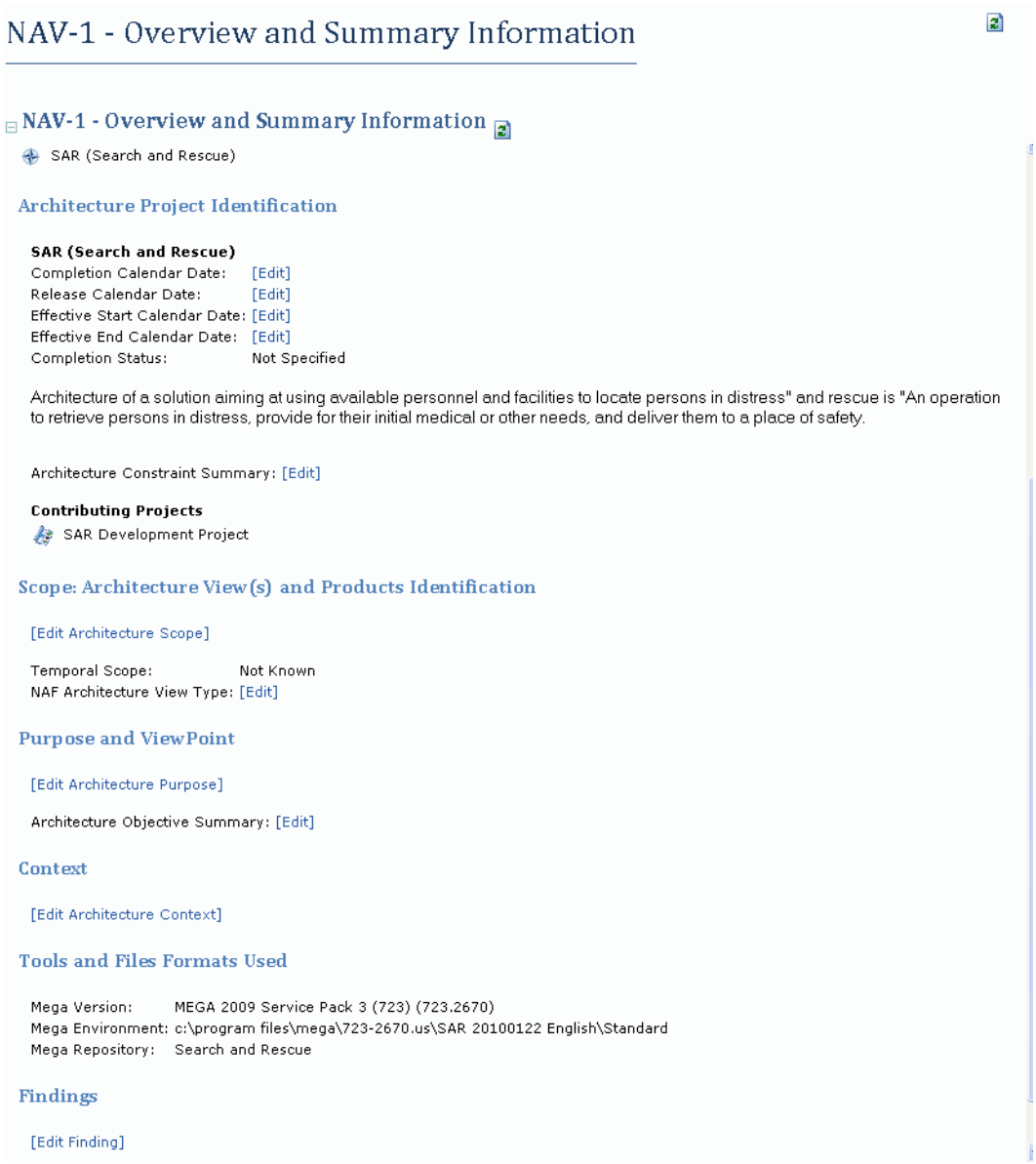
- **Context**: Describes the setting in which an architecture exists. Context includes things such as mission, doctrine, relevant goals and vision statements, operation concepts, scenarios, information assurance context, other threats and environmental conditions, and geographical areas addressed, where 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.

If the report is generated in the **HOPEX** Modeling tool, additional information is inserted to help the user identify the appropriate properties to include, in order to obtain this information in each section of the report. This additional information is removed in generated documents and Web sites.

Example

The figure below shows the example of an NAV-1 Overview and Summary Information chapter generated from the SAR architecture sample. The chapter is

generated within the **HOPEX** Modeling tool so a few [**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 their 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 therefore shows how the library containing the NAF architecture is structured hirerchically along with the other libraries that it accesses for data.

The NAV-1 Enterprise Phase Composition Hierarchy Chapter

The NAV-1 - Entreprise 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 and in table form. 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 NAV-1 Enterprise Phase Dictionary Chapter

The NAV-1 - Entreprise 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", page 60
✓ "The NAV-2 Complete Integrated Dictionary Chapter", page 61

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.

NAV-2 - Integrated Dictionary

Analysis Input

NAV-2 - Integrated Dictionary

SAR (Search and Rescue)

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

Nb. definitions:5

Name	Type	Comment
Vehicule	Artifact	A conveyance that transports people or objects
Tactical C2	Business Function	Provide overall management and coordination of SAR services, from activation through demobilization.
End	Event Type	**End** is the category of all **Event Steps** that manifest the end of a **Course**.
Direct Search & Rescue Tactical Operations	Functional Process	In response to notification of entrapment, provide management and coordination of SAR services, through demobilization for single or multiple teams.
SAR OV1	Note	This view set the scene by illustration the search and rescue operation at sea, which involves a yatch in distress. A monitoring unit picks up the distress calls of the yatch and passes them to a Command and Control (C2) center which coordinates the operation which involves helicopters, a naval ship and a RNLI lifeboat.

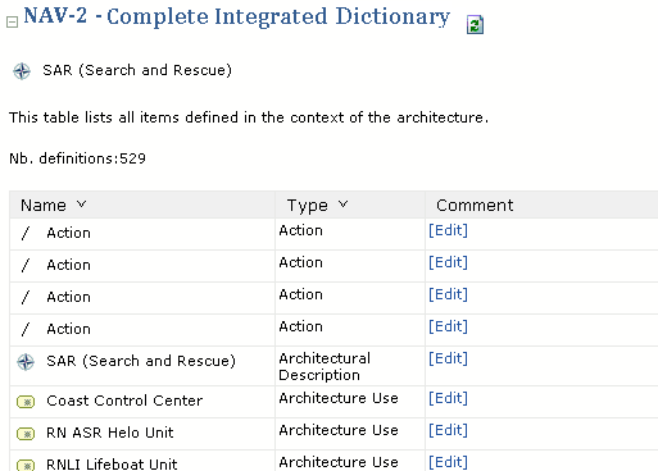
Example of an Integrated Dictionary Chapter

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 on the **HOPEX** Modeling desktop 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,*

*right-click the report and in its **Properties** dialog box, select the "Report Chapters" tab and change the check-marks.*



NAV-2 - Complete Integrated Dictionary

SAR (Search and Rescue)

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

Nb. definitions:529

Name	Type	Comment
Action	Action	[Edit]
Action	Action	[Edit]
Action	Action	[Edit]
Action	Action	[Edit]
SAR (Search and Rescue)	Architectural Description	[Edit]
Coast Control Center	Architecture Use	[Edit]
RN ASR Helo Unit	Architecture Use	[Edit]
RNLI Lifeboat Unit	Architecture Use	[Edit]

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.

Three 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).

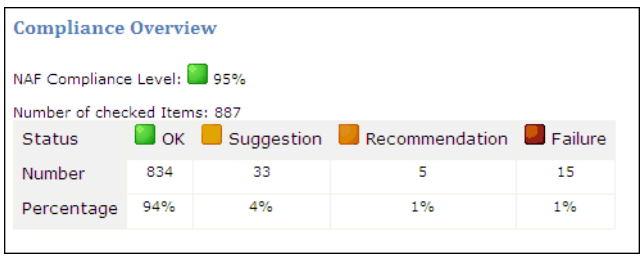
An assessment is then carried out on the object with all the rules that are attached to it.

- 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.

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:

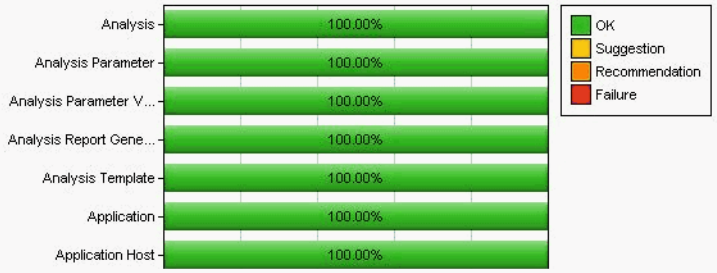
- The global percentage of object compliance. Depending on the percentage value a colored square indicates the degree of compliance
- The number of objects checked
- A pie chart with the status percentages: OK, Suggestion, Recommendation, Failure
- A table with the number and percentage of objects with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



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

Compliance Details

This graphic shows the compliance for each type of designed items.



Type/Status	Average Status	Worst Status	OK	Suggestion	Recommendation	Failure
Analysis	100%		49	0	0	0
Analysis Parameter	100%		46	0	0	0
Analysis Parameter Value	100%		50	0	0	0
Analysis Report Generator	100%		122	0	0	0
Analysis Template	100%		45	0	0	0
Application	100%		1	0	0	0
Application Host	100%		3	0	0	0
Architecture Use	100%		4	0	0	0
Artifact	100%		24	0	0	0
Artifact Component	100%		9	0	0	0
Business Service	100%		8	0	0	0
Business Service Component	100%		5	0	0	0
Capability	100%		10	0	0	0
Capability Composition	100%		8	0	0	0
Capability Dependency	100%		3	0	0	0
Content	100%		30	0	0	0
Control System	100%		1	0	0	0
Data Model	100%		2	0	0	0
Data Object	100%		14	0	0	0
DataType Package	33%		0	0	1	0
Entity (DM)	100%		20	0	0	0
Event	91%		21	0	0	2

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

Status	Object Name	Object Type	Details
	Data Types Reference	DataType Package	...
	End	Event	

Rule Status	Regulation	Rule Definition	Diagnosis
	BPMN regulation	*Requirement* The nature of the event shall comply with the sequence flows coming in and going out. An event shall have a correct nature depending of the incoming and outgoing sequence flows.	The event has some previous or next sequence flow that are not compliant with the event nature (Start, End or Boundary). The nature is not coherent with the sequence flows. (And) ✓ Boundary events should be connected to an internal event of the bound process ✗ End events should have an incoming sequence flow ✓ Intermediate catching event have an outgoing sequence flow ✓ Intermediate catching event have incoming sequence or triggering sequence flow ✓ Intermediate throwing event have incoming sequence flow ✓ Intermediate throwing event have outgoing sequence or triggered sequence flow ✓ No going in sequence flow for start event ✓ No going in sequence flows for boundary events ✓ No going out sequence flow for end event ✓ Start events should have an outgoing sequence flow

	Start	Event	...
	Activate Distress Device	Functional Activity	

Rule Status	Regulation	Rule Definition	Diagnosis
	BPMN regulation	*Suggestion* An activity must be implemented by an organizational process	✗ Activity/Procedure?
	BPMN regulation	*Suggestion* Any activity of BPMN (elementary tasks or sub-processes) shall be commented.	The comment for the BPMN activity is empty. ✗ Has the comment been filled in?

	Activate Search & Rescue	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", page 67
✓ "The NAV-3b NAF Metamodel Chapter", page 67

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.

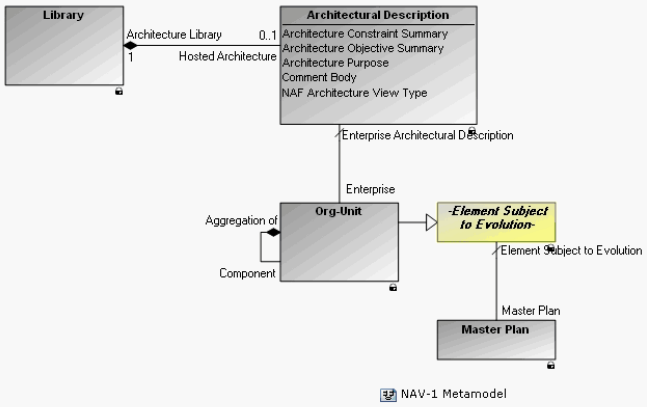
The NAV-3b NAF Metamodel Chapter

The NAF Metamodel chapter lists the metamodels of the **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 followed by 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. Definitions are also displayed for each metaclass.

The associations that can be made between metaclasses are also displayed in a table. For each association, the source and target metaclass as well as the source and target Roles are also displayed.

NAV-1



NAV-1 Metamodel

Concepts

Architectural Description

The following properties are available for this concept.

Properties (5)

Name	Comment
Architecture Constraint Summary	This text summarizes the limitations for the goal of the architecture.
Architecture Objective Summary	This text characterizes the aim of the architecture.
Architecture Purpose	This text characterizes the primary use of the architecture.
Comment Body	
NAF Architecture View Type	

Element Subject to Evolution

An element that motivates the elaboration of an evolution plan.

Library

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

Master Plan

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

Org-Unit

An org-unit represents a person or a group of persons that intervenes in the enterprise business processes or information system. An org-unit can be internal or external to the enterprise.
- An internal org-unit is an organizational element of the enterprise structure such as a department, a service, or a workstation. An internal org-unit is defined based on how detailed you require your view of the enterprise to be (cf org-unit-type). Example: financial management, sales management, marketing department, account manager.
- An external entity is an organization that exchanges flows with the enterprise. Example: Customer, Supplier, Government Office.

Associations

Source	Target	Source Role	Target Role
Org-Unit	Architectural Description	Enterprise	Enterprise Architectural Description
Element Subject to Evolution	Master Plan	Element Subject to Evolution	Master Plan
Library	Architectural Description	Architecture Library	Hosted Architecture
Org-Unit	Org-Unit	Component	Aggregation of

Example of a section of an NAV-3b NAF Metamodel Chapter

NAF Capability Views Subviews

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

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 NCV-1 subview is high-level in nature and does not specify the success criteria for an architecture. NCV-2 provides metrics against each capability which may be used to successfully measure fielded capabilities.

The main items dealt with in this subview are goals/objectives and the enterprise vision.

The enterprise vision describes the overall aims of an enterprise over a given period of time. This can be supported by a master plan in which the evolutions that help to achieve the overall objectives of the enterprise are described. The selected master plan is then used to describe capability availability and show the associations between enterprise phases and capabilities.

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

The Strategic Master Plans are located in the Enterprise Vision folder of the NCV-1 Capability Vision subview and the objectives are located in the Enterprise Goals folder.

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 Report Template", page 71
- ✓ "The NCV-1 Enterprise Vision Chapter", page 71
- ✓ "The NCV-1 Enterprise Vision Details Chapter", page 72
- ✓ "The NCV-1 Enterprise Goal Hierarchy Chapter", page 72
- ✓ "The NCV-1 Enterprise Goal Dictionary Chapter", page 72

The NCV-1 Report Template

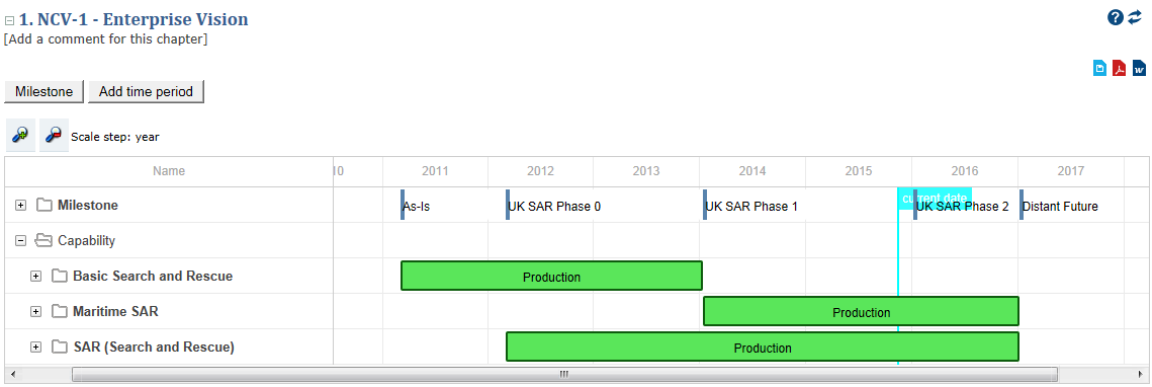
The NCV-1 report template comes with three parameters.

- 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.

The NCV-1 Enterprise Vision Chapter

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

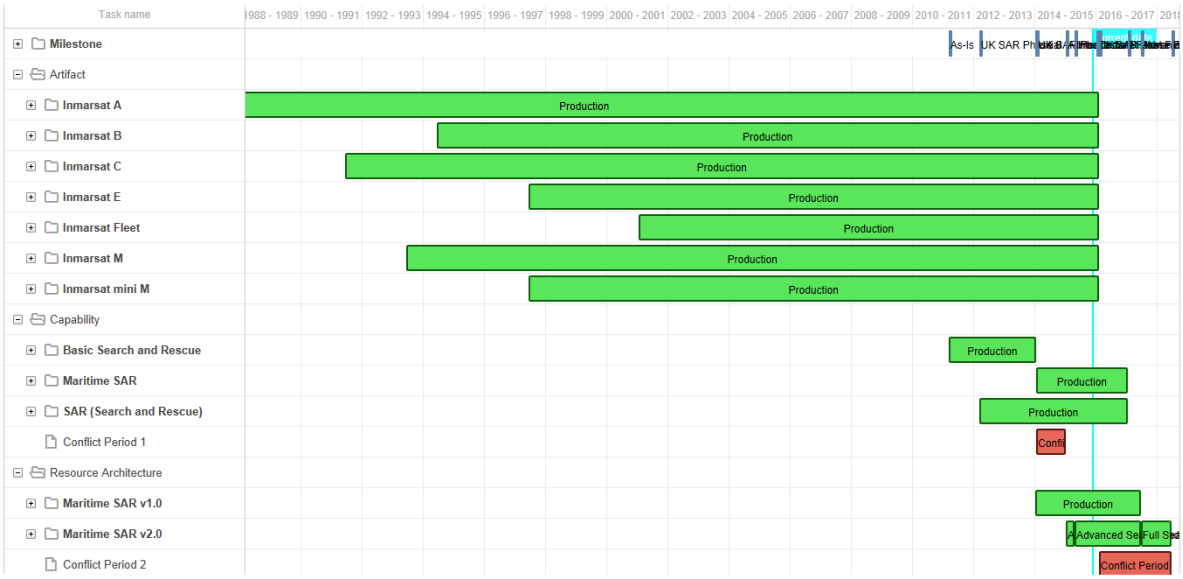
The data of the Gantt Chart can be modified in the report. For more information on creating Master Plans, see NCV-3.



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

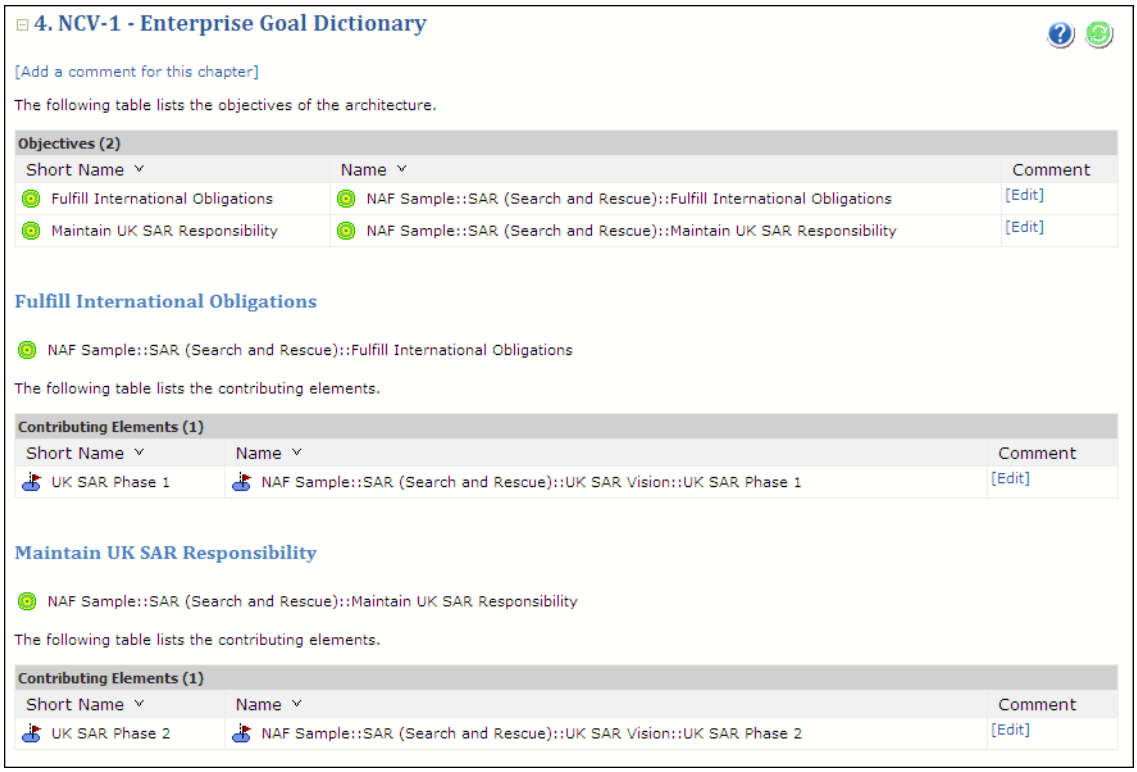
This chapter shows the hierarchy of the objectives of the architecture.

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. 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

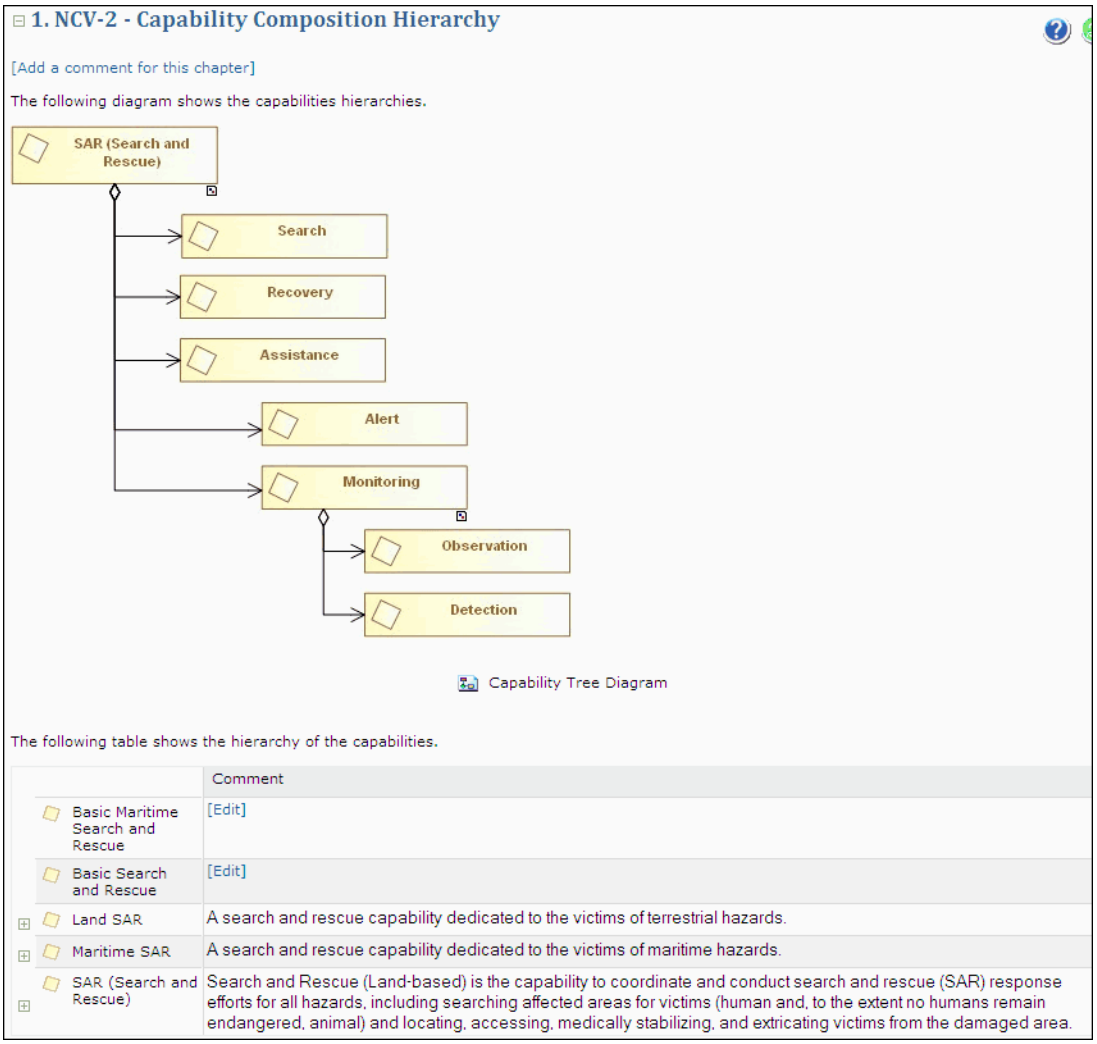
The NCV-2 report template comes with two parameters:

- 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 Chapte

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)

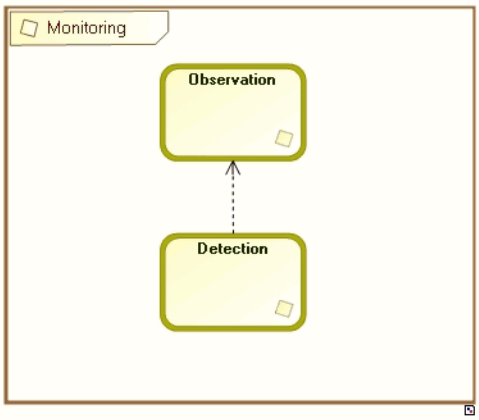
Short Name	Name	Comment
Alert (As in Land SAR)	SAR (Search and Rescue)::SAR (Search and Rescue)::Alert::Alert	[Edit] **Composition:** [Edit]
Assistance (As in Land SAR)	SAR (Search and Rescue)::SAR (Search and Rescue)::Assistance::Assistance	[Edit] **Composition:** [Edit]
Monitoring (As in Land SAR)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

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

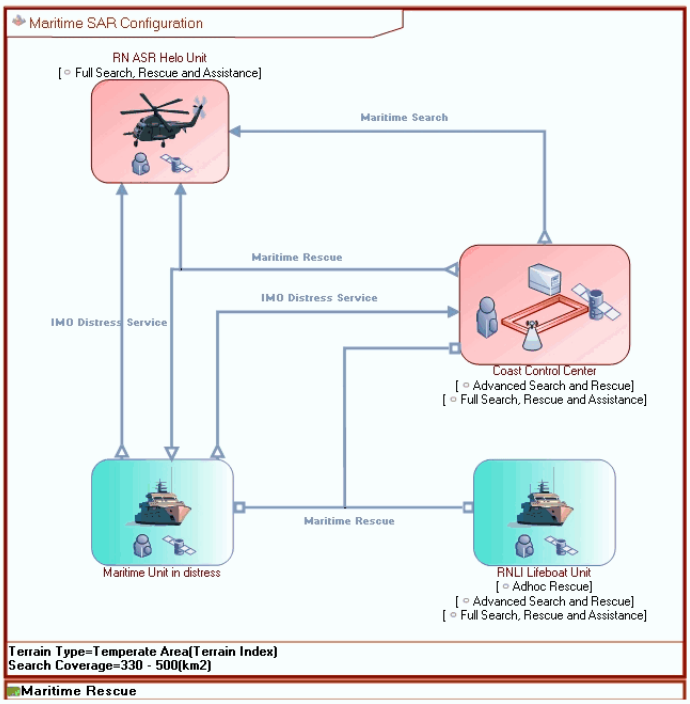
The following table lists the configured capabilities of the current capability configuration.

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

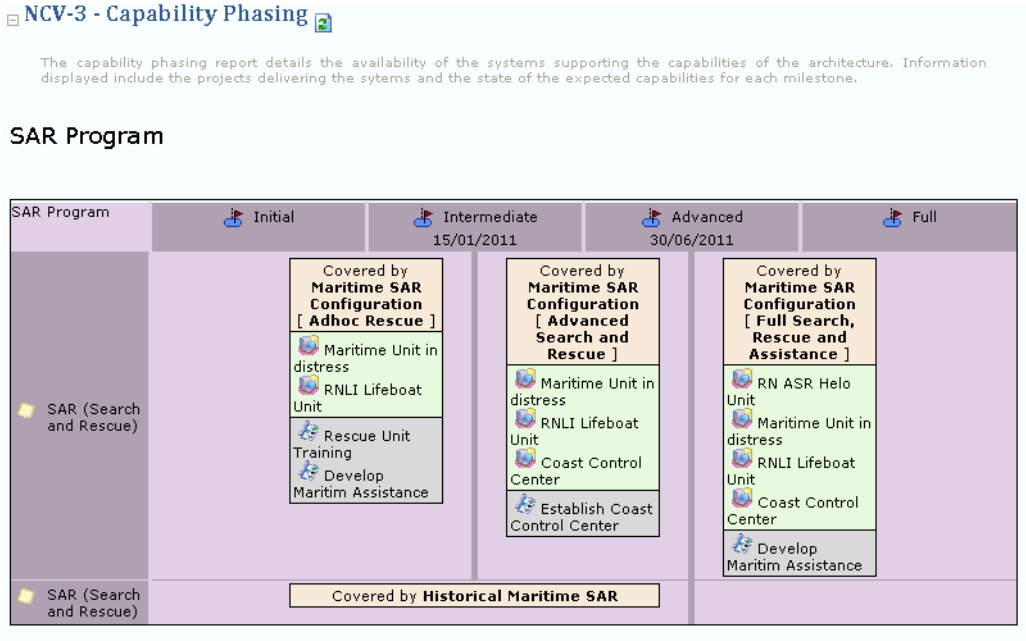


Example of a Capability Configuraton Dictionary Chapter

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

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

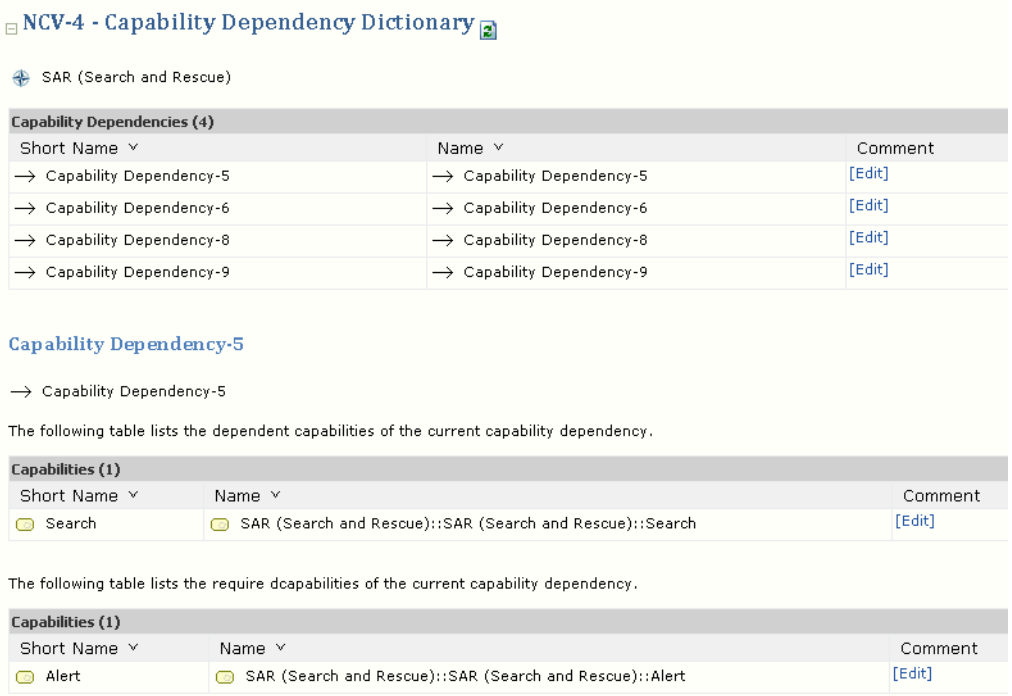
The NCV-4 report template comes with three parameters:

- 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.
- 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 fcapability dependencies of the architecture are taken into account.
- 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.

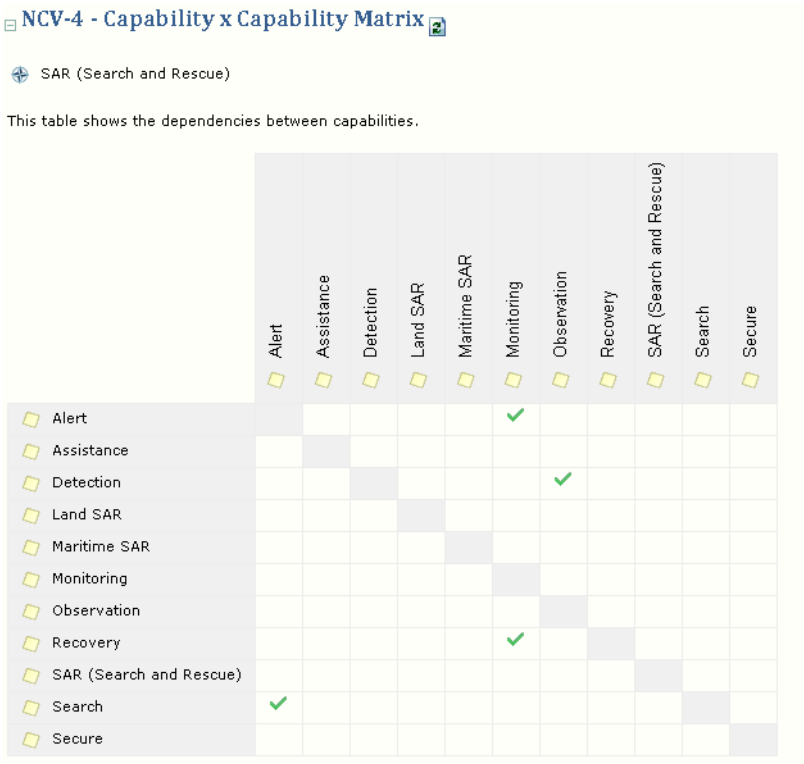
Capabilities (1)		
Short Name	Name	Comment
Alert	SAR (Search and Rescue)::SAR (Search and Rescue)::Alert	[Edit]

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											

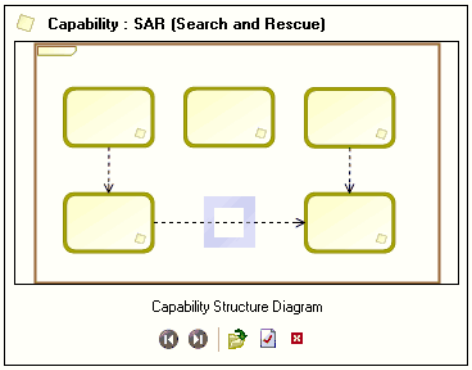
Example of a Capability X Capability Matrix

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.

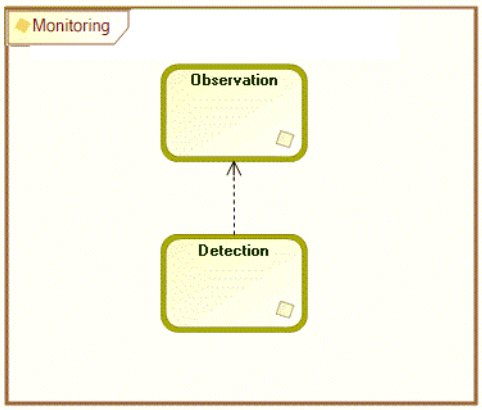
To create a dependency:

- 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

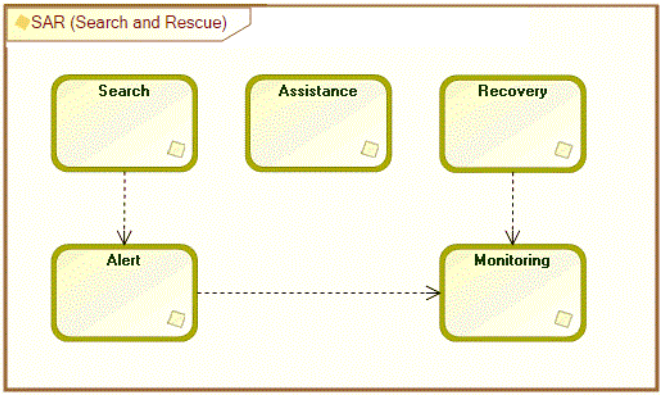
The name of the dependency is a combination of the name of the upper capability and those of the source and target capabilities of the dependency.

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

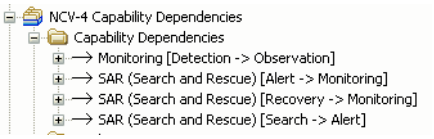
same structure diagram or where a composition is part of dependencies in different structure diagrams. Whatever the case, the naming rule is the same.

☛ *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 eneble 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", page 91
✓ "The NCV-5 Capability to Organizational Deployment Mapping Chapter", page 91

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 arhitectures 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 and the projects in rows. The projects are displayed with their start and end dates. These projects match the time period the organization is slated to use in the resource architecture.

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

You can click in the matrix to add or remove checkmarks.

Example of an NCV-5 Capability toOrganizational Deployment Mapping Chapter

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.

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:

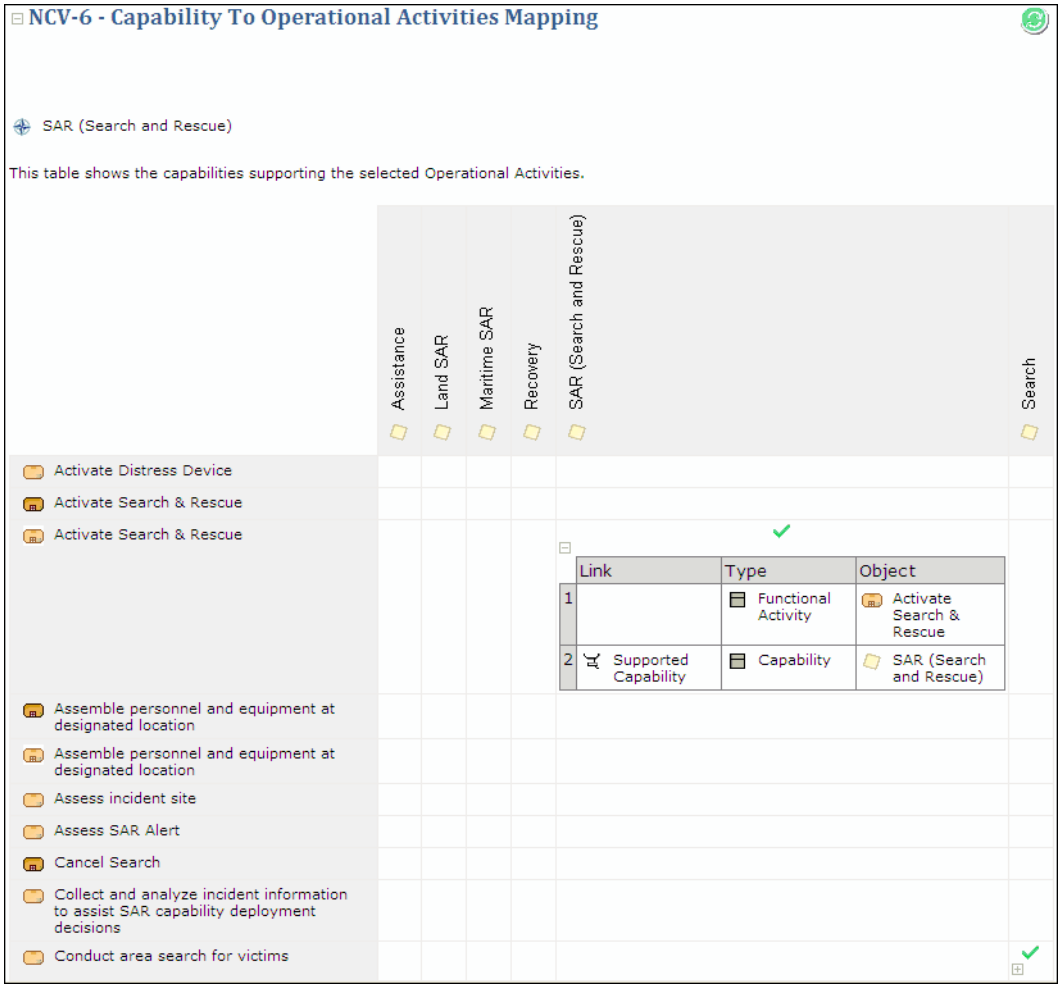
- The Architecture parameter, which specifies the architecture to be analyzed.
- The Capability Subset is an optional parameter. You can limit the scope of the study to a subset of the selected capabilities by setting one or more capabilities. All the capabilities of the architecture are used in the 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 proceses 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.

You can 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

NCV-7 Capability to Services Mapping

The NCV-7 capability to services mapping subview describes the mapping between capabilities and the services that these capabilities enable.

An NCV-7 subview shows the capabilities that are required for providing particular services, or which services can be provided with a given set of capabilities. The mapping is an aggregation of other NAF subviews. By combining NCV-6 (maps capabilities to operational activities) with NSOV-4 (maps operational activities to services), essentially a mapping between capabilities and services is obtained.

✓ "The NCV-7 Report Template", page 94
✓ "The NCV-7 Capability to Services Mapping Chapter", page 94

The NCV-7 Report Template

The NCV-7 report template comes with three parameters:

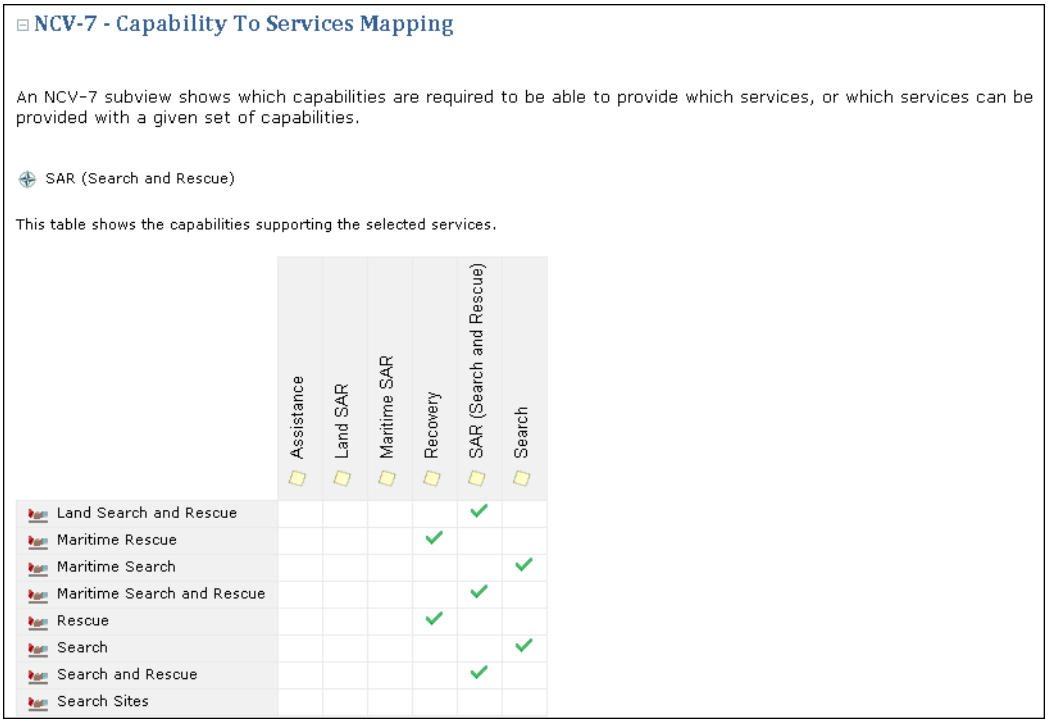
- The Architecture parameter, which specifies the architecture to be analyzed.
- The Business Service Subset parameter, which is optional. This parameter is set with a subset of all the business services in the NSOV-1 view. If this parameter is not set, all the business services of the architecture are taken into account.
- The Capability Subset parameter, which is also optional. This parameter is set with a subset of all the capabilities defined in the NCV-2 view. If this parameter is not set, all the capabilities of the architecture are taken into account.

The NCV-7 Capability to Services Mapping Chapter

This chapter displays a table with the selected capabilities in columns and the business services in rows.

For each capability-service couple, a checkmark is displayed if a link already exists.

You can click in the matrix to add or remove checkmarks.



Example of an NCV-7 Capability to Services Mapping Matrix

NAF Operational Views Subviews

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

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 free-form diagrams, external documents and notes and not formal models. Even though diagrams used to describe NOV-1 contain no particular type of picture, **HOPEX** objects can be inserted so that repository features such as name consistency and hyperlinks are available.

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 Template", page 98
- ✓ "The NOV-1 High-Level Operational Concept Graphic Chapter", page 98
- ✓ "The NOV-1 Environments Dictionary Chapter", page 100

The NOV-1 Report Template

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.

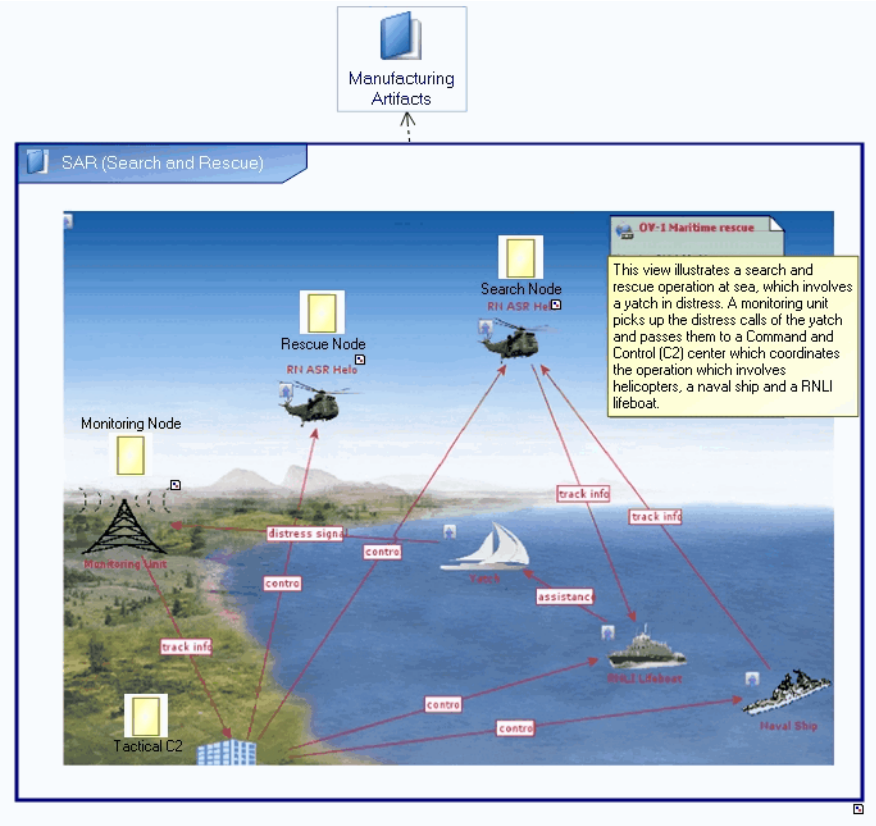
First, the chapter shows the "High-Level Operational Graphic" diagram, which gives a general understanding of the architecture.

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 or the NAF start page. The external references then appear under the architecture in the NAF navigation window and the NAF start page.*

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 which includes JPEG images and vector shapes. This is a **HOPEX** diagram so it can also include **HOPEX** objects allowing access to detailed definitions.



Example of an NOV-1 Library Diagram

The NOV-1 Environments Dictionary Chapter

Environments (conditions) are used to define certain properties (such as temperature, place, colour, etc.) that the objects to which they are linked, must respect.

This chapter lists the environments (conditions) of the architecture retrieved during the report generation.

A paragraph dedicated to each environment (condition) follows with the object defined by the condition. The properties of the environments (conditions) are presented in a table with their name, property type and property value, their unit and a comment.

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) as well as 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. NOV-2 does not depict the connectivity between nodes.

Creating Operational Nodes

Creating operational nodes is quite simple. This can be done from the navigation tree as well as from the **HOPEX NAF** start page.

To create an operational node from the start page:

1. In the **Architectures** frame of the start page, click the name of the architecture concerned.
2. In the **SubViews** frame, click **Operational Views > NOV-2**.
3. The operational nodes available for the operational view appear in the **Root Operational Nodes** tab in the right frame of the start page.
4. Click the **New** icon and in the dialog box that appears enter the name of the operational node.
5. Click **OK**.

The new operational node appears in the list in the **Root Operational Nodes** tab as well as in the NAF navigation tree along with the other operational nodes for the architecture (**NAF > Architectures > Architecture concerned > Operational Views > NOV-2 - Operational Nodes > ...**).

Operational nodes can also be initially created in the operational node structure diagram.

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 therefore shows how operational nodes and interactions interact with each other.

An Operational Node Structure Diagram actually details the structure of an operational node. The node is composed of sub-nodes that are connected to each other through interactions. 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. In this way you can differentiate the interactions performed in one node context from those performed in another node context. This principle is replicated in all structure diagrams and is further detailed in the **Artifact Type and Instance** technical article. 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 **Architectures** frame of the NAF start page, click the name of the architecture concerned.
2. In the **SubViews** frame, click **Operational Views > NOV-2**.
3. Select the **All Operational Nodes (EAF)** tab in the right frame of the start page.
4. Right-click the operational node concerned and select **New > Diagram.**
5. Select **Operational Node Structure Diagram** and click **Create**.
   The new diagram opens in the **HOPEX** desktop with a Root Operational Node positioned in it.

Adding operational nodes to the diagram

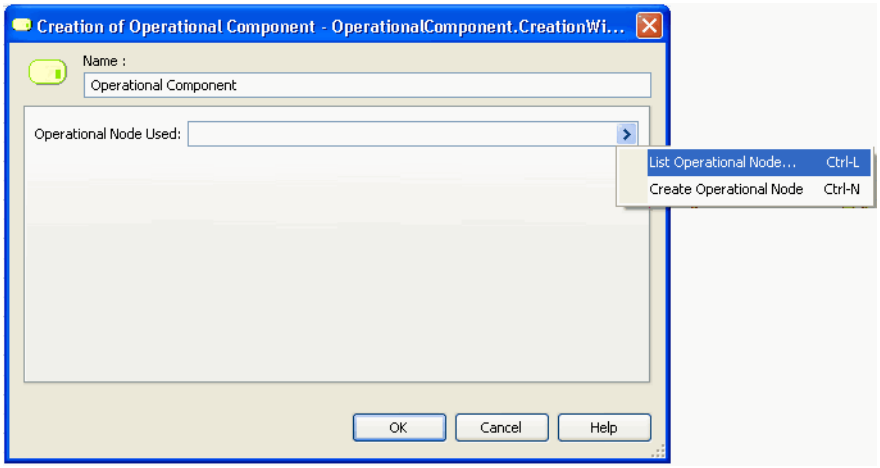
Operational nodes in the Structure diagram are referred to as operational components.

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 **Creation of Operational Component** dialog box that appears, if you so desire, enter the name of the operational node.

   ☛ *You do not have to enter a component name as the name of the operational node that is used for the component is automatically displayed in the structure diagram. The operational node name replaces the name entered in the Operational component name field.*

3. Click the arrow to the right of the "Operational Node Used" field and select **List Operational Nodes** to find an operational node that already exists.



☛ *If the operational node to be used does not exist, you can create it by selecting **Create Operational Node**.*

4. Click **OK**.
   The new operational node appears in the diagram.
5. 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 links two interacting items and is based on a definition (a service definition or a protocol): this definition can be detailed later on in the NSOV-2 view. The definition is used to accurately describe the information exchanged between the two nodes, the roles played by the two nodes (customer or provider) and the service and request points to which the interaction is connected. Service and request points are explained in the paragraph below.

All the structure diagrams (operational node, resource architecture or artifact assembly in NSV-1) are based on the same interaction principle. For more details on the interaction models, see the **Interaction Modeling** technical article..

To add an interaction to operational nodes:

1. In the object bar of the diagram, click the **Interaction** button
2. Click the first operational node to be connected and holding down the mouse button, drag the cursor to the second operational node and release the mouse button.

   ☛ *The line should always be drawn in the direction in which you want the interaction information to flow.*

3. In the **Add Interaction** dialog box that appears enter the interaction definition name or select an existing interaction definition from the list.

   ☛ *It is not necessary to enter an interaction name. If an interaction name is not entered, the interaction will automatically inherit the name of its definition/protocol.*

4. Click **OK**.
   A line representing the interaction appears in the diagram between the operational nodes.

The access points on interactions vary depending on particular situations.

- Turned out arrow: Consumer role is specified for the interaction
- Turned in arrow: Provider role is specified for the interaction
- Square: No role is specified
- Empty square or arrow: no access point is specified for the service
- Full square or arrow: an access point is specified for the service

Adding content to interactions

It is possible to add and display the content of interactions in the different structure diagrams. The information elements (contents) exchanged in an interaction are defined in a shared interaction protocol (service definitions). Content must be added in this definition and impacts all the interactions based on it.

To add content to an interaction:

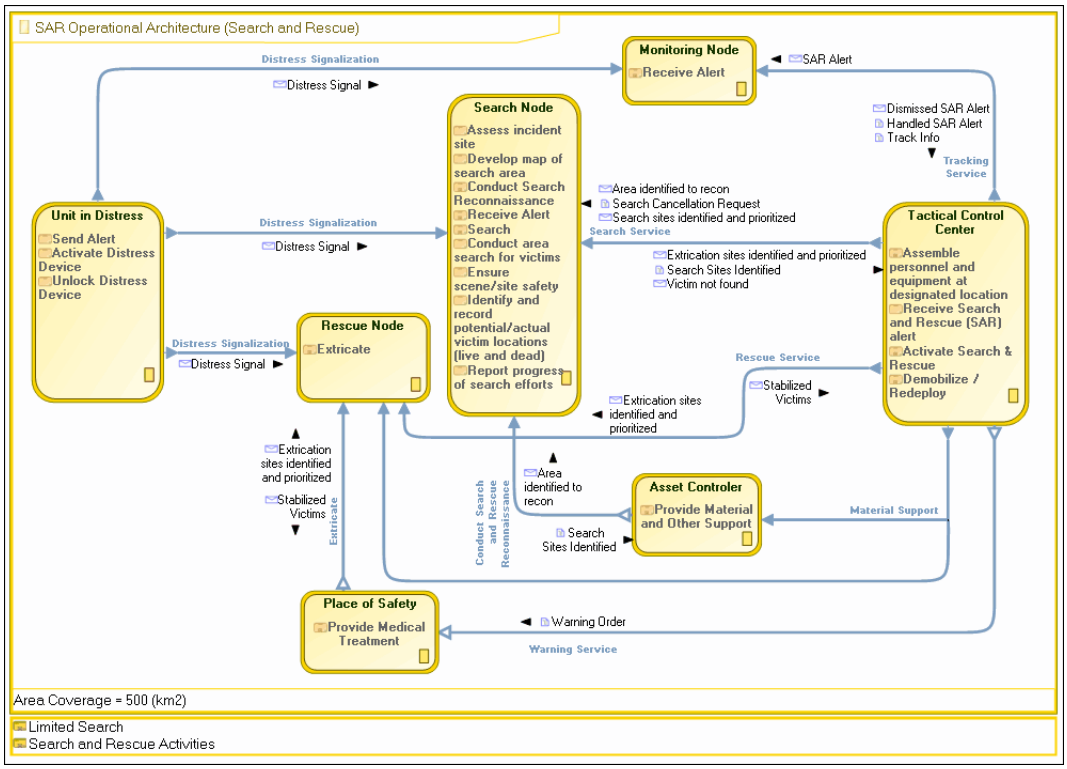
1. Right click the name of the interaction.
2. In the pop-up menu that appears select the service definition **> New > Message**.
3. Launch the query to find the Content to be attached.
4. Select your Content and click **OK**.
   The new content appears near the interaction.
   Black arrows indicate the direction flow of the content.

   ☛ *You are advised to insert the content in the service definition diagram if it exists. Doing so ensures that the graphical representation of the service definition is properly maintained.*

   ☛ *You can otherwise create the content in the Interaction Definition Diagram of the interaction concerned then add it to the structure diagram. You can access the Interaction Definition Diagram from the service definition pop-up menu.*

If you do not wish to display the message content in the diagram, in the **Views and Details** dialog box uncheck the **Interaction Upstream Contents** and **Interaction Downstream Contents** display options (right click the name of the interaction **> Shapes and Details**).

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

Adding Service and Request Points to 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 and request point. 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 on the other hand are used for interacting with the node that is considered to be the consumer in the interaction.

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.

For more details on service and request points, see the **Interaction Modeling** technical article.

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. Open the property page of the operational node that interests you.
2. Select the **Characteristics** tab and the **Realizations** subtab.
3. In the **Composite Realization** section, click **New**.
   The selection window opens.
4. Select "Business Capabilty realization" and click **OK**.
5. In the **Base type** field, select **Connect**.
6. Chose the capability implemented.
7. Click **OK.**
   The capabilty realization appears in the properties page.

The NOV-2 Report Template

NOV-2 reports are generated from the NOV-2 report template. This template is composed of eight chapters:

- The Operational Node Composition Hierarchy chapter
- The Operational Node Specialization Hierarchy chapter
- The Operational Node Generalization Hierarchy chapter
- The Operational Node Dictionary chapter
- The Operational Node Exchange chapter
- The Operational Node Exchange Balance chapter
- The Operational Node Exchange Compliance chapter
- The Operational Node Realization chapter

The NOV-2 report template is structured to analyze the architecture to which it is associated. However, two additional and optional parameters called Subsets can be used to restrict the scope of the report. These are the Operational Node Subset and the Resource Type Subset.

- **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 or 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 hierararchy 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

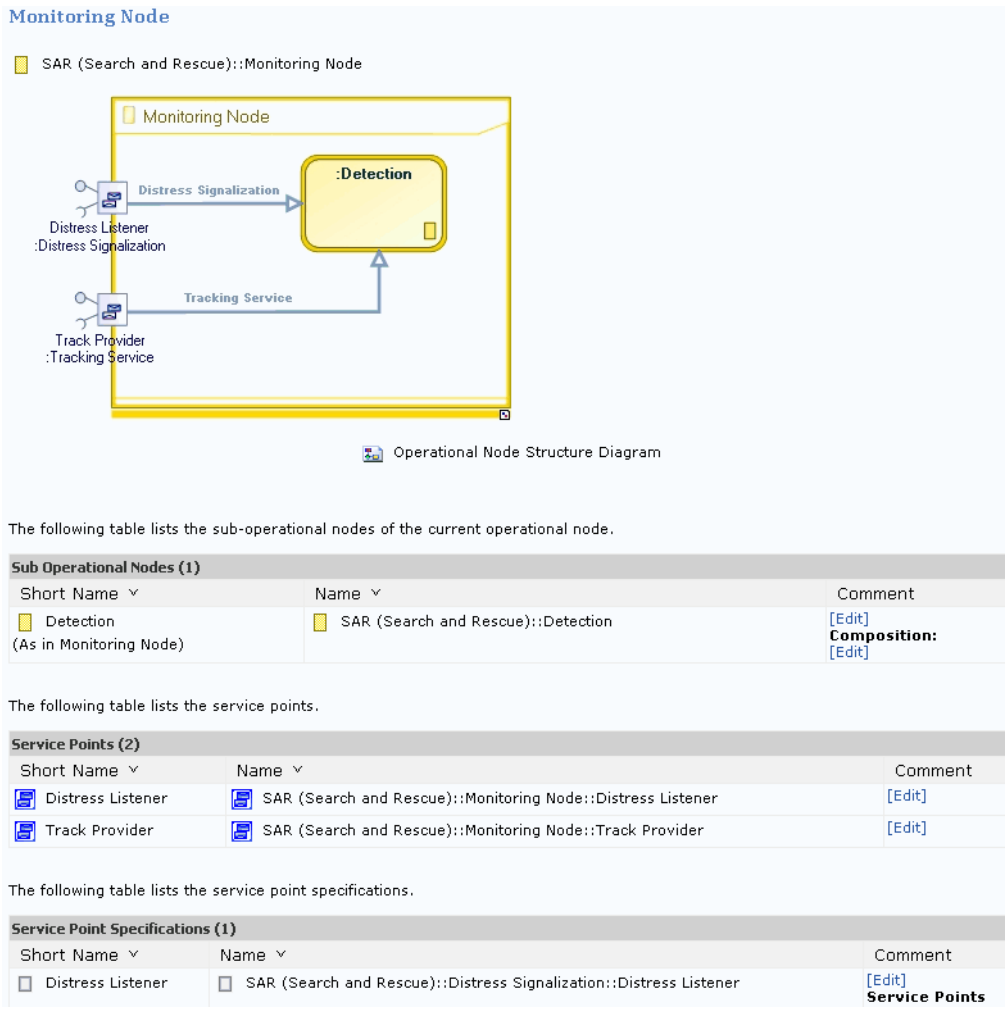
The NOV-2 operational node generalization hierararchy chapter shows how the operational nodes of the architecture are generalized.

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.

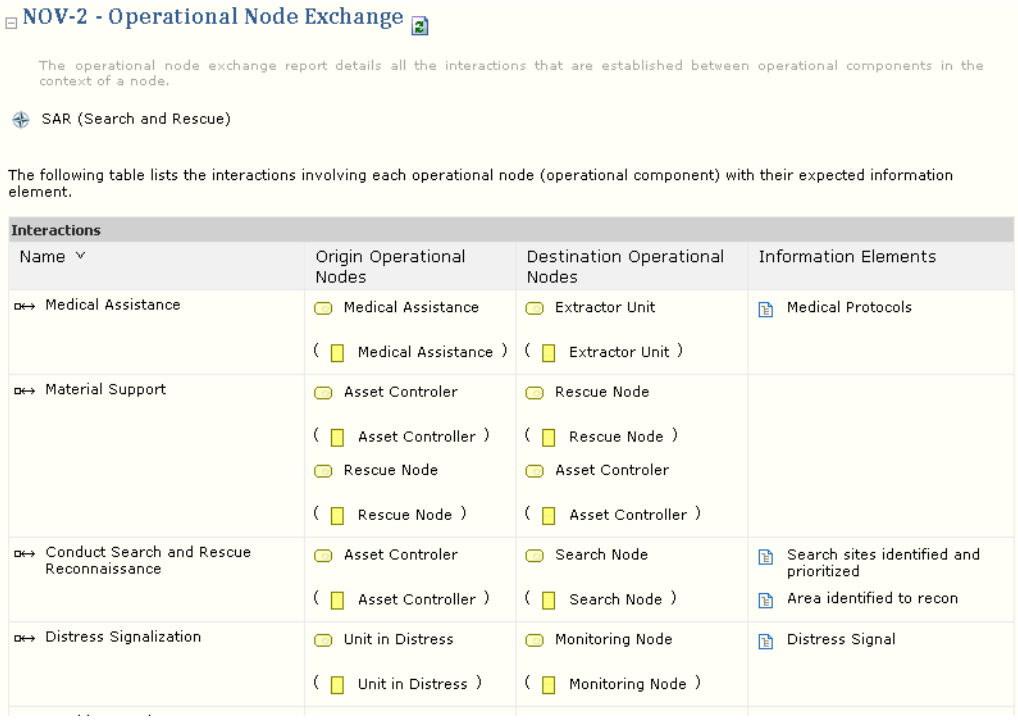


Example of Operational Node Dictionary Chapter with Diagram

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 also displayed.



NOV-2 - Operational Node Exchange

The operational node exchange report details all the interactions that are established between operational components in the context of a node.

SAR (Search and Rescue)

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
Medical Assistance	Medical Assistance (Medical Assistance)	Extractor Unit (Extractor Unit)	Medical Protocols
Material Support	Asset Controler (Asset Controller) Rescue Node (Rescue Node)	Rescue Node (Rescue Node) Asset Controler (Asset Controller)	
Conduct Search and Rescue Reconnaissance	Asset Controler (Asset Controller)	Search Node (Search Node)	Search sites identified and prioritized Area identified to recon
Distress Signalization	Unit in Distress (Unit in Distress)	Monitoring Node (Monitoring Node)	Distress Signal

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

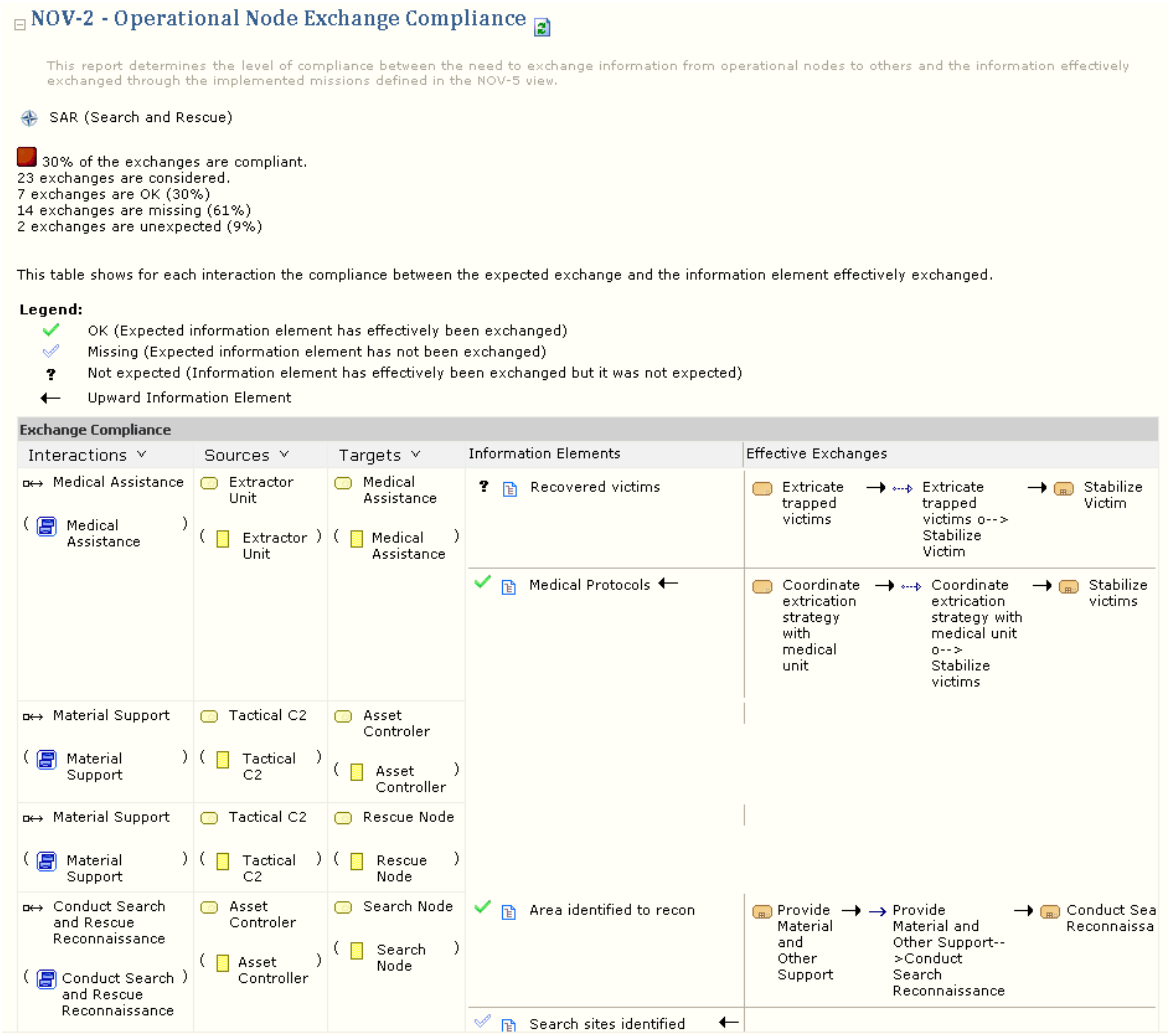
A node structure does not indicate how missions are performed. It only describes the nodes involved in the mission and how they are structured. Through interactions, you can guess the potential information exchanges that occur between nodes. To add a dynamic perspective to nodes, a process must be described (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.

The figure below is an example of an NOV-2 compliance chapter. The table displays expected and correct content.



NOV-2 - Operational Node Exchange Compliance

This report determines the level of compliance between the need to exchange information from operational nodes to others and the information effectively exchanged through the implemented missions defined in the NOV-5 view.

SAR (Search and Rescue)

30% of the exchanges are compliant.
23 exchanges are considered.
7 exchanges are OK (30%)
14 exchanges are missing (61%)
2 exchanges are unexpected (9%)

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

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

Exchange Compliance

Interactions	Sources	Targets	Information Elements	Effective Exchanges
Medical Assistance / (Medical Assistance)	Extractor Unit / (Extractor Unit)	Medical Assistance / (Medical Assistance)	Recovered victims	Extricate trapped victims → Extricate trapped victims o--> Stabilize Victim → Stabilize Victim
			Medical Protocols ←	Coordinate extrication strategy with medical unit → Coordinate extrication strategy with medical unit o--> Stabilize victims → Stabilize victims
Material Support / (Material Support)	Tactical C2 / (Tactical C2)	Asset Controler / (Asset Controller)		
Material Support / (Material Support)	Tactical C2 / (Tactical C2)	Rescue Node / (Rescue Node)		
Conduct Search and Rescue Reconnaissance / (Conduct Search and Rescue Reconnaissance)	Asset Controler / (Asset Controller)	Search Node / (Search Node)	Area identified to recon	Provide Material and Other Support → Provide Material and Other Support-->Conduct Search Reconnaissance → Conduct Sea Reconnaissa
			Search sites identified ←	

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.

In the matrix, the resource items are displayed in columns while the operational nodes are displayed 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 and identifies who exchanges information, what information is exchanged, why the information is necessary, and 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", page 112
✓ "The NOV-3 Operational Information Exchange Dictionary", page 112
✓ "The NOV-3 Contents Linked to Operational Items Chapter", page 113
✓ "The NOV-3 Operational Information Exchange Matrix Chapter", page 114

The NOV-3 Report Template

The report template used to generate tne 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. In this case, the NOV-3 report focuses on this set of 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 parameter is used to specify the interactions used for the exchanges, that should be included in the analysis. If no exchange property is set, all the exchanges are taken into account.

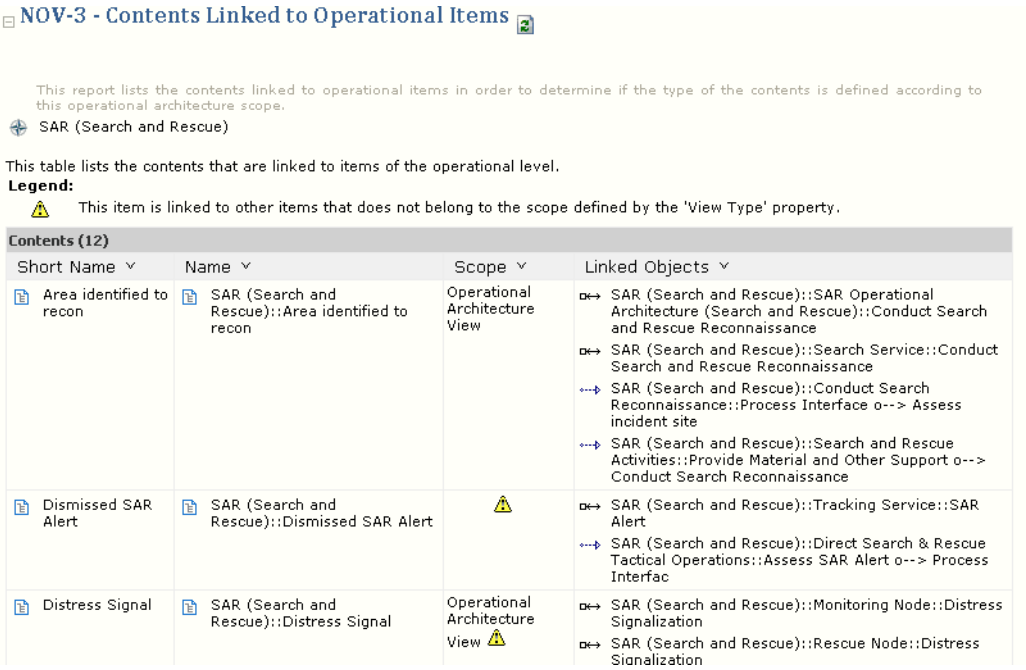
The NOV-3 Operational Information Exchange Dictionary

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.



NOV-3 - Contents Linked to Operational Items

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

SAR (Search and Rescue)

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

Legend:

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

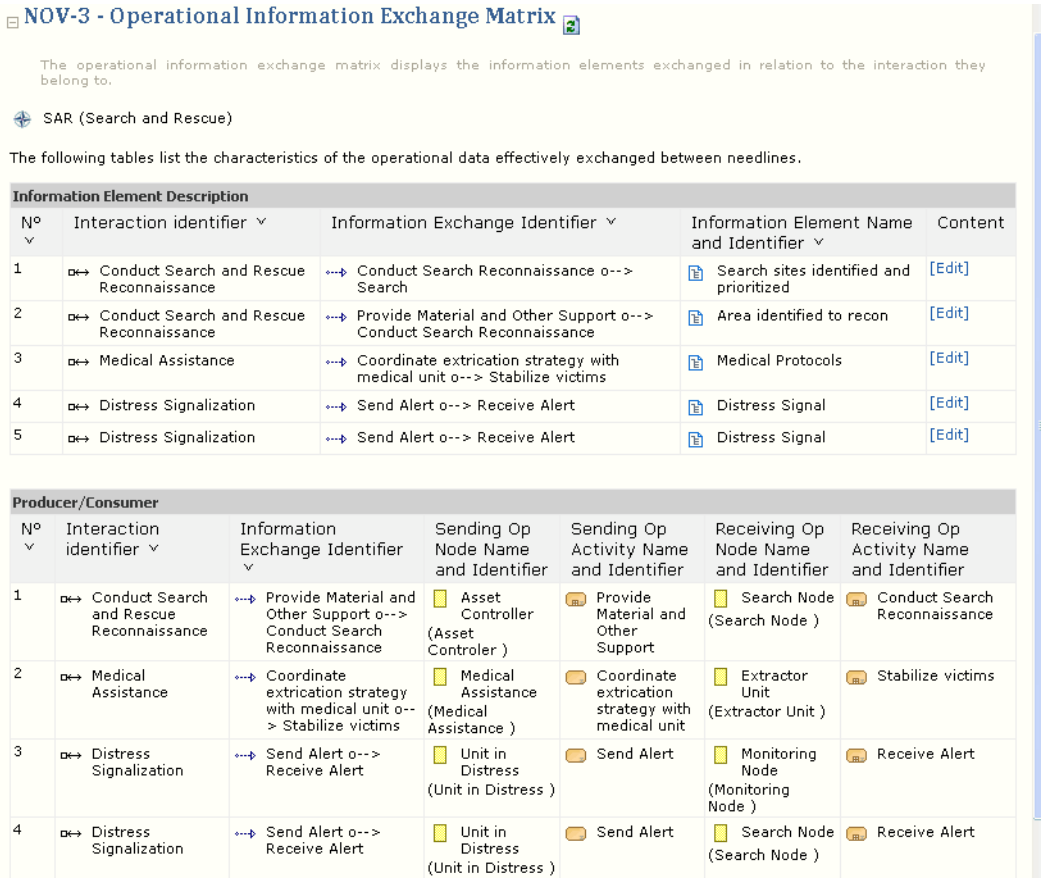
Contents (12)

Short Name	Name	Scope	Linked Objects
Area identified to recon	SAR (Search and Rescue)::Area identified to recon	Operational Architecture View	SAR (Search and Rescue)::SAR Operational Architecture (Search and Rescue)::Conduct Search and Rescue Reconnaissance SAR (Search and Rescue)::Search Service::Conduct Search and Rescue Reconnaissance SAR (Search and Rescue)::Conduct Search Reconnaissance::Process Interface o--> Assess incident site SAR (Search and Rescue)::Search and Rescue Activities::Provide Material and Other Support o--> Conduct Search Reconnaissance
Dismissed SAR Alert	SAR (Search and Rescue)::Dismissed SAR Alert	⚠	SAR (Search and Rescue)::Tracking Service::SAR Alert SAR (Search and Rescue)::Direct Search & Rescue Tactical Operations::Assess SAR Alert o--> Process Interfac
Distress Signal	SAR (Search and Rescue)::Distress Signal	Operational Architecture View ⚠	SAR (Search and Rescue)::Monitoring Node::Distress Signalization SAR (Search and Rescue)::Rescue Node::Distress Signalization

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.



NOV-3 - Operational Information Exchange Matrix

The operational information exchange matrix displays the information elements exchanged in relation to the interaction they belong to.

SAR (Search and Rescue)

The following tables list the characteristics of the operational data effectively exchanged between needlines.

Information Element Description

N°	Interaction identifier	Information Exchange Identifier	Information Element Name and Identifier	Content
1	Conduct Search and Rescue Reconnaissance	Conduct Search Reconnaissance o--> Search	Search sites identified and prioritized	[Edit]
2	Conduct Search and Rescue Reconnaissance	Provide Material and Other Support o--> Conduct Search Reconnaissance	Area identified to recon	[Edit]
3	Medical Assistance	Coordinate extrication strategy with medical unit o--> Stabilize victims	Medical Protocols	[Edit]
4	Distress Signalization	Send Alert o--> Receive Alert	Distress Signal	[Edit]
5	Distress Signalization	Send Alert o--> Receive Alert	Distress Signal	[Edit]

Producer/Consumer

N°	Interaction identifier	Information Exchange Identifier	Sending Op Node Name and Identifier	Sending Op Activity Name and Identifier	Receiving Op Node Name and Identifier	Receiving Op Activity Name and Identifier
1	Conduct Search and Rescue Reconnaissance	Provide Material and Other Support o--> Conduct Search Reconnaissance	Asset Controller (Asset Controler)	Provide Material and Other Support	Search Node (Search Node)	Conduct Search Reconnaissance
2	Medical Assistance	Coordinate extrication strategy with medical unit o--> Stabilize victims	Medical Assistance (Medical Assistance)	Coordinate extrication strategy with medical unit	Extractor Unit (Extractor Unit)	Stabilize victims
3	Distress Signalization	Send Alert o--> Receive Alert	Unit in Distress (Unit in Distress)	Send Alert	Monitoring Node (Monitoring Node)	Receive Alert
4	Distress Signalization	Send Alert o--> Receive Alert	Unit in Distress (Unit in Distress)	Send Alert	Search Node (Search Node)	Receive Alert

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 start page or 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:

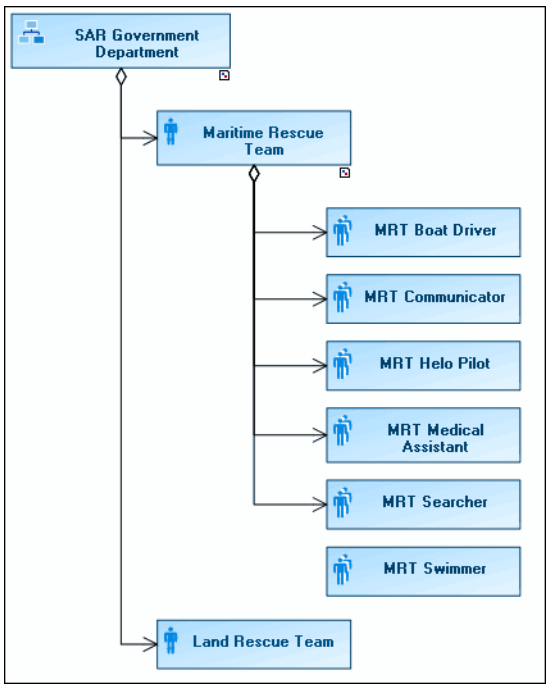
1. Expand the folders **Architectures > Architecture concerned > Operational Views > NOV-4 - Organizational Relationship Chart**.
2. Right-click the **Root Org-Units** folder and select **New > Org-Unit**.
3. In the dialog-box that appears, enter the name of the Org-unit and click **OK**.
   The new Org-unit appears in the list of objects in the Root Org-Units folder. You can then specify what type of org-unit this is.

To specify the org-unit type:

1. Right-click the org-unit and select **Properties**.

2. In the **Characteristics** tab, select one of the types from the Org-Unit Type drop-down list (Company, Institute, Structure, Function, Generic, Manager).

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. Otherwise, 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.

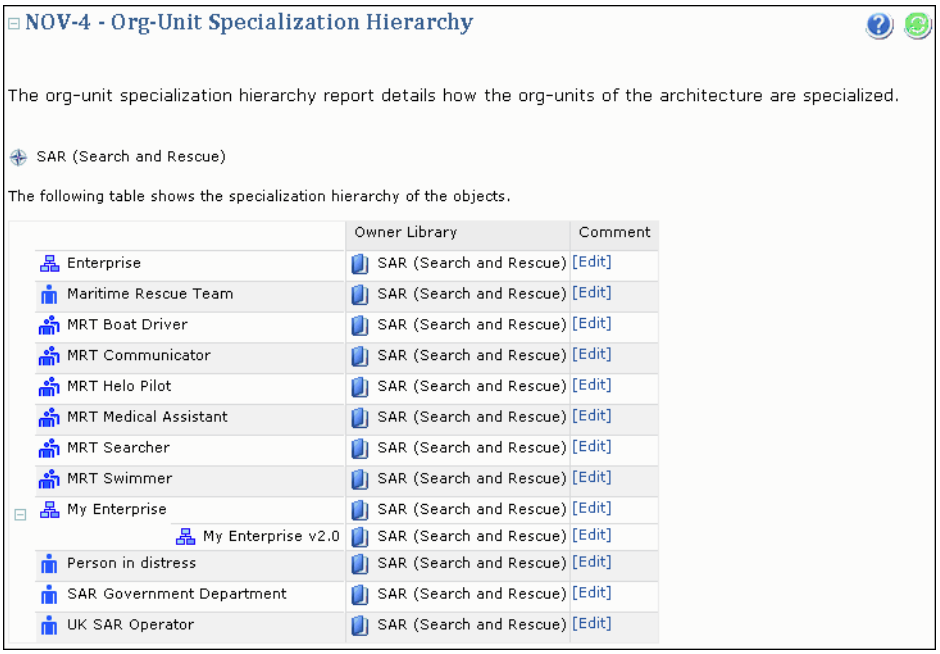
The NOV-4 Org-Unit Specialization Hierarchy Chapter

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 and 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.

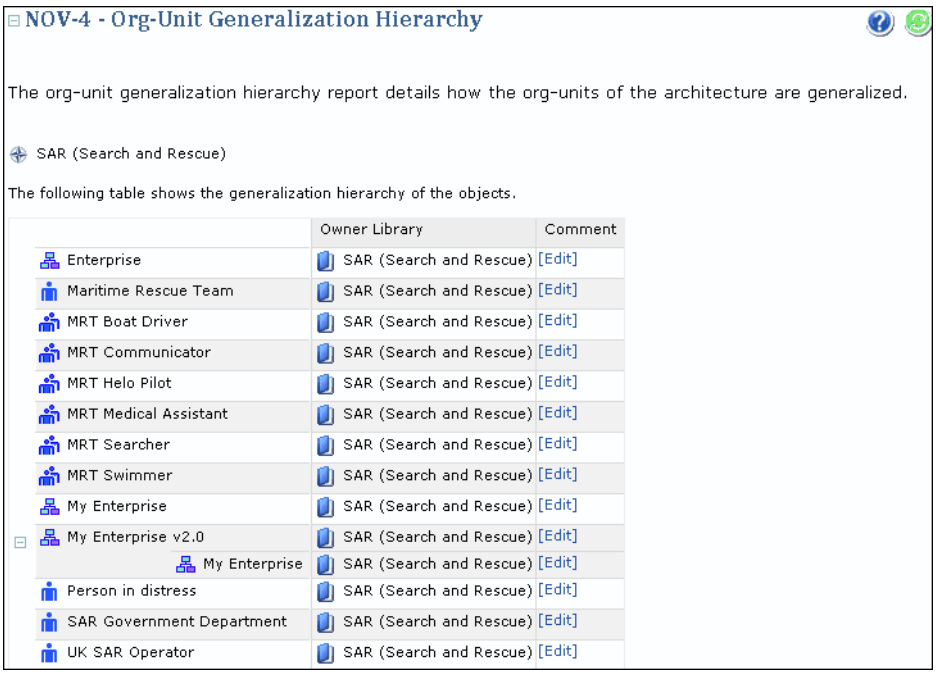
	Owner Library	Comment
Enterprise	SAR (Search and Rescue)	[Edit]
Maritime Rescue Team	SAR (Search and Rescue)	[Edit]
MRT Boat Driver	SAR (Search and Rescue)	[Edit]
MRT Communicator	SAR (Search and Rescue)	[Edit]
MRT Helo Pilot	SAR (Search and Rescue)	[Edit]
MRT Medical Assistant	SAR (Search and Rescue)	[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 simply creates an alphabetical list of all org-units mentioned in the Org-Unit Subset parameter or of all the 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.

Tree diagrams and structure 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 also 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 tab.

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.

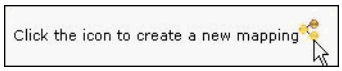
The list of Organizational Processes is found in the NOV-6c subview.

The NOV-4 Org-Unit 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.

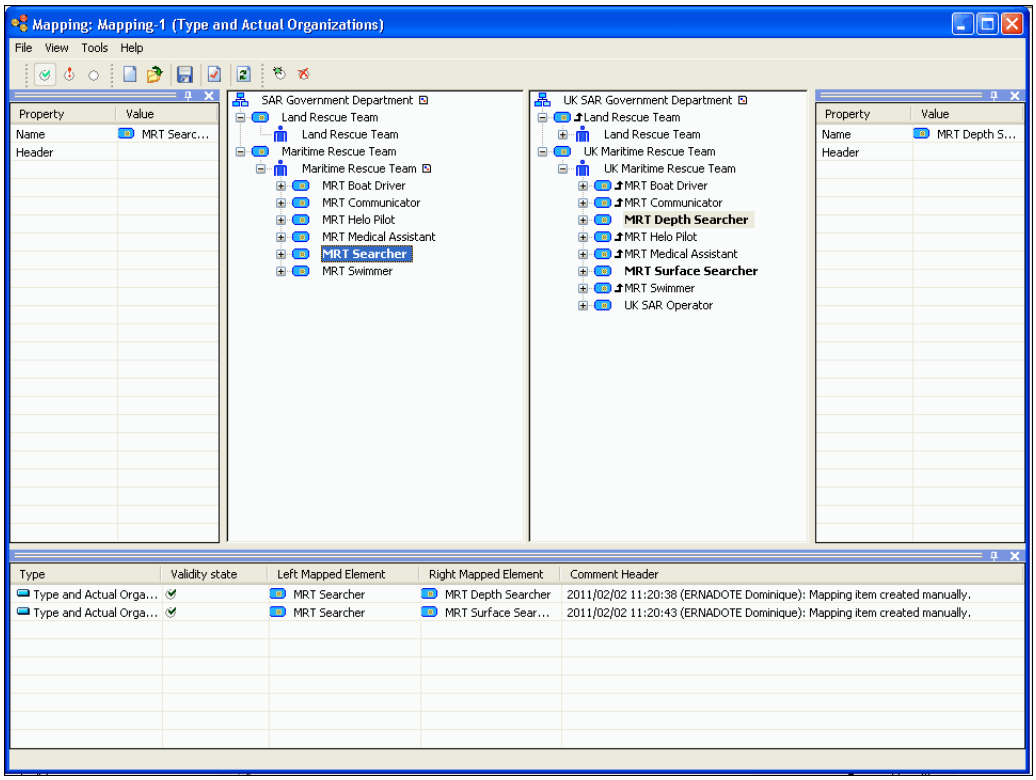
This chapter gives an overview of the mapping levels that exist between org-unit types and the actual organizational resources (org-units). It also gives the percentage of mapping compliance between the types and actual org-units.

Mappings can be created in the report chapter.



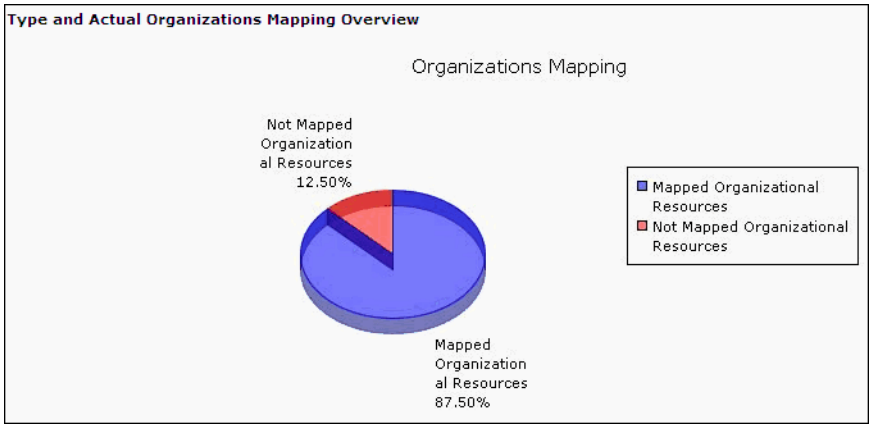
In the figure below, the two middle columns contain the generic and the actual org-units: "types" to left and "actual" to the right. The generic "SAR Government Department" organization is mapped to the "UK SAR Government Department" (mapping not indicated) and the "MRT Searcher" org-unit component type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

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



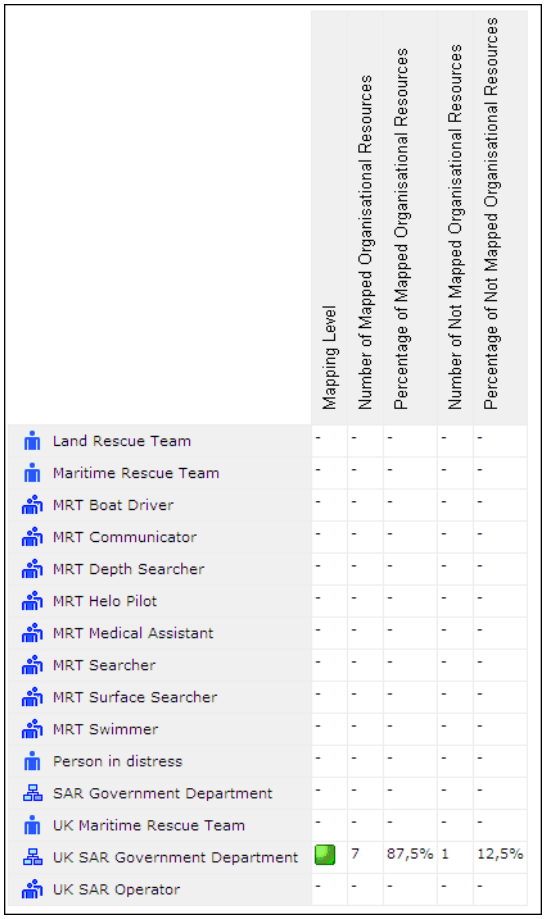
☛ *To verify if a mapping exists for an org-unit type, right click the type and click **Locate**. If no mapping exists, the **Locate** command is grayed. For more details on mapping objects, see the **HOPEX IT Architecture** user guide, "Application Deployment" chapter.*

The Org-Unit Type and Actual Organizations Mapping Overview chapter begins with a pie chart that indicates the percentage of mapped and unmapped organizational resources.



This pie chart is then followed by a list of the organizational resources of the architecture and 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.

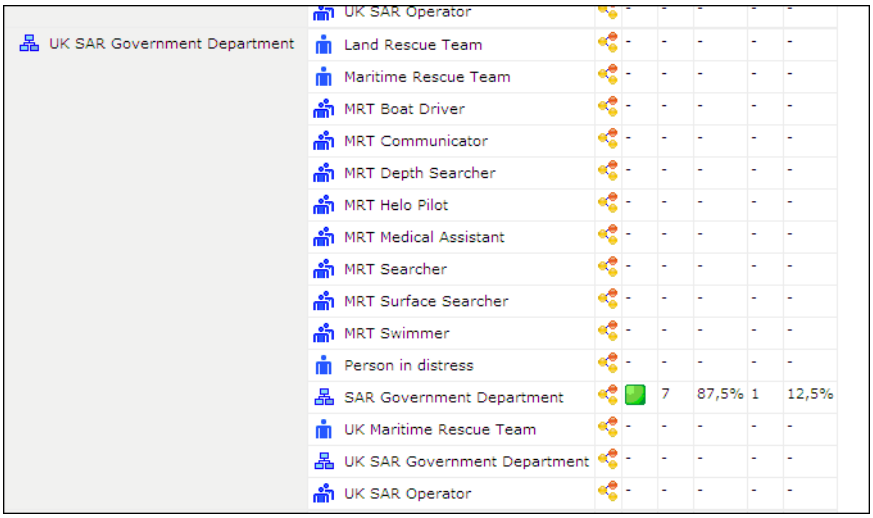


	Mapping Level	Number of Mapped Organisational Resources	Percentage of Mapped Organisational Resources	Number of Not Mapped Organisational Resources	Percentage of Not Mapped Organisational Resources
Land Rescue Team	-	-	-	-	-
Maritime Rescue Team	-	-	-	-	-
MRT Boat Driver	-	-	-	-	-
MRT Communicator	-	-	-	-	-
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 Org-Unit 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.

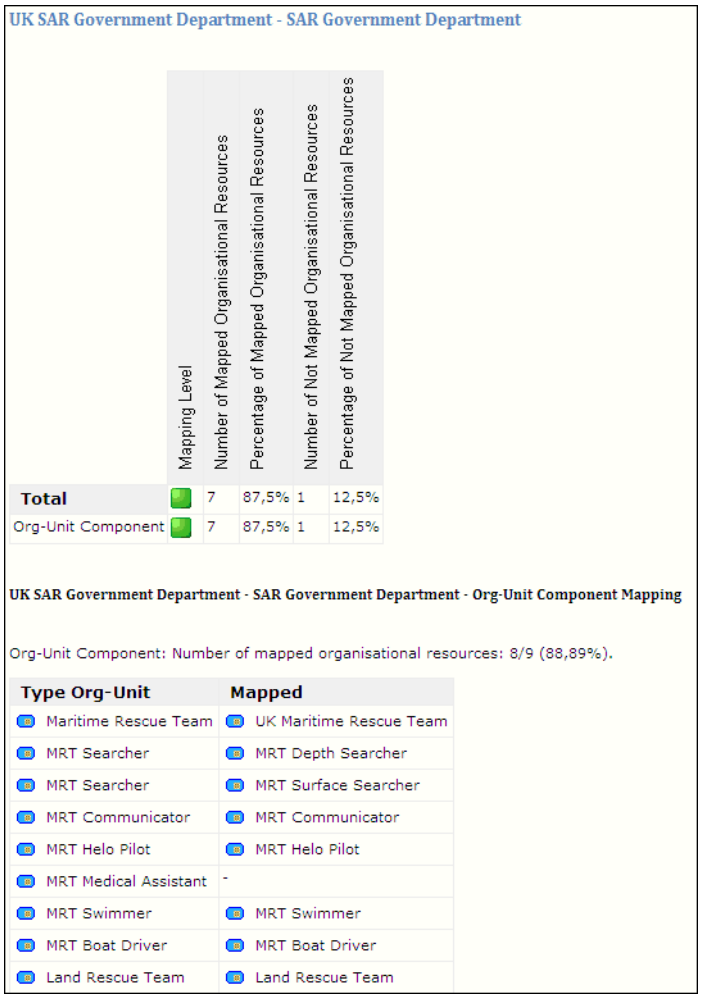


The NOV-4 Org-Unit Type and Actual Organizations Mapping Deep Details Chapter

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

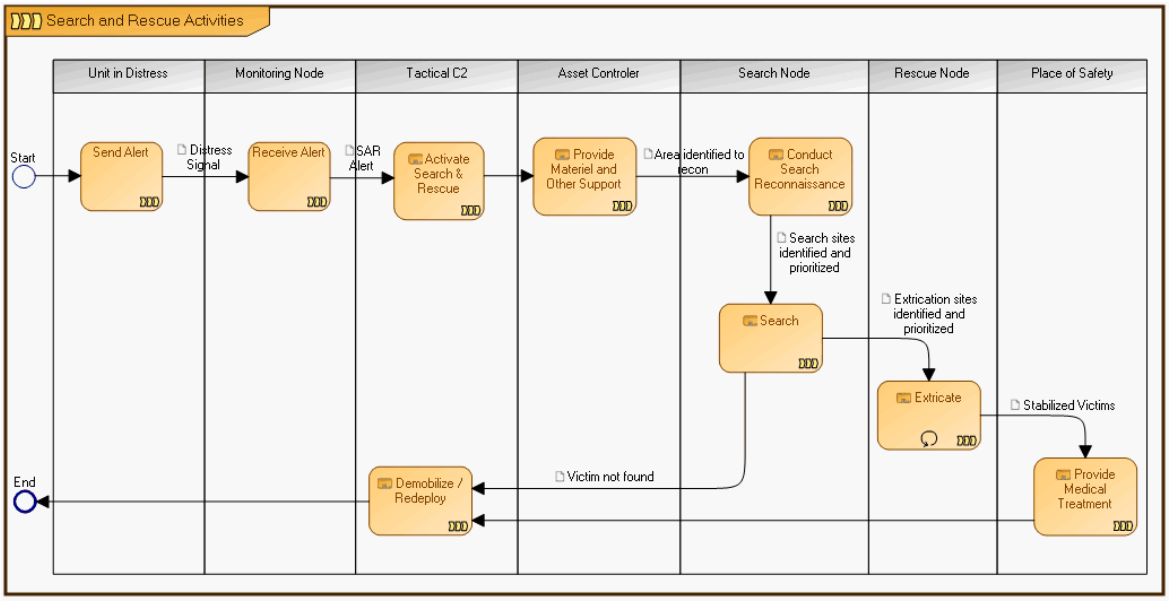
Creating Activity Models

The entry point for the operational activity models is the functional process concept. Functional processes describe a sequence of transformation activities. 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.

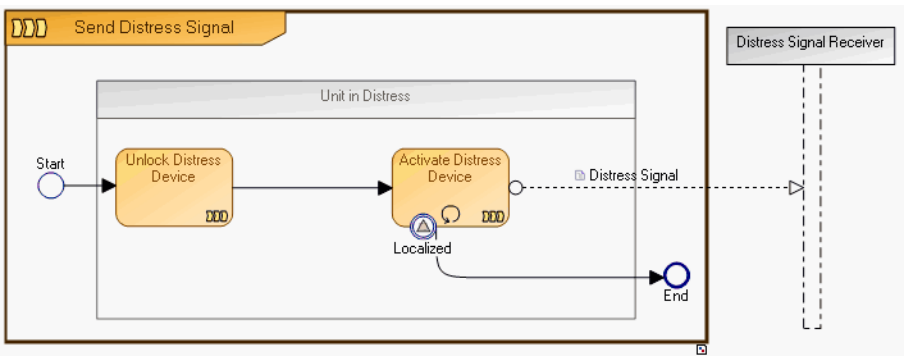
The figure below is an example of a Functional Process Diagram. It displays the Search process of the architecture. In this figure the process contains activities that are assigned to different operational nodes.

☛ *Note that activities can sometimes call different processes. See the **HOPEX Business Process Analysis** user guide 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 window, expand the **NOV-5 Operational Activity Model** and **Root Functional Pocesses** folders.

2. Right-click the functional process for which you wish to create the diagram and select **New > Diagram > Functional Process Diagram > Create**.
   A diagram opens in the desktop, on the right with a frame representing the functional process to be described.

Participants

As you will have noticed, you can place your participants inside as well as outisde 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.

Adding participants to the Functional Process diagram

Before you place a participant in a diagram, you must first create the paticipant and assign it to an operational node or operational component. When this is done, only the assigned component or node can be attached to this participant in the diagram.

It is more accurate to assign an operational component than an operational node to a participant. This is so as operational components are already defined in particular contexts.

Activities

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

To add an activity:

1. In the objects bar select the **Functional Activity** icon and click in the diagram, on the participant responsible for the activity.
2. Enter the name of the functional process and click **OK**.
   The new activity appears in the diagram.

> ☛ *If the activity represents another functional process, you can call this process from within the **Properties** dialog box, **Characteristics** tab, "Called Functional Process" box. Note the shape difference between activities with and without called processes.*

You can describe your activities in detail through Activity Decomposition diagrams.

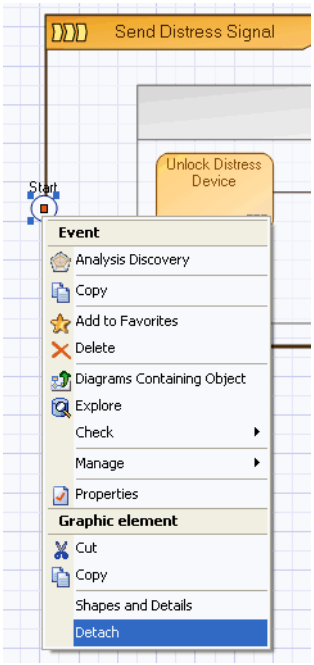
Events

When you add events to your diagram (Start End, etc.), you may choose to place them in the process or on the frame of the Process described by the diagram: this can be done through a drag and drop motion. If you place the event on the frame, a red rectangle appears in the event.

> ☛ *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:

- Right-click the event and in its contextual menu, select **Detach**.



- You can then drag and drop the event into the frame.

Message

Messages are used to pass on information from one element to another. This exchange of information in the process diagram is displayed in message flows through the use of the message flow icon.

Sequence

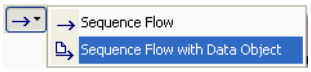
The activities in a process are not all carried out simultaneously. Often the execution or completion of one activity triggers or leads to another activity. Sequence flows are therefore used to indicate the steps to be followed (sequence of activities) in a process. The sequence flows of activities are indicated through the use of the sequence flow icon.

Messages are also sent from one activity to another and, as mentioned before, are represented through the use of message flows.

If you so desire, you can combine the sequence of activities with the exchange of information between these activities. This can be done through the use of the "Sequence Flow with Data Object" icon.

To create a sequence flow with message 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 Interesting 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.
- 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 that are 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 two parameters: the architecture and the activity.

- 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. It can be set with either just a few

activities or a few functional processes. If this parameter 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 hierararchy 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 begins with a list of the all the functional processes in the architecture.

A detailed paragraph for each functional process follows with the describing Functional process diagram and a list of the involved activities. Sub processes are also listed if they exist.

A list of all the activities in the architecture then follows with paragraphs to describe each activity.

The content of each paragraph varies depending on the type of item being described. Generally, functional processes have diagrams that describe them. Activities may also have sub-activities. The paragraphs relating to activities display the describing diagrams and list the contained activities with their name, comment and describing diagrams. On the other hand, if the item is a functional process, a paragraph is generated if the functional process has at least one describing diagram or if it is composed of activities. This paragraph will display the describing diagrams and lists the contained activities.

The called functional processes of activities are also listed.

The NOV-5 Operational Activity Exchange Chapter

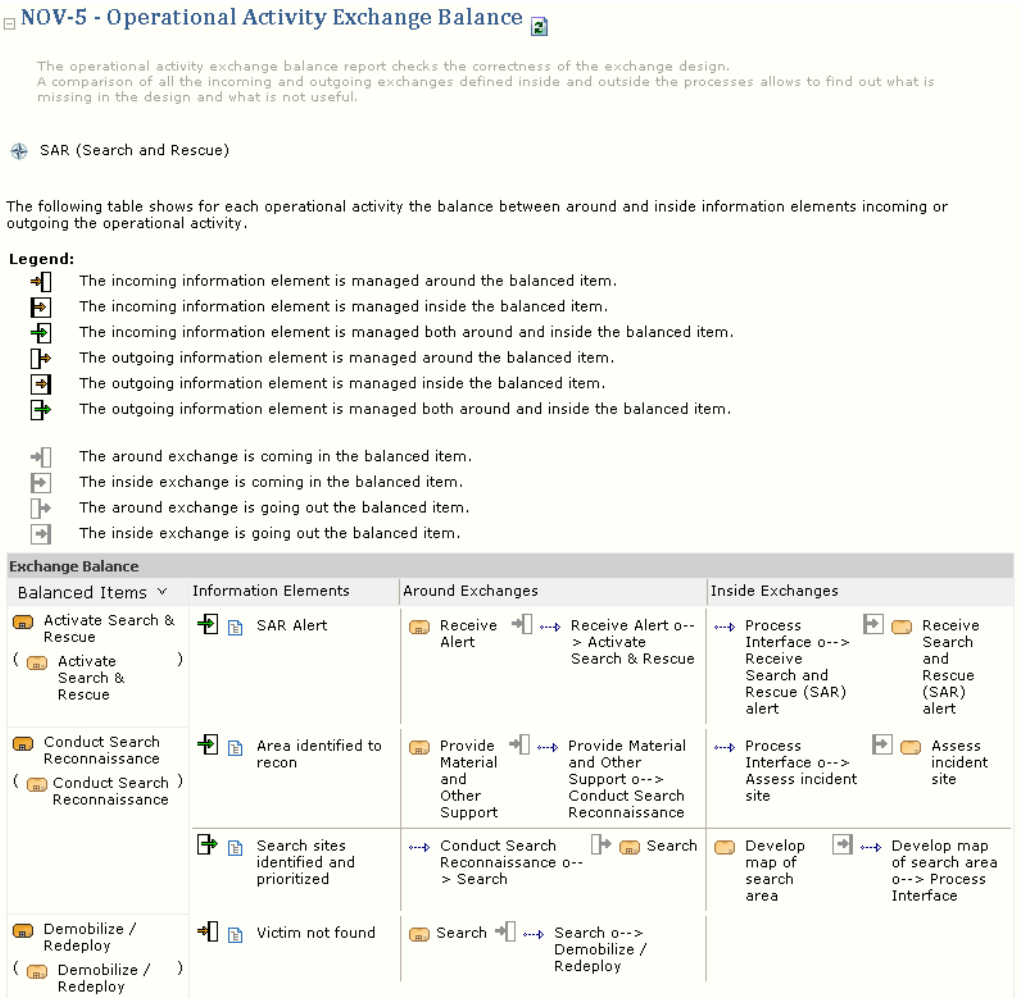
This chapter shows the diagrams that display information exchanges between functional processes and activities. It also details the information exchange between the selected activities. All messages or interactions connecting activities are displayed in a list with their name, comment and source and target activities.

The NOV-5 Operational Activity Exchange Balance Chapter

This chapter helps to determine the compliance of the exchanges defined within functional processes and the exchanges received and/or sent by these functional processes. The chapter 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 to 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 and start page. 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: 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 and the NOV-6a Rules Linked to Operational Items chapter.

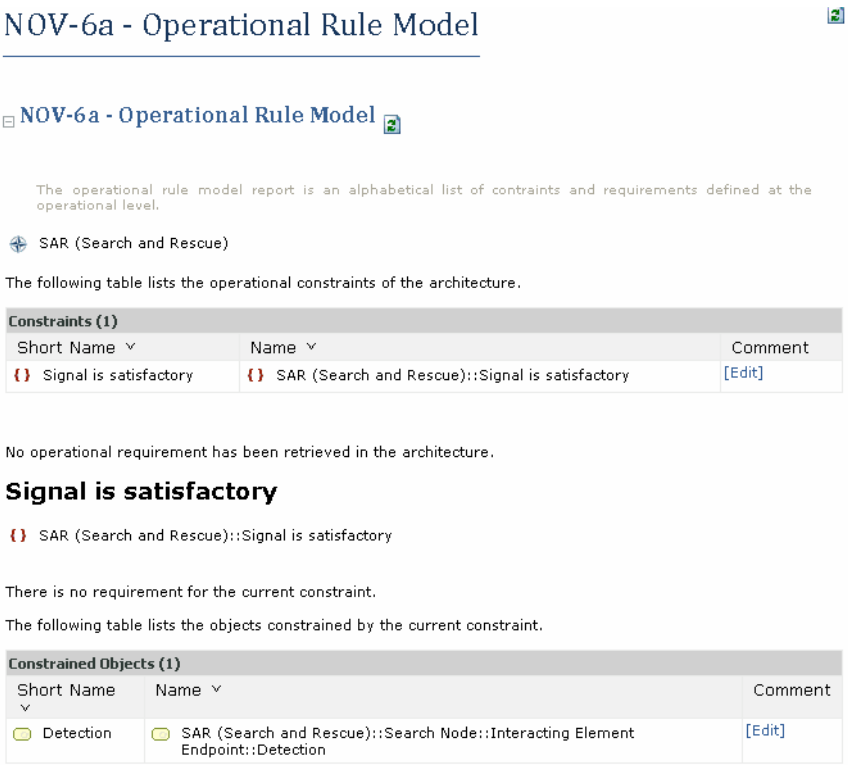
The NOV-6a Operational Rule Model Chapter

This chapter lists the constraints and requirements attached to the operational items of the architecture. The constraints are the external constraining elements that set the terms for whatever solutions are implemented while the 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. The constraints are then retrieved in the Operational Constraints folder

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



NOV-6a - Operational Rule Model

NOV-6a - Operational Rule Model

The operational rule model report is an alphabetical list of contraints and requirements defined at the operational level.

SAR (Search and Rescue)

The following table lists the operational constraints of the architecture.

Constraints (1)		
Short Name	Name	Comment
Signal is satisfactory	SAR (Search and Rescue)::Signal is satisfactory	[Edit]

No operational requirement has been retrieved in the architecture.

Signal is satisfactory

SAR (Search and Rescue)::Signal is satisfactory

There is no requirement for the current constraint.

The following table lists the objects constrained by the current constraint.

Constrained Objects (1)		
Short Name	Name	Comment
Detection	SAR (Search and Rescue)::Search Node::Interacting Element Endpoint::Detection	[Edit]

Example of the NOV-6a Operational Rule Model Chapter

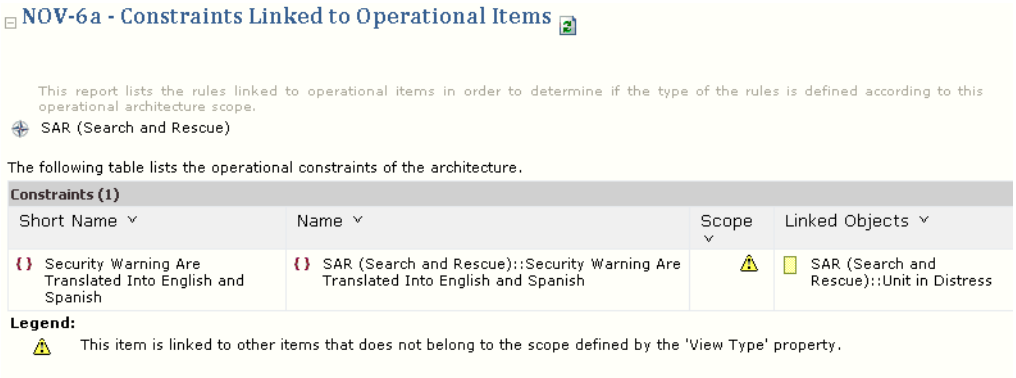
The NOV-6a Rules Linked to Operational Items Chapter

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

This chapter displays a table with an alphabetical list of the constraints that are linked to operational items. 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.

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. The "View Type" property for constraints with a warning can, however, be changed from the table (right click the

constraint **> Properties > Model of Data Exchanged > NAF Architecture Type >** etc.). Refresh the report to show the changes.



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

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.

The **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.*

Creating State Machines

Two methods exist for creating state machines: from an activity or from the navigation tree. For a state machine to be set in the operational range, it must meet one of the following constraints: be connected to an activity or 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.

Once a state machine has been created it can be described using state diagrams. These diagrams contain events and successions from event to event.

To create an operational architecture state machine:

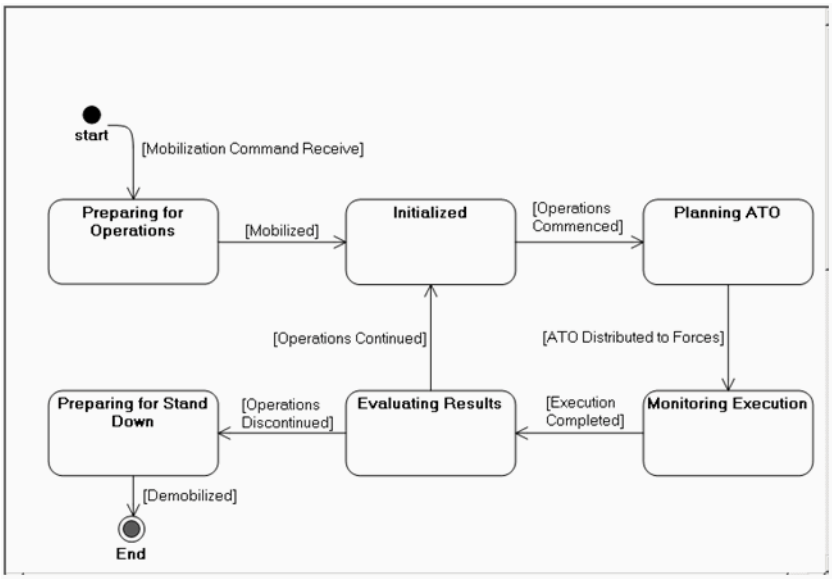
1. In the NAF navigation tree, expand the **NOV-6b - Operational State transition Description** folder.
2. Right click **Operational State Machines** folder and select **New > State Machine**.

3. In the dialog box that appears, enter the name of the state machine and click **OK**.
   The new state machine is created in the **Operational State Machines** folder.

You can change the property of the behavior in the **Properties** page of the folder in the NAF State Level tab.

If you select "All Views", the behavior also appears in the NSV-10b folder.

The figure below is an illustration of a state diagram.



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

An event relates to a point in time while the succession matches what happens between two events. To create a state diagram the user must remember that each arrow corresponds to a state and each graphical circle corresponds to the transition.

If a state can be reached from two or more previous states, then a synchronization object must be used.

If a state can reach two or more following states then a decision object must be used (for example the **Execution Completed** state in the previous figure).

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

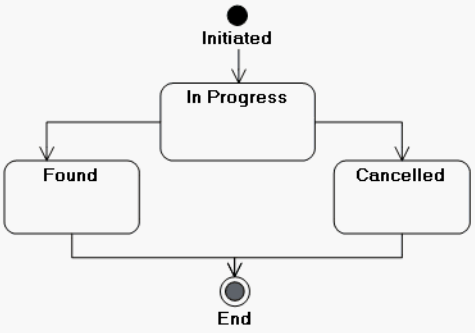
This chapter lists all the state machines attached to the selected activities. The report starts with a simple table of the state machines (short name, long name, comment) and the linked activities.

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 and the (commented) describing diagrams. It also lists the events (states) with their name and comment

in a table. The successions (transitions) are listed 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.

The **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. This description is done in two ways: either by splitting the organizational process into smaller parts (sub-organizational processes), or by describing how the organizational process is actually performed by the different participants. The participants themselves are designed within the context of the organizational process and are later associated to org-units of the organization. The Organizational Process Diagram is used to detail how the organizational process is performed.

The NOV-6c Report Template

The NOV-6c Report Template supplies reports to detail business 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 proces 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.

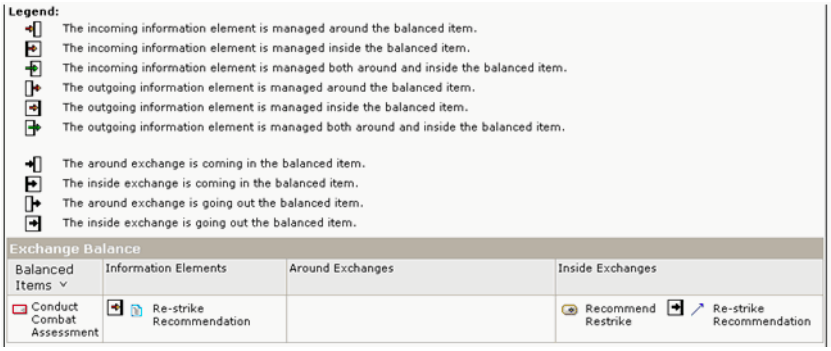
Paragraphs are similarly generated for each interaction scenario.

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.

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

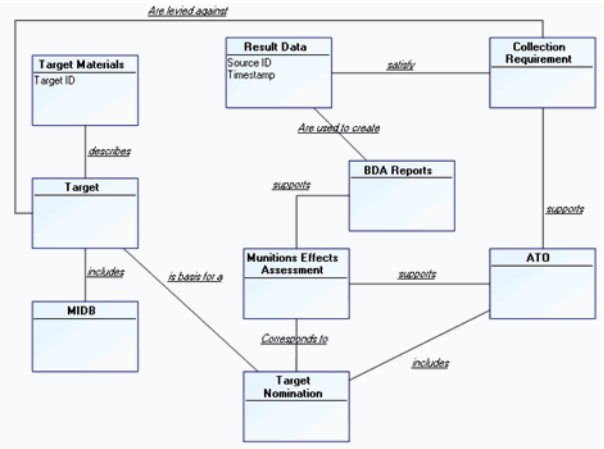
NOV-7 INFORMATION MODEL

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

The **HOPEX NAF** can be used to design data models. The data model is a concept that can be graphically designed in Data model diagrams. The user creates new data models from the NAF start page or 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 Model

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.

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

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 the:

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.

The different services are classified as operations, application or information services.

With the **HOPEX NAF** product, the user can create services with categories. They 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.

Creating a Service

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

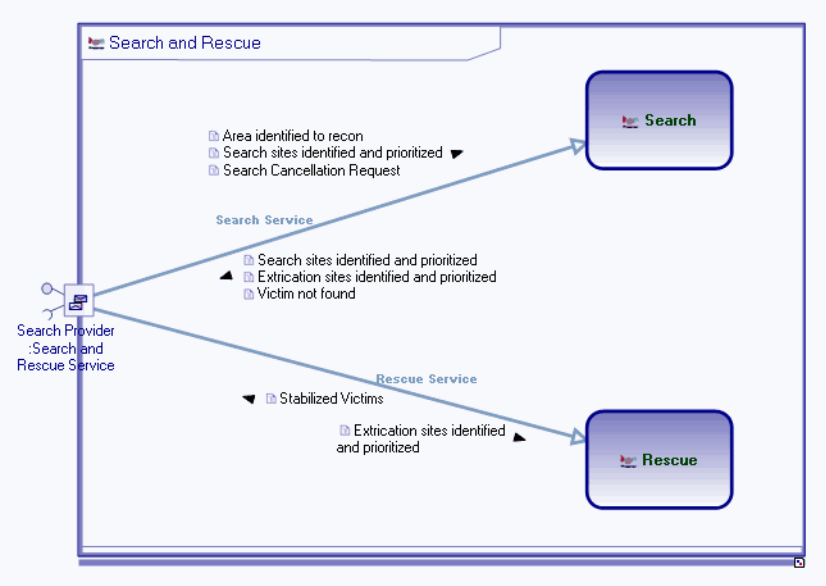
To create a service:

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" and "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.

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.



Example of a Service Structure Diagram.

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

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

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 NCV-7 and NSV-12.

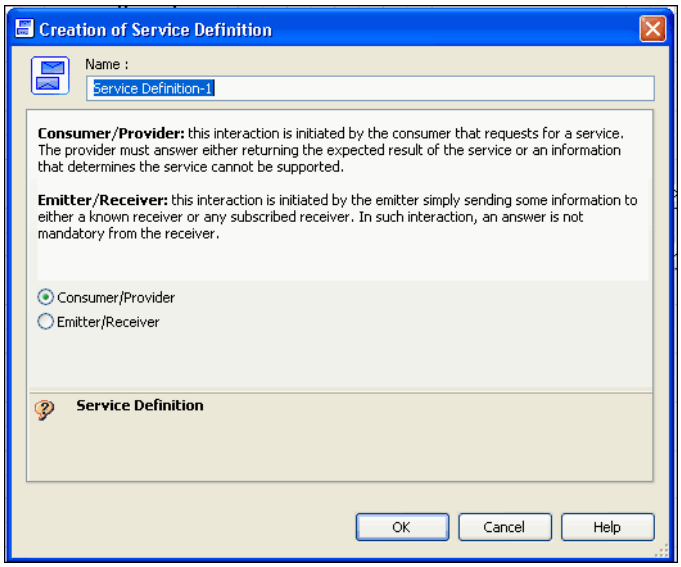
The main entry point of the NSOV-2 view is the Service Definition folder. This concept matches the definition of the exchange protocol that must be fulfilled to benefit from the exposed services.

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 or during the creation of interactions in diagrams.

During the creation of the service definition, the creation wizard presents two types of roles: "Consumer / Provider" and "Emitter / Receiver".



The "Consumer" and the "Emitter" are both initiators of the exchange.

The "Consumer / Provider" notion is used to indicate that an exchange contract exists between the consumer and the provider. When the consumer sends information requesting a particular service, a response is expected from the provider.

The "Emitter / Receiver" notion is used when the exchange occurs on the operational level to asses the possible information exchange between two nodes. With this kind of exchange, the "Emitter" sends out information to a receiver, however, a response from the receiver is not mandatory.

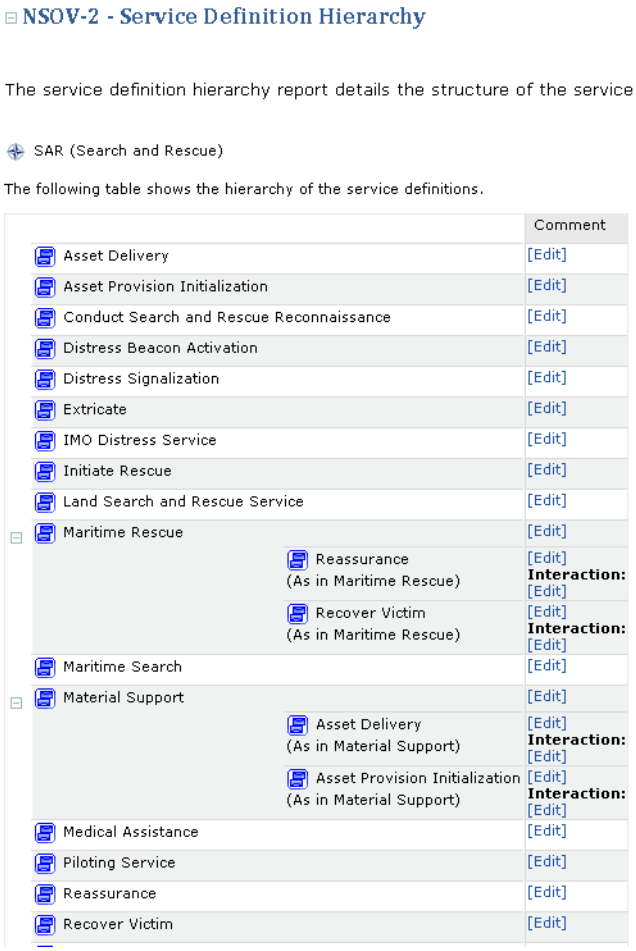
The NSOV-2 Report Template

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 Definition Subset 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 chapter lists the protocols of the architecture with their long name, short name and comment.

A paragraph is then displayed for each exchange protocol with a description diagram.

NSOV-3 Services to Operational Activities Mapping

The purpose of the Services to Operational Activities Mapping subview is to provide traceability by illustrating which services support which operational activities. In other words, it forms the traceability link between operational view and the service-oriented view.

The NSOV-3 subview shows which operational activities are supported by which services through the use of a mapping matrix. This subview is similar to other mapping matrices in NAF, for example, NCV-5 (Capabilities x System Deployments), NCV-6 (Capabilities x Operational Activities), NCV-7 (Capabilities x Services), NSV-5 (System functions x Operational activities), and NSV-12 (Systems x Services). Together, with these mapping subviews, NSOV-3 forms a line of reasoning that interrelates capabilities, operational (functional) activities, services and systems, through the use of traceability links.

✓ "The NSOV-3 Report Template", page 152
✓ "The NSOV-3 Functional Activity to Services Mapping Chapter", page 152

The NSOV-3 Report Template

Three parameters are used for the NSOV-3 report template:

- The default Architecture parameter that enables the retrieval of the default objects.
- The Business Service Subset parameter, which is optional. The type of object used is the Business Service. This parameter must be set with a subset of all the service definitions defined in the NSOV-2 view.
- The Activity subset parameter. This is an optional parameter. 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.

The NSOV-3 Functional Activity to Services Mapping Chapter

This chapter displays a matrix of the service definitions in rows and the functional activities in columns.

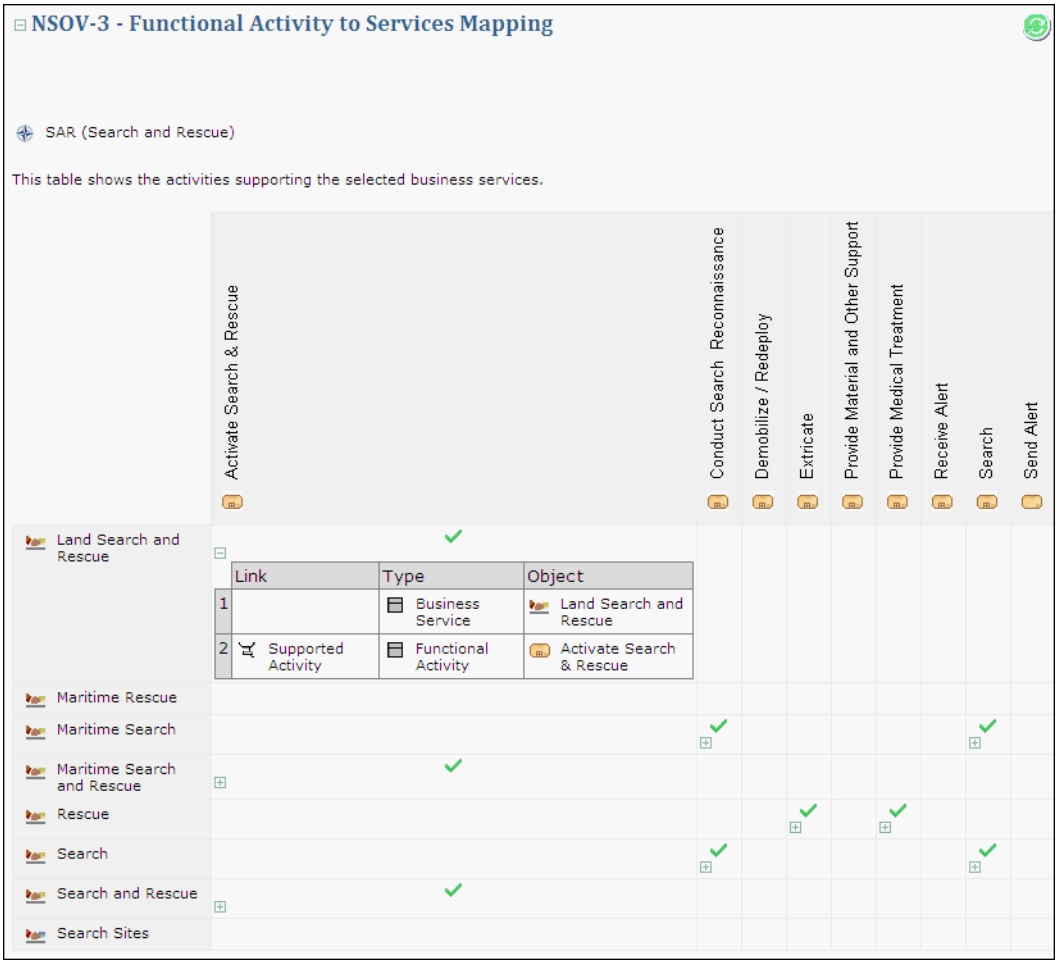
If no value is set for the Activity Subset parameter, all the functional activities and functional processes of the architecture are included.

If no value is set for the Service Definition Subset parameter, all the service definitions of the architecture are included.

A table displays the activities that support the selected business services. This support is indicated by a checkmark between the functional activity and the business service it supports, which shows that a link exists between the two.

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

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 the NSOV-3 Functional Activity to Services Mapping Chapter

NSOV-4 Service Orchestration

The NSOV-4 subviews help to determine the functional activities that actually support the services.

The information in these subviews derives from the model in the repository. This is not explicit information that is specified by the designer as is the case in NSOV-3.

A comparison can then be made of the intent declared in NSOV-3 based on the designer knowledge and the results computed from the study of the possible support by the functional processes (NOV-5).

The NSOV-4 reports shows this potential support through a comparison of the information flows that result from a functional activity and the information flow that is defined as a result in the exchange protocols displayed by the services. If the same results are retrieved, this could be a hint t o determine that a particular activity can be used to support a particular service.

✓ "The NSOV-4 Report Template", page 154
✓ "The NSOV-4 Service Orchestration Chapter", page 154
✓ "The NSOV-4 Detailed Service Orchestration Chapter", page 155

The NSOV-4 Report Template

The NSOV-4 Report Template produces two report chapters that show the activities that are supported by services. The template is defined with three parameters:

- The Architecture parameter, which specifies the architecture to be analyzed.
- The Service Subset parameter, which is optional. This parameter is set with services and is used to reduce the scope of the study. If no value is set for this parameter, all the services of the architecture are taken into account for the report.
- The Functional Activity parameter, which is also optional. This parameter is used to reduce the scope of the study to a subset of the functional activities of the architecture. If this parameter is not set all the functional activities defined in the architecture are taken into account.

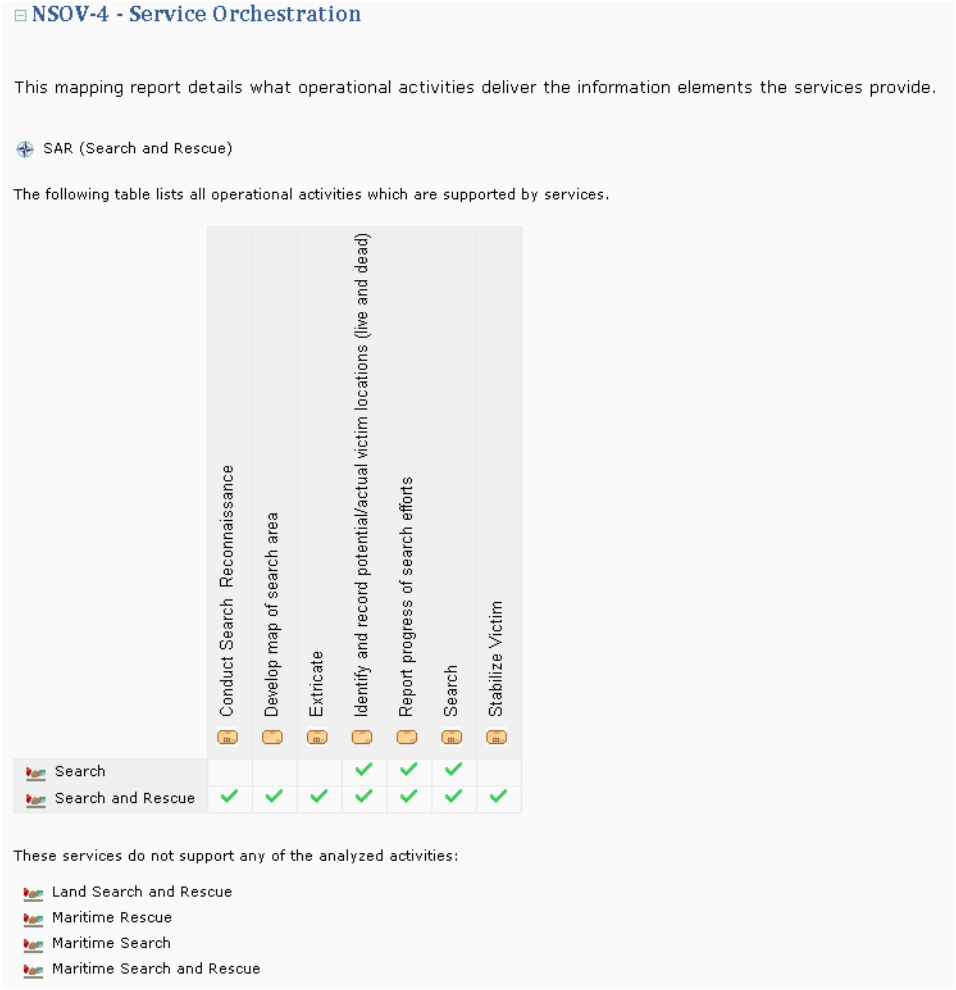
The NSOV-4 Service Orchestration Chapter

The NSOV-4 Service Orchestration chapter is used to evaluate the potential support of an activity by a service. This information is presented in a matrix composed of services in rows and activities in columns.

Checkmarks are added in the cells of the matrix to show the activities that are supported by services. Since the checkmarks are established from a complex path

research, it is not possible to add or remove them from the cells. This declaration must be performed in the NSOV-3 view.

The services that do not support any activities and the activities that are not supported by any services are presented in alphabtical lists.



NSOV-4 - Service Orchestration

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

SAR (Search and Rescue)

The following table lists all operational activities which are supported by services.

	Conduct Search Reconnaissance	Develop map of search area	Extricate	Identify and record potential/actual victim locations (live and dead)	Report progress of search efforts	Search	Stabilize Victim
Search				✓	✓	✓	
Search and Rescue	✓	✓	✓	✓	✓	✓	✓

These services do not support any of the analyzed activities:

- Land Search and Rescue
- Maritime Rescue
- Maritime Search
- Maritime Search and Rescue

Example of an NSOV-4 Service Orchestration Chapter

The NSOV-4 Detailed Service Orchestration Chapter

This chapter also shows a matrix with the activities that are supported by services. However, instead of checkmarks, the information elements that are shared by both the service and the activity that it supports. These information elements help to give an understanding of why the service is mentioned as a potential support of the activity.

The unsupported activities and the services that support no activities are listed below the matrix.

NSOV-4 - Detailed Service Orchestration

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

SAR (Search and Rescue)

The following table lists all operational activities which are supported by services and their common contents.

	Conduct Search Reconnaissance	Develop map of search area	Extricate	Identify and record potential/actual victim locations (live and dead)	Report progress of search efforts	Search	Stabilize Victim
Search				Victim not found	Extrication sites identified and prioritized	Extrication sites identified and prioritized Victim not found	
Search and Rescue	Search sites identified and prioritized	Search sites identified and prioritized	Stabilized Victims	Victim not found	Extrication sites identified and prioritized	Extrication sites identified and prioritized Victim not found	Stabilized Victims

These services do not support any of the analyzed activities:

- Land Search and Rescue
- Maritime Rescue
- Maritime Search
- Maritime Search and Rescue
- Rescue
- Search Sites

These activities are not supported by any of the analyzed services:

- Activate Distress Device
- Activate Search & Rescue
- Assemble personnel and equipment at designated location

Example of an NSOV-4 Detailed Service Orchestration Chapter

NSOV-5 SERVICE BEHAVIOUR

The aim of the NSOV-5 Service Behaviour subview is to specify the function and behaviour of individual services in an architecture. This is done through the creation of service interaction scenarios in the **HOPEX NAF** tool.

An interaction scenario is graphically described by a sequence diagram to which a specific set of instances is attached. In this view, services and service compositions are the two kinds of instances that can be created.

An interaction scenario diagram therefore illustrates a scenario in which services are involved. Instances can also be service compositions if the services on which they are based are only used in particular contexts (the upper service to which the composition belongs).

With service instances you can place one or more instances that are based on the same service type, for example, to explain what happens if two services of the same nature are requested to exchange with each other.

When service instances have been created, you can add instance messages to describe the information exchanges performed in the scenario.

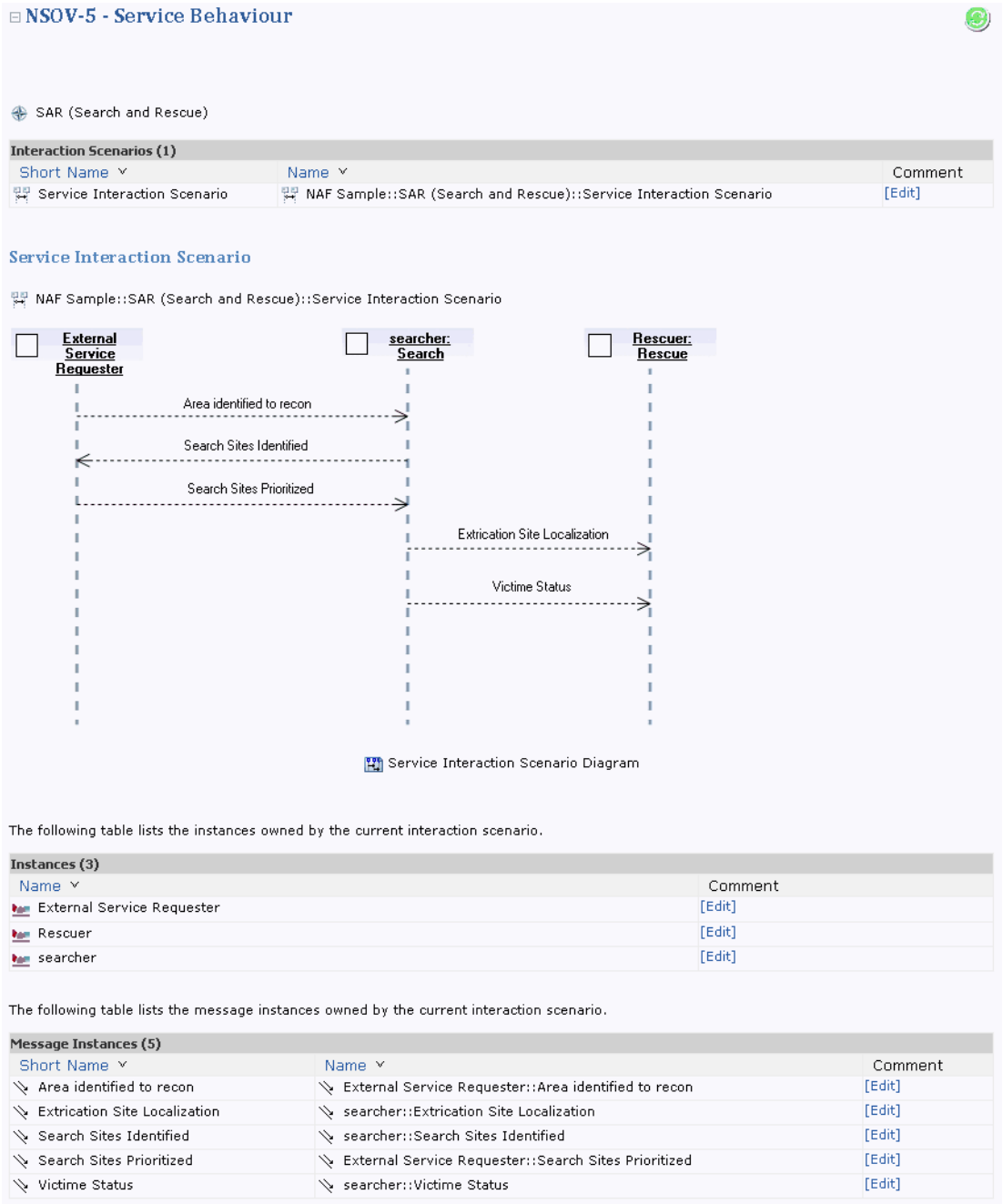
Instance messages are defined to specify the information exchange element (content) that should be taken from the set of possible information elements defined in the interactions between the two communicating services.

NSOV-5 Service Behaviour Chapter

This chapter presents a table with all the service scenario interaction diagrams retrieved during the analysis.

The diagram for each scenario is also presented.

Each instance and message instance owned by the interaction scenario is presented in tables with its name and comment.



NSOV-5 - Service Behaviour

SAR (Search and Rescue)

Interaction Scenarios (1)

Short Name	Name	Comment
Service Interaction Scenario	NAF Sample::SAR (Search and Rescue)::Service Interaction Scenario	[Edit]

Service Interaction Scenario

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

Service Interaction Scenario Diagram

The following table lists the instances owned by the current interaction scenario.

Instances (3)

Name	Comment
External Service Requester	[Edit]
Rescuer	[Edit]
searcher	[Edit]

The following table lists the message instances owned by the current interaction scenario.

Message Instances (5)

Short Name	Name	Comment
Area identified to recon	External Service Requester::Area identified to recon	[Edit]
Extrication Site Localization	searcher::Extrication Site Localization	[Edit]
Search Sites Identified	searcher::Search Sites Identified	[Edit]
Search Sites Prioritized	External Service Requester::Search Sites Prioritized	[Edit]
Victime Status	searcher::Victime Status	[Edit]

Example of an NSOV-5 Service Interaction Scenario Chapter

NAF TECHNICAL VIEWS SUBVIEWS

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

- ✓ "NTV-1 - Standards Profile", page 160
- ✓ "NTV-2 Standards Forecast", page 162
- ✓ "NTV-3 - Standard Configurations", page 165

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

In the **HOPEX NAF** product, 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", page 277 and "NTV-2 Standards Forecast", page 277.

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.*

Example:

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

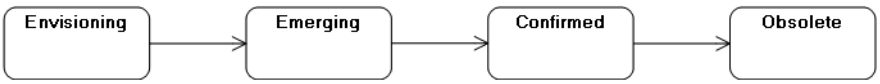
This chapter presents the standards and guidelines that apply to the analyzed architecture(s). Paragraphs NSV-1 to each standard follow, listing the elements that use the stanard in question.

NTV-2 Standards Forecast

The purpose of the Standards Forecast subview (NTV-2) is to identify emerging, obsolete and fragile standards, and 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 the **HOPEX NAF** application, 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", page 162
✓ "Customization", page 163
✓ "The NTV-2 Report Template", page 163
✓ "The NTV-2 Standards Forecast Chapter", page 163
✓ "NTV-2 Standard Forecast Description Chapter", page 163

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" and "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 subview comes with a report template that produces two chapters.

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.

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

The NTV-3 Report Template

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 capabilty configurations and then the related standards match up with the configurations that have been agreed on.

NAF System Views Subviews

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

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, thiscontract 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

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**, **NSV-1 System Interface Description** and **Resource Architectures** and **All Resource Architectures** folders.
2. Right-click the resource architecture concerned and select **Properties**.
3. In the **Characteristics** tab, **Configured Capabilty** box, 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.*

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 hierararchy 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 hierararchy 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 hierararchy 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 hierararchy 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 hierararchy 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 hierararchy 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

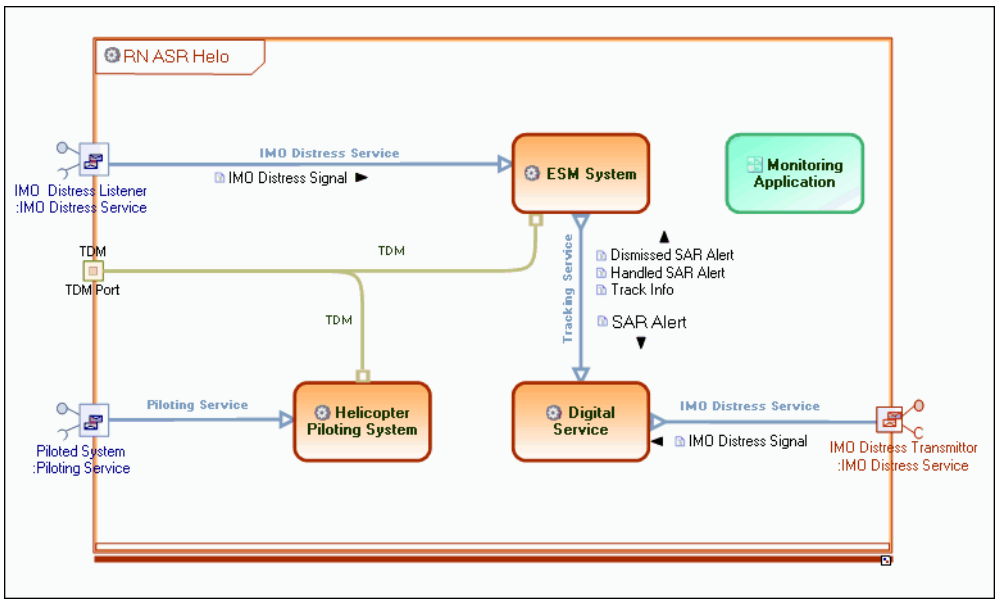
NSV-2c System Connectivity Clusters

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

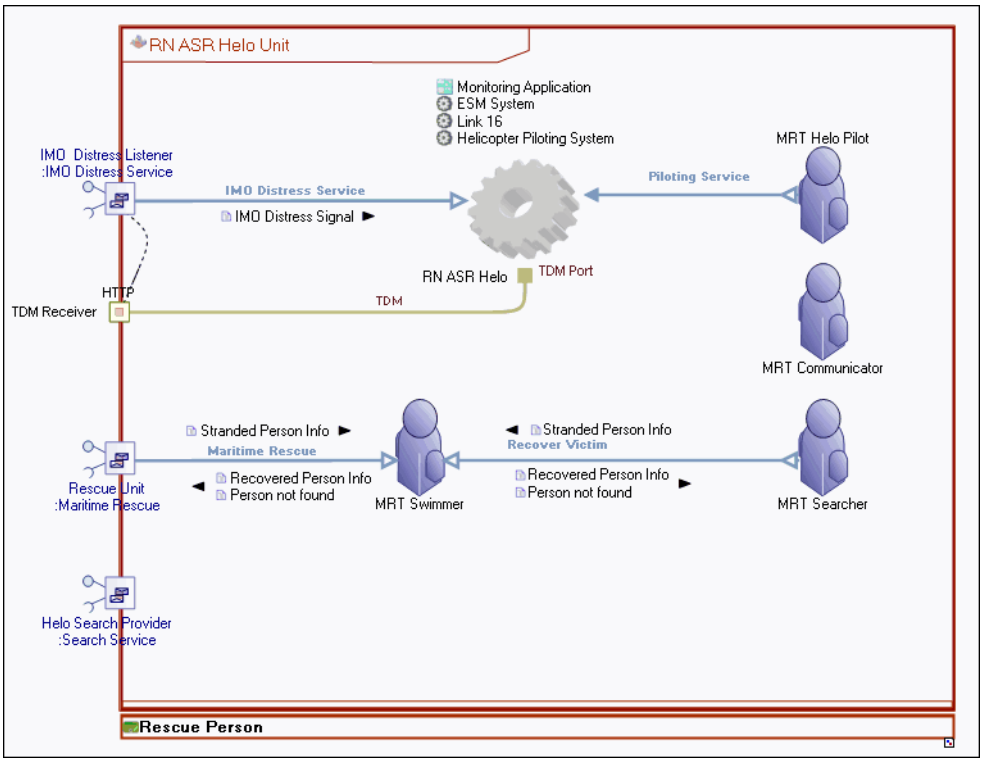
In the example below, the RN ASR Helo system displays a TDM port that connects the ESM System and the Helicopter Piloting Systems.



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 connectivtiy clusters.

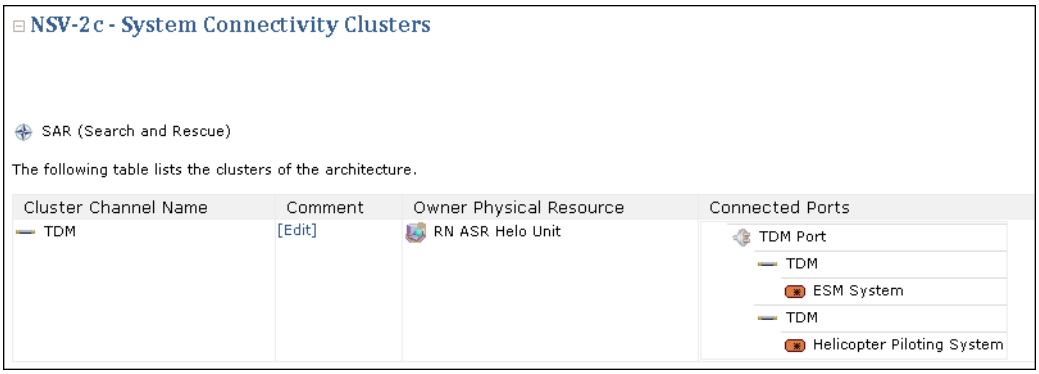
The parameters taken into account for the report are:

- 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 Connectivty 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 Ssyetm 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 onl category supported by this subview is the "Electromagnetic Spectrum and Bandwidth" categoty.

In addition to NSV-7 that specificies the quality requirements for systemson 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 and on the NAF start page. No other objects are displayed since the aim is to synthesize information already designed in the NSV-1 section.

✓ "The NSV-3 Report Template", page 184
✓ "The NSV-3 Resources to Resources Channel Matrix Chapter", page 184
✓ "The NSV-3 Resources to Resources Interaction Matrix Chapter", page 185

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.

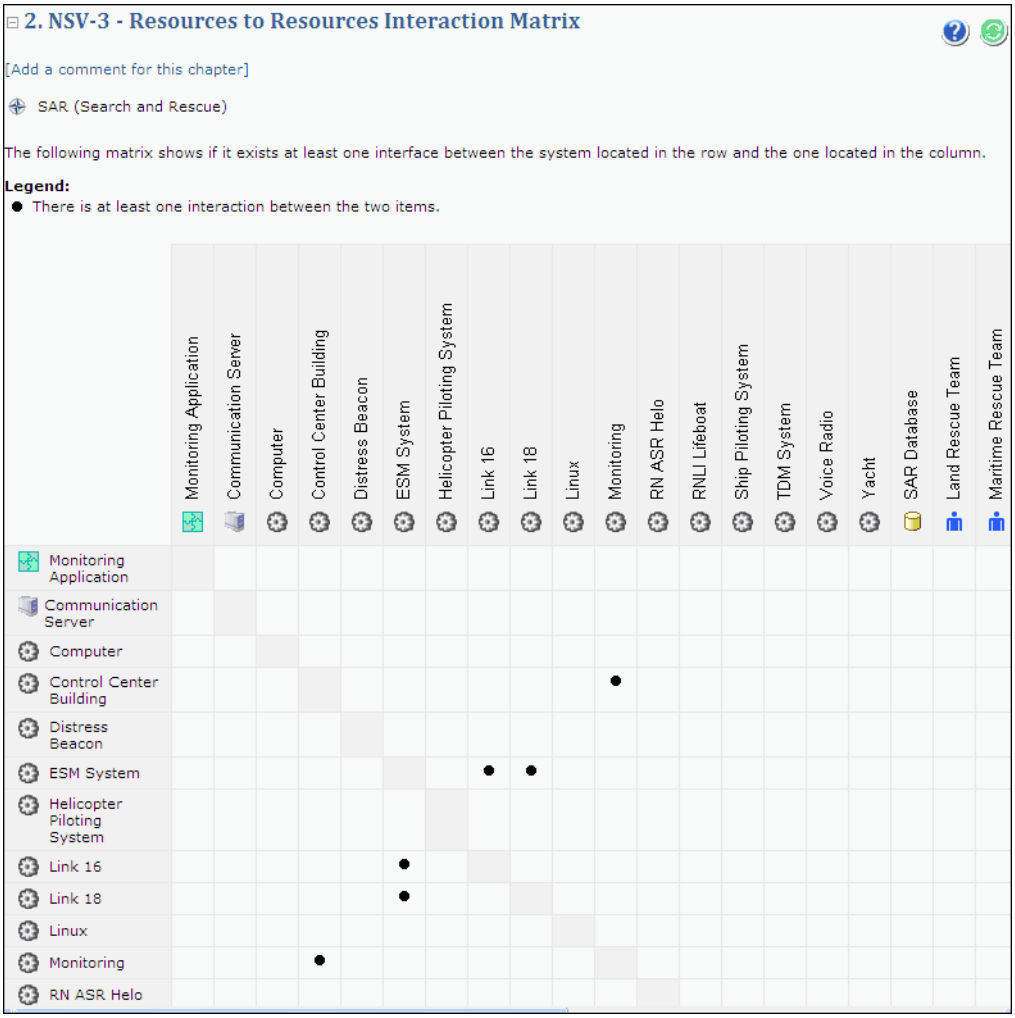
Diagonal cells dealing with the same source and target are grayed.

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 squared 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.



2. NSV-3 - Resources to Resources Interaction Matrix

[Add a comment for this chapter]

SAR (Search and Rescue)

The following matrix shows if it exists at least one interface between the system located in the row and the one located in the column.

Legend:

● There is at least one interaction between the two items.

	Monitoring Application	Communication Server	Computer	Control Center Building	Distress Beacon	ESM System	Helicopter Piloting System	Link 16	Link 18	Linux	Monitoring	RN ASR Helo	RNLI Lifeboat	Ship Piloting System	TDM System	Voice Radio	Yacht	SAR Database	Land Rescue Team	Maritime Rescue Team
Monitoring Application																				
Communication Server																				
Computer																				
Control Center Building											●									
Distress Beacon																				
ESM System								●	●											
Helicopter Piloting System																				
Link 16						●														
Link 18						●														
Linux																				
Monitoring				●																
RN ASR Helo																				

Example of an NSV-3 Ressouces to Ressouces Interaction Matrix

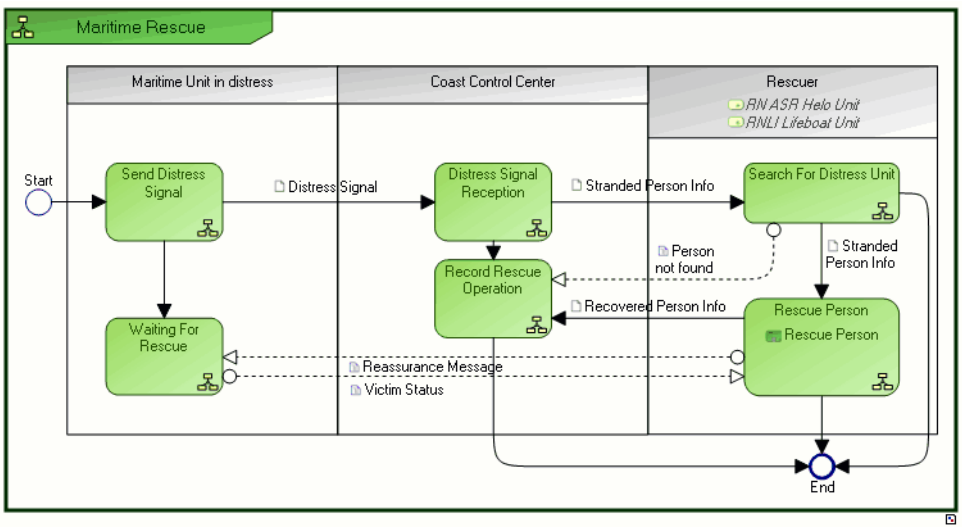
NSV-4 Systems Functionality Description

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

System process modeling is similar to NOV-5 and NOV-6a process modeling (BPMN-based).

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

See:

✓ "The NSV-4 Report Template", page 187
✓ "The NSV-4 System Function Composition Hierarchy Chapter", page 187
✓ "The NSV-4 System Function Dictionary Chapter", page 188
✓ "The NSV-4 System Function Exchange Chapter", page 188
✓ "The NSV-4 System Function Exchange Balance Chapter", page 188

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 hierararchy 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 proces 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 hierararchy 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

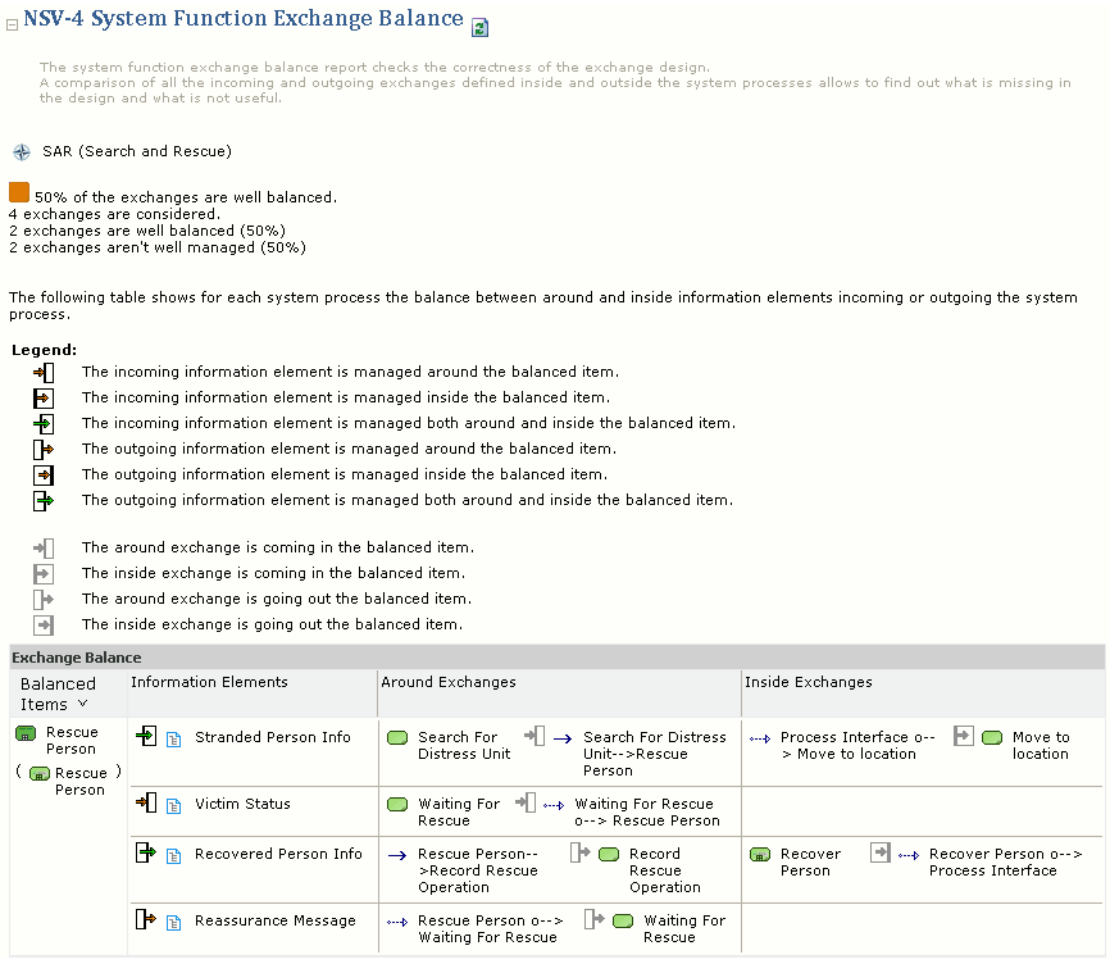
The NSV-4 System Function Exchange chapter lists the interactions that are set between the tasks of system processes. It shows the diagrams that display information exchanges between system processes and system functions. It also details the information exchange between the selected tasks. All messages 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.



NSV-4 System Function Exchange Balance

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

SAR (Search and Rescue)

50% of the exchanges are well balanced.
4 exchanges are considered.
2 exchanges are well balanced (50%)
2 exchanges aren't well managed (50%)

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

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
Rescue Person (Rescue Person)	Stranded Person Info	Search For Distress Unit → Search For Distress Unit-->Rescue Person	Process Interface o--> Move to location → Move to location
	Victim Status	Waiting For Rescue ⇢ Waiting For Rescue o--> Rescue Person	
	Recovered Person Info	Rescue Person-->Record Rescue Operation → Record Rescue Operation	Recover Person ⇢ Recover Person o--> Process Interface
	Reassurance Message	Rescue Person o--> Waiting For Rescue → Waiting For Rescue	

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", page 191
✓ "The NSV-5 Systems Function to Operational Activity Traceability Matrix Chapter", page 192

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 where rows contain the selected system functions and columns contain the operational activities. Mappings between system functions and operational 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 nonattainment 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.

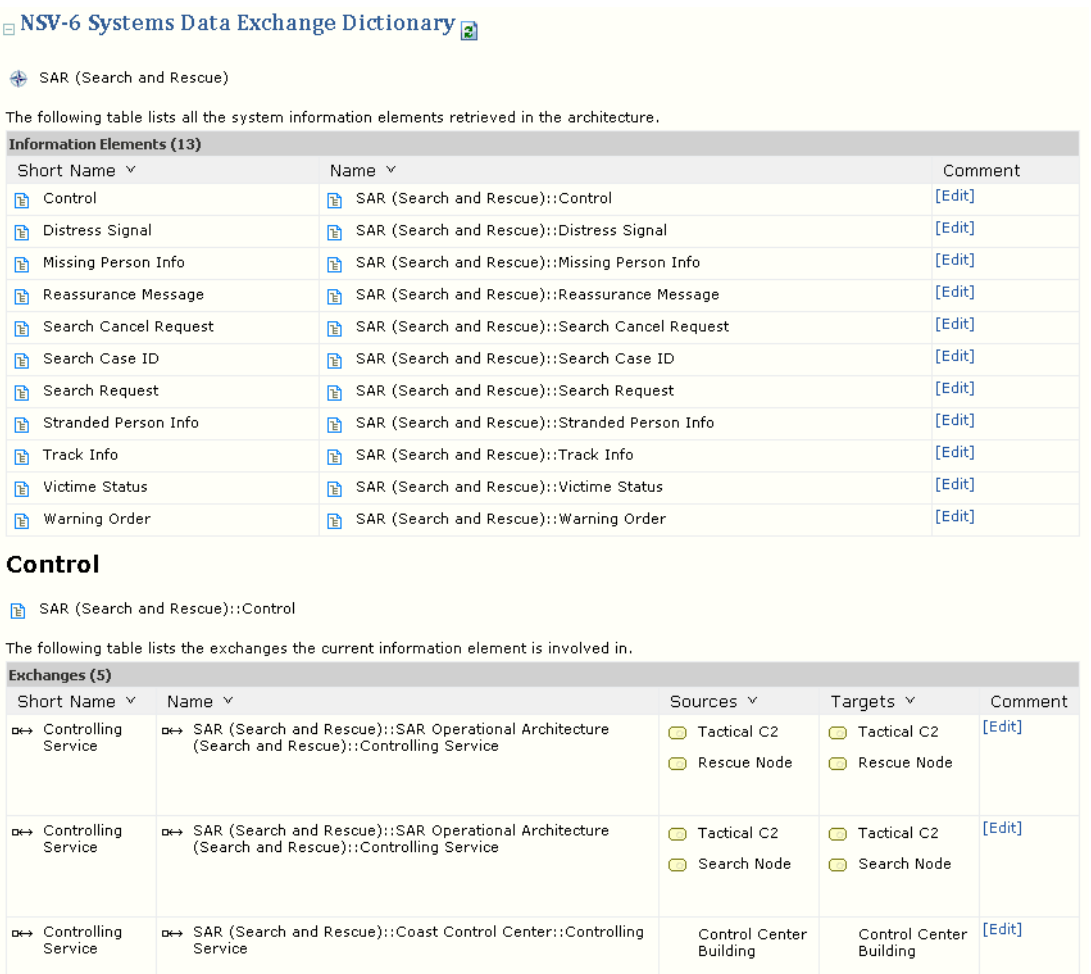
- The first parameter is the Architecture to be analyzed.
- The second parameter is an optional value that is used to reduce the scope of the study. This parameter is set with interfaces described in the architecture. In this case, the NSV 6 report focuses on this set of interfaces and on the information 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]

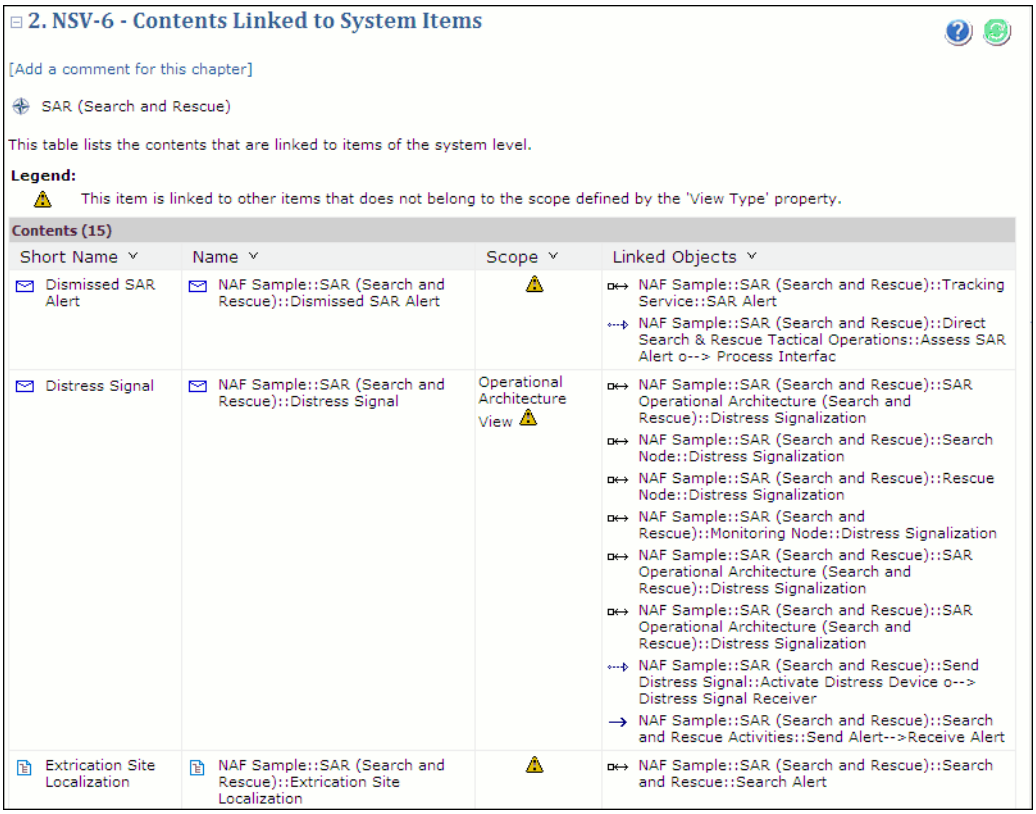
Example of an NSV-6 Systems Data Exchange Dictionary

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) identifes the quality requirements considered crucial to the successfull 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), and 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).

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", page 197
✓ "Displaying the Gantt Chart", page 198
✓ "The NSV-8 Report Template", page 200
✓ "The NSV-8 Gantt chart Chapter", page 200
✓ "The NSV-8 Systems Evolution Description Chapter", page 200
✓ "The NSV-8 System Evolution Comparison Chapter", page 200

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. Right click on the new master plan.
2. Select the **Open Gantt Chart** command.
   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:

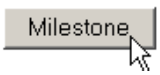
- Create milestones
- Create planned configuration capabilities
- Add the time periods of objects to be planned.
- Indicate the lifecycle status of the planned objects.

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.*

Creating planned capability configurations with time periods and lifecycle status

You can describe the life periods of configuration capabilities.

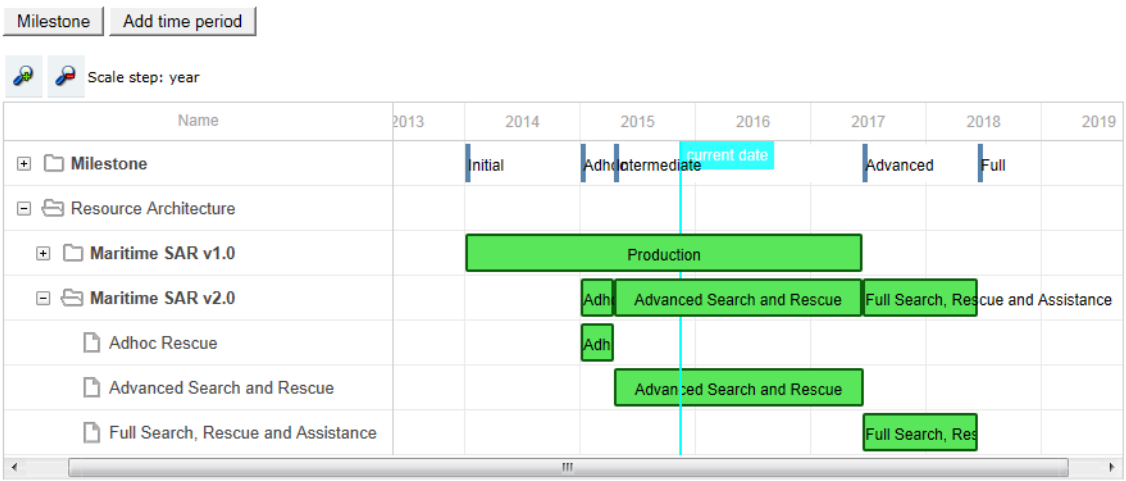
To create planned configuration capabilities with time periods and lifecycle status:

1. Above the master plan table, click the **Add Time Period** button.
2. In the **Query** dialog box that appears, use the **Find** button to find the required object, in this case the resource architecture of the capability configuration.
3. In the list proposed, select your object and click **OK**.

The dialog box for adding planned items opens with the chosen object.

4. Click **Next** to continue.
   You are then required to select the lifecycle status of the object. This cycle status is attached to the planned configuration capability and is integrated between two milestones.
   You have the choice between three cycle states:
   - Preparation
   - Production
   - Retirement
5. Select the lifecycle status that applies and click **Next**.
   You are then required to select dates that indicate the beginning and end of the availability of the planned configuration capability.
6. Select the start and end dates for the planned item then click **Finish**.
   The planned system appears in the diagram along with the name of the system.

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



Example of an NSV-8 Gantt Char

☺ *You can access the properties of an object by selecting this object then in the "Selected Elements" box right-click the name of the object.*

For more details on master plan modeling, refer to the **HOPEX Planning** user guide.

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", page 201
✓ "The NSV-9 Technology Forecast Chapter", page 201
✓ "The NSV-9 Technology Forecast Description Chapter", page 201

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 and the NAF start page 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 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.

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

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

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

The NSV-10b Report Template

The NSV-10b report template comes with two parameters:

- 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.

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.

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", page 210
- ✓ "NPV-2 - Programme to Capability Mapping", page 214

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.

The NPV-1 Report Template

The NPV-1 report template uses three parameters:

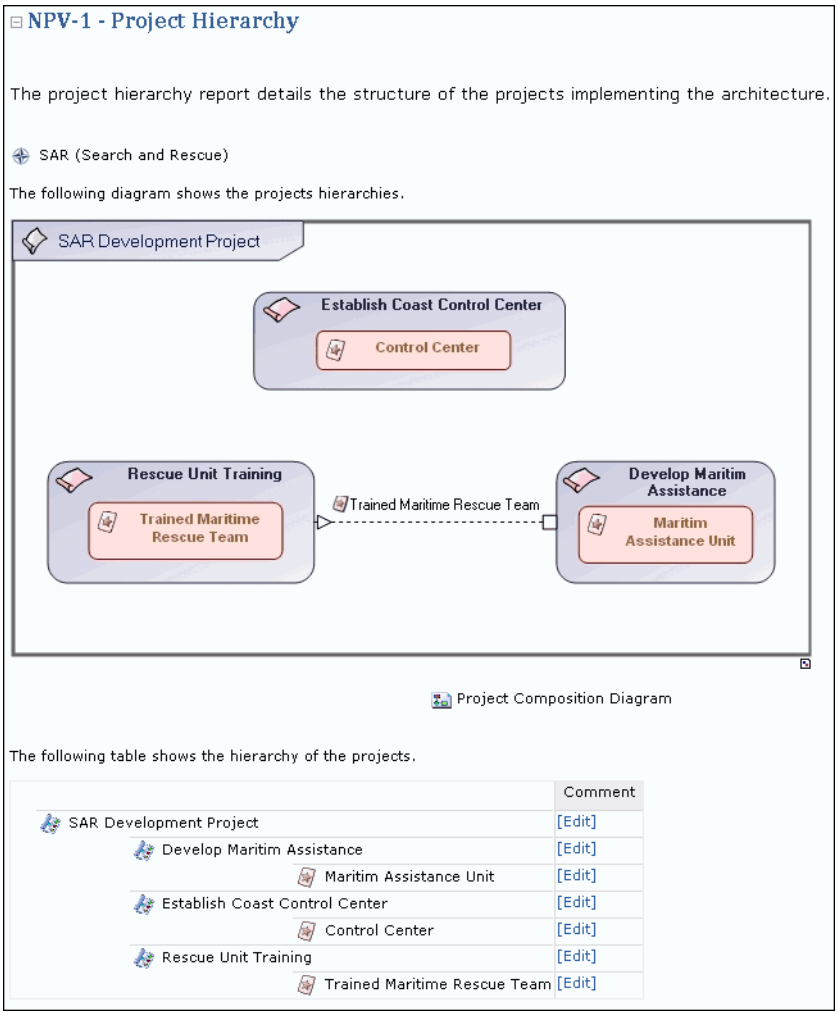
- The Architecture parameter, which is the architecture to be analyzed. This is the sole parameter that enables the retrieval of all the root projects of the architecture.
- The Projects parameter, which is optional. This parameter is used to reduce the scope of the study. The selected projects must 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.



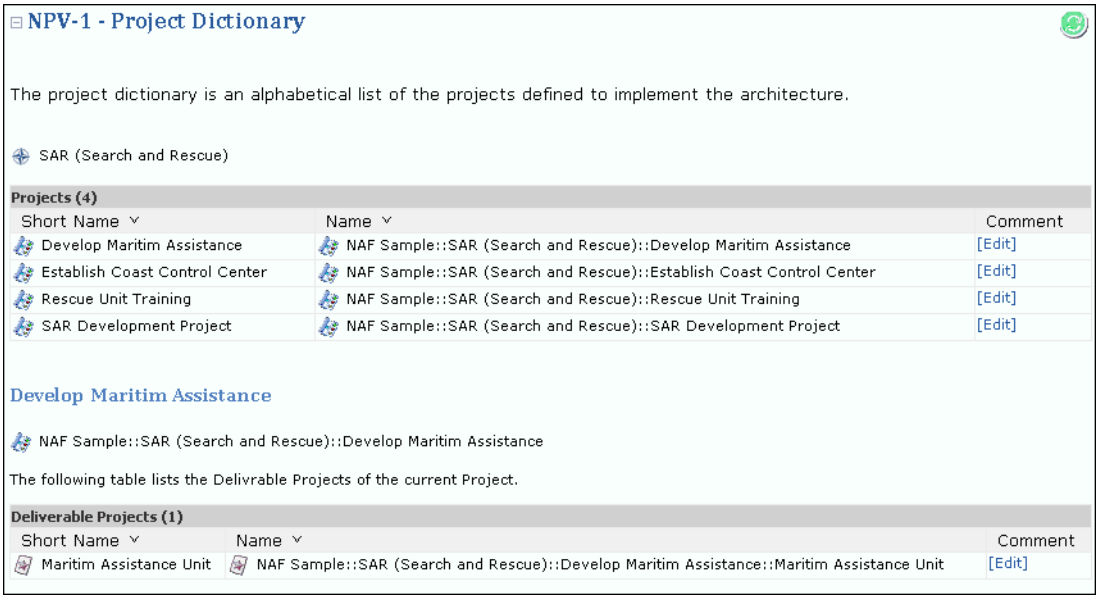
Example of a Project Hierarchy Report

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]

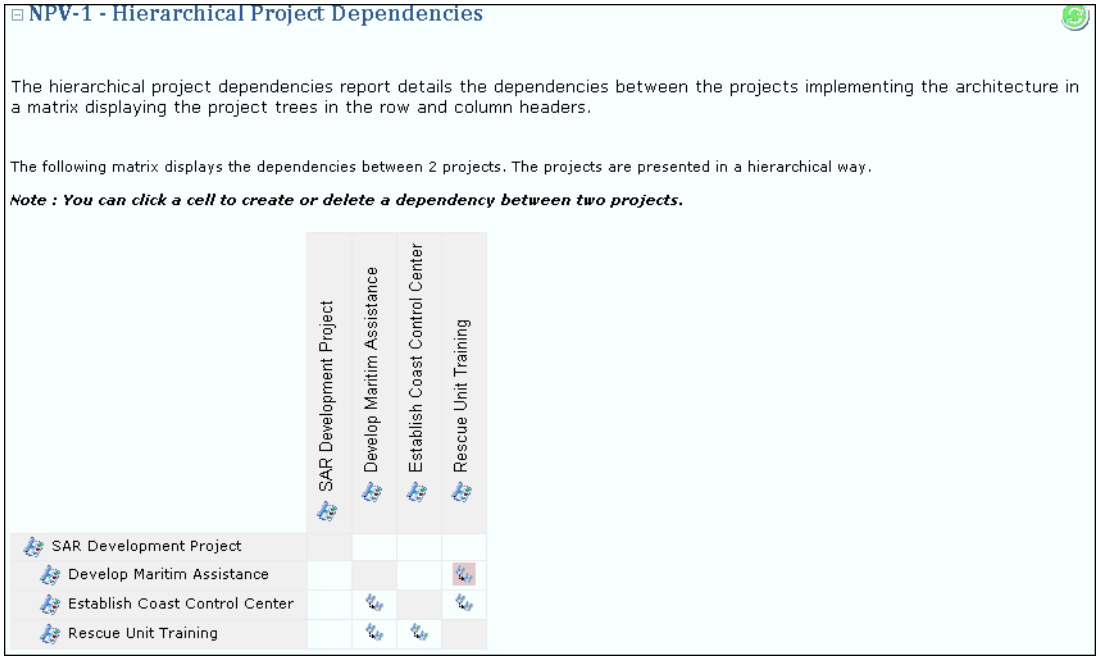
Example of a Project Dictionary Chapter

The NPV-1 Hierarchical Project Dependencies Chapter

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 Template

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.

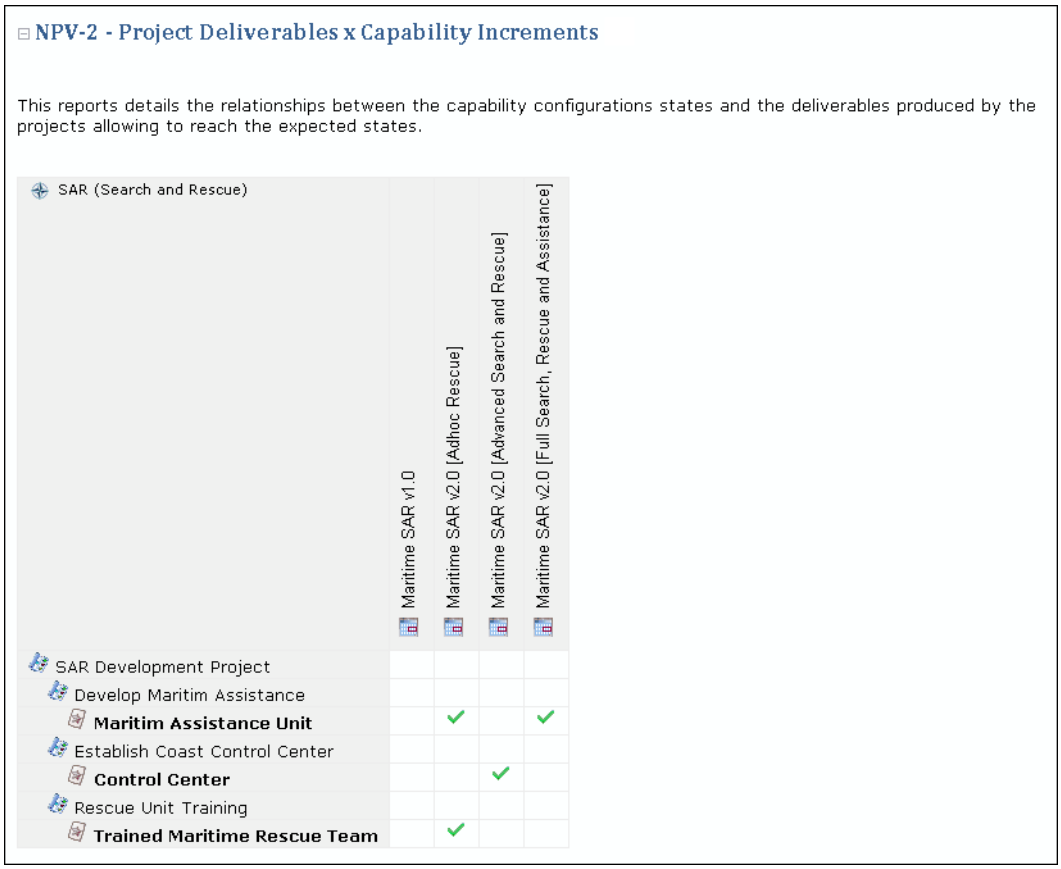
The NPV-2 Project Deliverables X Capability Increments Chapter

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 and time periods as column headers.

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

11

MODELING RULES AND REGULATIONS

The points dealt with in this chapter are:

SUBMODELING REGULATION

The NAF Modeling Regulation is a new regulation (set of consistency rules) dedicated to NAF. It includes the following sub-regulations that already exist:

- Report Regulation
- BPMN Regulation
- Isolated Object Regulation
- Message Flow Regulation
- Project and Method Regulation
- Structure & Organization Regulation
- Time Constraint Regulation
- Variation Modeling Regulation

NAF MODELING RULES

The HOPEX Modeling tool comes with a set of consistency rules that can be used to verify that modeled architectures are well-designed. You can also set your own rules. For further information on rules and regulations in **HOPEX** Modeling Suite, see the **HOPEX Common Features** EN manual, "Consistency Rules" paragraph.

In order to assist the user designing NAF architectures, a set of rules have been added that can be activated to dynamically check the relevance of certain configurations. When these rules are set, warnings are displayed in the navigation tree and/or diagrams that explain the pending issue for each architecture item and offer solutions to correct them.

The table below displays additional rules that are specific to the NAF context. All rules that are already defined can be used to check a NAF architecture.

Component Naming

Rule	Type of Rule	Metaclass concerned	Rule description
An organizational process should not contain more than one operation with the same name	Recommendation	Organizational Process	An organization process having two or more operations with the same may lead to confusing report since it is not possible to distinguish one from the other. List the objects having the same names.
A system process should not contain more than one task having the same	Recommendation	System Process	A system process having two or more tasks with the same may lead to confusing report since it is not possible to distinguish one from the other. List the objects having the same names. Development Note: This already existing rule should be adapted to manage task and operation in a system process
A capability should not contain more than one capability composition having the same name	Recommendation	Capability	A capability having two or more capability compositions with the same may lead to confusing report since it is not possible to distinguish one from the other. List the objects having the same names.
An operational node should not contain more than one operational component having the same name	Recommendation	Business Function (Operational Node)	An operational node having two or more components with the same may lead to confusing report since it is not possible to distinguish one from the other. List the objects having the same names.

Rule	Type of Rule	Metaclass concerned	Rule description
A functional process should not contain more than one functional activity having the same name	Recommendation	Functional Process	A functional process having two or more activities with the same may lead to confusing report since it is not possible to distinguish one from the other. List the objects having the same names.
An artifact should not contain more than one artifact component having the same name	Recommendation	Artifact	An artifact having two or more components with the same may lead to confusing report since it is not possible to distinguish one from the other. List the objects having the same names.
An artifact should not contain more than one application host having the same name	Recommendation	Artifact	An artifact having two or more application hosts with the same may lead to confusing report since it is not possible to distinguish one from the other. List the objects having the same names.

Rule	Type of Rule	Metaclass concerned	Rule description
A resource architecture should not contain more than one architecture use having the same name	Recommendation	Resource Architecture	A resource architecture having two or more architecture uses with the same may lead to confusing report since it is not possible to distinguish one from the other. List the objects having the same names.
A resource architecture should not contain more than one physical asset having the same name	Recommendation	Resource Architecture	A resource architecture having two or more physical assets with the same may lead to confusing report since it is not possible to distinguish one from the other. List the objects having the same names.
A resource architecture should not contain more than one human asset having the same name	Recommendation	Resource Architecture	A resource architecture having two or more human assets with the same may lead to confusing report since it is not possible to distinguish one from the other. List the objects having the same names.

Ownerships

Rule	Type of Rule	Metaclass concerned	Rule description
An operation must be defined in the context of a organizational process	Requirement	Operation	An operation is a component defining an organization process. It cannot be defined outside the scope of an organizational process.
A task must be defined in the context of system process	Requirement	Task	A task is a component defining a system process. It cannot be defined outside the scope of a system process.
A capability composition must be defined in the context of a capability	Requirement	Capability Composition	A capability composition is a component defining a capability. It cannot be defined outside the scope of a capability.
A capability dependency must be defined in the context of a capability	Requirement	Capability Dependency	A capability dependency determines what components of a capability are dependent from eachother. This definition must be established in the context of the components owner (the upper capability).
An operational component must be defined in the context of an operational node	Requirement	Operational Component	An operational component is a component defining an operational node. It cannot be defined outside the scope of an operational node.
A functional activity must be defined in the context of an functional process	Requirement	Functional Activity	A functional activity is a component defining a functional process. It cannot be defined outside the scope of a functional process.

Rule	Type of Rule	Metaclass concerned	Rule description
An artifact component must be defined in the context of an artifact	Requirement	Artifact Component	An artifact component is a component defining an artifact. It cannot be defined outside the scope of an artifact.
An application host must be defined in the context of an artifact	Requirement	Application Host	An application host is a component defining an artifact. It cannot be defined outside the scope of an artifact.
An architecture use must be defined in the context of a resource architecture	Requirement	Architecture Use	An architecture use is a component defining a resource architecture. It cannot be defined outside the scope of a resource architecture.
An physical asset must be defined in the context of a resource architecture	Requirement	Physical Asset	A physical asset is a component defining a resource architecture. It cannot be defined outside the scope of a resource architecture.
An human asset must be defined in the context of a resource architecture	Requirement	Human Asset	A human asset is a component defining a resource architecture. It cannot be defined outside the scope of a resource architecture.

Data Modeling

Rule	Type of Rule	Metaclass concerned	Rule description
The use of the Entity (DM) / Data Model relationship is not recommended in the DoDAF 2 context	Recommendation	Entity (DM)	The relationship allowing the linking of an data entity to a detailing data model does not belong to the UML notation. It is not recommended to use to create data model since it will not be handle by the NAF reports.

Participation

Rule	Type of Rule	Metaclass concerned	Rule description
A participant of an organizational process should only be assigned by org-units or human assets	Requirement	Participant	It is recommended to not assign architecture items that are not org-units or human assets to the organizational process participants since the exchange compliance reports will not able to establish proper analysis. List assigned objects that are not conformed to the rule.
A participant of an functional process should only be assigned by operational nodes or operational components	Requirement	Participant	It is recommended to assign architecture items that are not operational nodes or operational components to the functional process participants since the exchange compliance reports will not able to establish proper analysis. List assigned objects that are not conformed to the rule.
A participant of an system process should only be assigned by artifacts, artifact components, applications, application hosts, org-units, human assets, resource architectures or architecture uses	Requirement	Participant	It is recommended to assign architecture items that are not artifacts, artifact components, applications, application hosts, org-units, human assets, resource architectures or architecture uses to the system process participants since the exchange compliance reports will not able to establish proper analysis. List assigned objects that are not conformed to the rule.

Service Definition (Protocol)

Rule	Type of Rule	Metaclass concerned	Rule description
A message must be connected to one producer and one consumer role of the service definition	Requirement	Message	Development Note: Check that there is a message rule that checks that a message is not connected to two roles. This rule must be updated to include the protocol context exception.
A service point must play a provider role of a service definition.	Requirement	Service Point	A service point plays one of the roles of a service definition. This role must be a provider role so that the service point is a supporter for the exposed service. List the wrong roles. Appliance Condition: The service point is linked to any role that is not a provider role of a service definition (protocol).

Rule	Type of Rule	Metaclass concerned	Rule description
A request point must be play a consumer role of a service definition	Requirement	Request Point	A request point plays one of the roles of a service definition. This role must be a consumer role so that the request point is a consumer of the requested service. List the wrong roles. Appliance Condition: The request point is linked to any role that is not a consumer role of a service definition (protocol).
A service point should be linked to a service definition role.	Suggestion	Service Point	A service point is designed to explain what service is exposed to the consumers. It is more accurately defined if a provider role of a service definition is connected. Appliance Condition: The service point is not connected to a role.
A request point should be linked to a service definition role.	Suggestion	Request Point	A request point is designed to explain what service is requested by the interacting item. It is more accurately defined if a consumer role of a service definition is connected. Appliance Condition: The service point is not connected to a role.

BPMN Modeling

Rule	Type of Rule	Metaclass concerned	Rule description
A functional process must include only functional activities that can be based on sub-processes	Requirement	Functional Process	In order to ensure process analyses, each functional process is described by sequencing and interacting functional activities. If a sibling functional process is involved then a call to it is specified linking the appropriate activity. This enables the separation between the functional process definition and the references to other existing processes.
An organizational process must include only operations that can be based on sub-processes	Requirement	Organizational Process	In order to ensure process analyses, each organizational process is described by sequencing and interacting operations. If a sibling organizational process is involved then a call to it is specified linking the appropriate operation. This enables the separation between the organizational process definition and the references to other existing processes.
An system process must include only tasks that can be based on sub-processes	Requirement	System Process	In order to ensure process analyses, each system process is described by sequencing and interacting tasks. If a sibling system process is involved then a call to it is specified linking the appropriate task. This enables the separation between the system process definition and the references to other existing processes.

Dependency

Rule	Type of Rule	Metaclass concerned	Rule description
Only one dependency should be set between two capability compositions	Recommendation	Capability Dependency	There is more than one capability dependency that links the same capability sources and targets. List all the dependencies. Development Note: Check all the sources and targets thinking that more than one source can exist.
Only one dependency should be set between two projects	Recommendation	Project Dependency	There is more than one project that links the same project sources and targets. List all the dependencies. Development Note: Check all the sources and targets thinking that more than one source can exist.

THE HOPEX METAMODEL FOR NAF

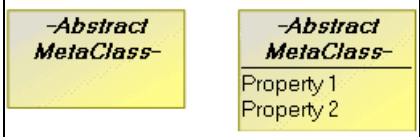
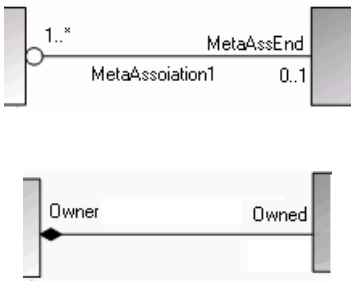
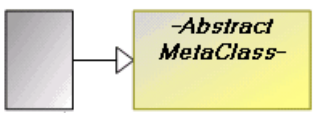
This chapter presents all the **HOPEX** metamodels used to implement the NAF views.

A graphical representation of each metamodel is included, each with **HOPEX** names. If the names are changed when the **NAF Metamodel Customizations** add-on is imported, the new names are highlighted in yellow near the **HOPEX** names.

☛ *Some views are not directly supported by a dedicated metamodel. They are instead linked to data modeled in two or several other views.*

Notation

The metamodel diagrams in this section are based on the UML notation. Below are a few points to help you better understand these diagrams.

Concept	Meaning	Notation
MetaClass	Defines object type. Each rectangle may include a list of properties.	
Abstract MetaClass	This concept cannot be instantiated and must be inherited from concrete metaclasses (gray rectangle). Each rectangle may include a list of properties.	
Association	For each association : - Association end names are displayed to indicate the relationship between the linked objects. - The multiplicity (1, 0..1, *) is displayed for some associations. This determines the range of linkable objects. - A white circle at the end of an association indicates which is the major object in the association, i.e., the object which is modified in case the association is removed. - A black diamond at the end of an association indicates that the other object is a component of the major object. It is a strong association between the two objects. When the major object is deleted, all is components are also deleted.	
Inheritance	Inheritances are illustrated by arrows drawn between concepts.	

Examples

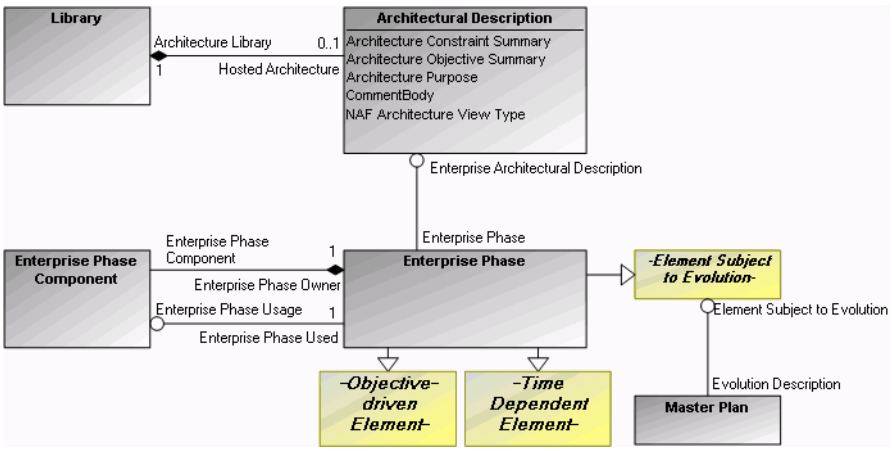
Where appropriate diagrams exist in, metamodels are illustrated by examples. These examples are used to illustrate how notions and relationships are drawn in specific diagrams. The content is therefore not at all concrete; the names of items are based on the names of notion suffixed by a random number so that the reader is able to associate the graphical representation to their corresponding object types.

ALL VIEWS METAMODELS

NAV-1 Overview and Summary Information

NAV-1 defines the architecture (Architecture Description) contained in a **HOPEX** library. All items of the architecture are automatically defined in the associated library. Additional properties define the general purpose of the architecture.

At the top level of the architecture description, the designer describes the enterprise structure that supports the architecture design and performance. Each item of this structure can be linked to a capability master plan that corresponds to the enterprise vision at the appropriate level.



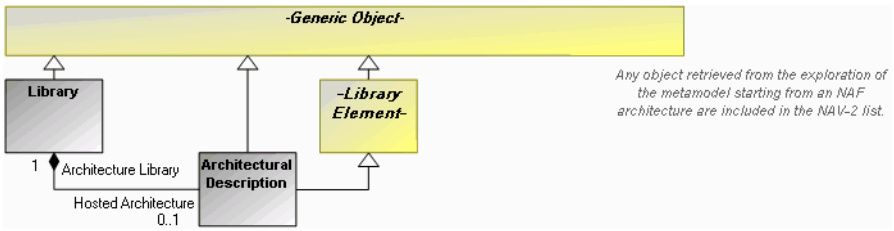
NAV-1 metamodel

An enterprise structure is composed of Enterprise Phase items that contain sub-items (Enterprise Phase Component). This is a whole-part model.

> ☛ *Note that compared with the 2009 SP4 release, the use of org-units with an Enterprise type is replaced by a dedicated notion (Enterprise Phase). For existing models, a conversion tool must be applied that creates as many enterprise phases as corresponding org-units. The previous org-units must be deleted by the user. The structure diagrams used to describe previous org-units are not converted since the enterprise phase should not introduce interactions between its sub-items like an org-unit model permits.*

NAV-2 Integrated Dictionary

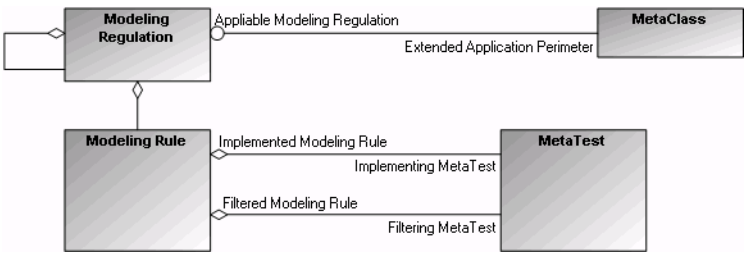
The integrated dictionary is composed of all items defined in the context of the Architectural Description object. From this object, the container library is retrieved via the Architecture Library association. The library is then explored to retrieve all contained items based on the **HOPEX** extraction tool. The dictionary is composed of these items.



NAV-3a Architecture Compliance Statement (Metadata)

This Metamodel proposes modeling regulations that apply to NAF architectures. It assists in checking and ensuring that models are NAF-compliant with the modeling rules defined by the Chief Architect.

Basically the NAF Architectures are checked against the NAF Regulation supplied in the tool but the set rule can be extended to match the customer-defined modeling rules.

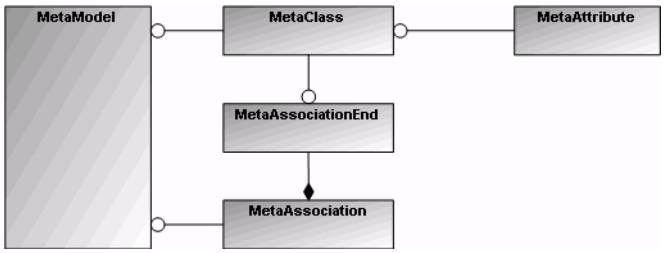


NAV-3a metamodel

NAV-3b Metadata (extensions)

The NAV-3b Metamodel is a top level definition allowing the support of metamodel definitions. This Metamodel allows the creation of classes, attributes and relationships that match the NAF Metamodel.

By default, the NAV-3b deliverables display the NAF Metamodel (NMM) implementation in the HOPEX tool while it is possible to only focus on extensions to add the relevant metamodels in the deliverables.

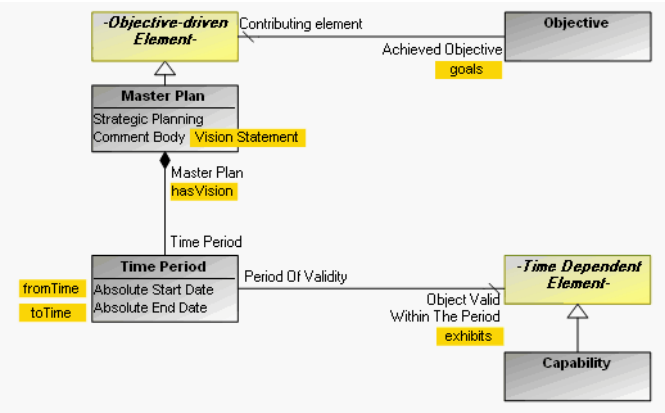


NAV-3b metamodel

CAPABILITY VIEWS METAMODELS

NCV-1 Capability Vision

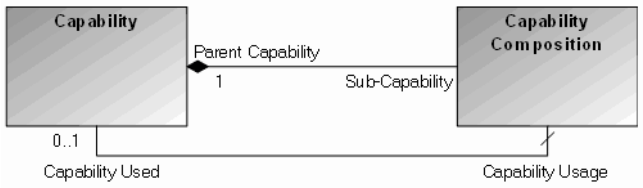
From the enterprise master plan defined in the NAV-1 metamodel, it is possible to refine the definition by inserting capability planning. Both the master plans and the milestones that divide enterprise life into sub-sections can be linked to goals that describe the expectations over a duration of time.



NCV-1 metamodel

NCV-2 Capability Taxonomy

NCV-2 defines capabilities and the composition of capabilities.

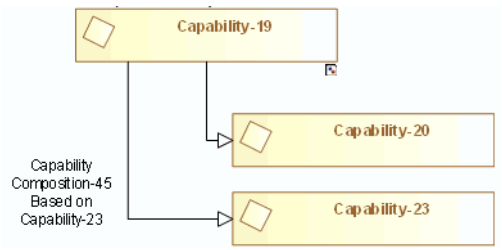


NCV-2 metamodel

Below is an example of a Capability Structure Diagram that describes a capability (Capability-19) that contains two capability compositions (12 and 45) based on other capabilities (respectively 20 and 23).

NCV-2 Example in a Capability Structure Diagram

In the example below, the same capability structure is drawn, however, this time in a tree diagram. By default, composition names are not displayed so that the drawing appears like a tree. It is, however, possible to force the display of composition names, if required (as done for composition-45).

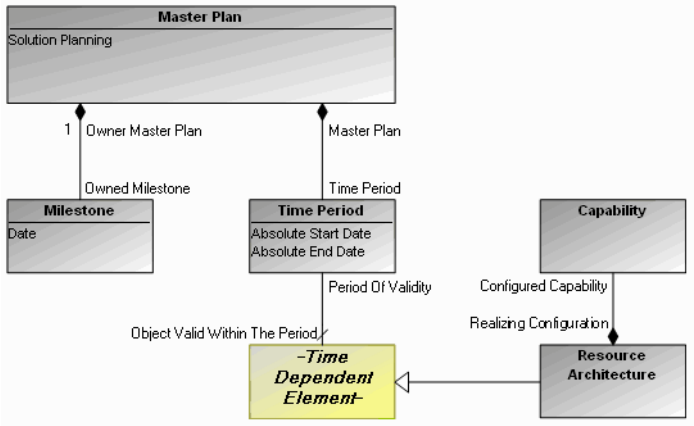


NCV-2 Example in a Capability Tree Diagram

NCV-3 Capability Phasing

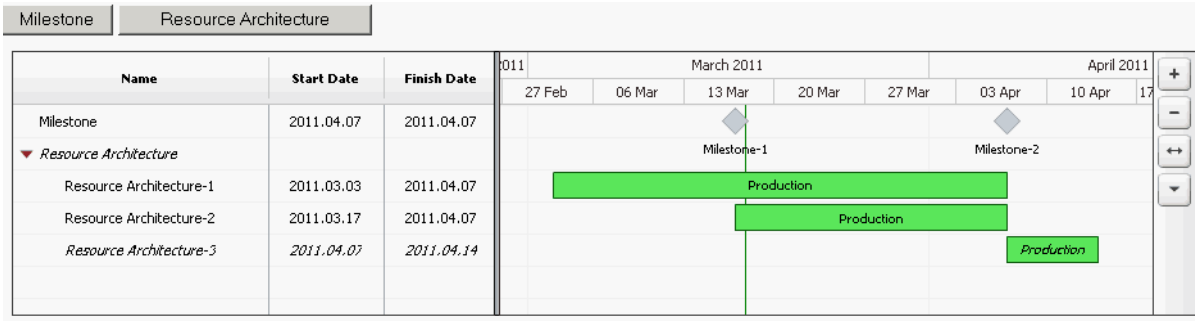
NCV-3 illustrates when solutions are implemented. Solutions are resource architectures that support a capability. The master plan defines the temporal views via milestones.

To be considered capability configurations the resource architectures must be linked to a capability.



NCV-3 metamodel

In the example below, the resource architectures are placed between milestones. This example represents the availability of the architectures. It is actually the time period of the architecture that is illustrated in the diagram.

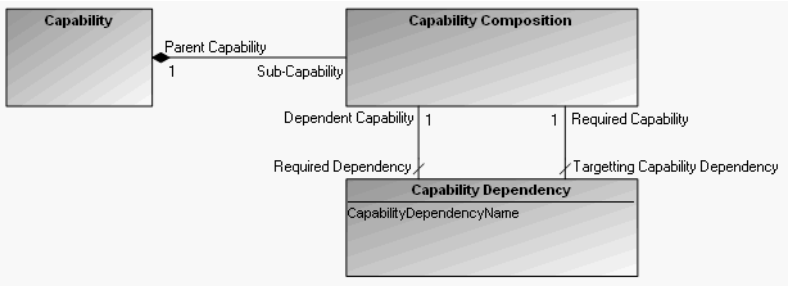


NCV-3 Example in a Master Plan Definition Diagram

NCV-4 Capability Dependencies

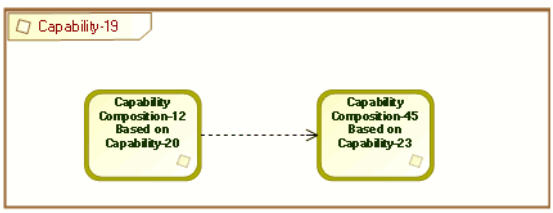
NCV-4 identifies the dependencies between capabilities. Dependencies are identified in relation to the context in which they are included and link the capability compositions that are defined within an upper capability. Two associations are required in order to set the source and target of each dependency.

The CapabilityDependencyName property is used to automatically find the name of the dependency that depends on the "container" capability and the capability source and target.



NCV-4 metamodel

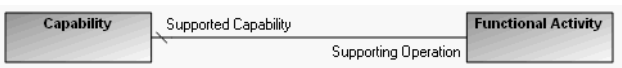
The NCV-4 and NCV-2 examples are similar as they rely on the same kind of diagram (Capability Structure Diagram). In the NCV-4 example, capability dependencies are represented by dashed arrows.



NCV-4 Example in the Capability Structure Diagram

NCV-6 Capability of Operational Activities Mapping

This Metamodel enables the possible of making links between capabilities (NCV-2) and functional activities (NOV-5).



NCV-7 Capability to Services Mapping

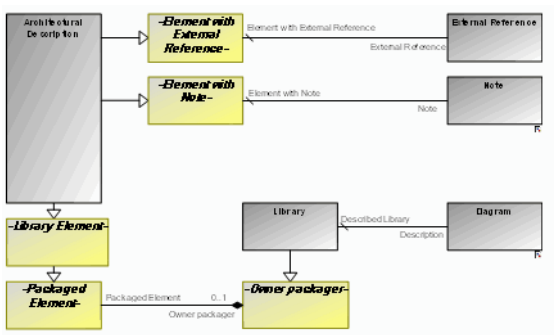
This Metamodel makes the link between capability and service to express the support by the services of the associated capabilities. When a service supports a capability, this relationship is illustrated with a direct link between the two concepts.

NCV-7 Metamodel

OPERATIONAL VIEWS METAMODELS

NOV-1 High-Level Operational Concept Description

NOV-1 is used to create high-level descriptions of architectures. Graphical descriptions are illustrated through diagrams that include items of the library that contains the architecture. Additional information can be attached to the architecture by way of external references and notes.

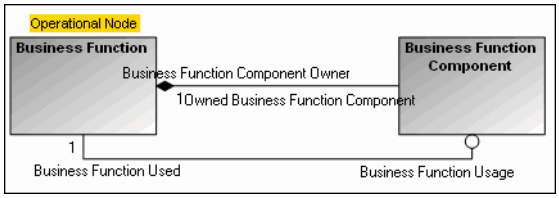


NOV-1 metamodel

NOV-2 Operational Node Connectivity Description

NOV-2 describes the operational nodes of architectures. Each node can be subdivided into different parts that are modeled as Operational Components. These subdivisions are typed by sub-operational nodes.

In **HOPEX** operational nodes are called Business Functions.

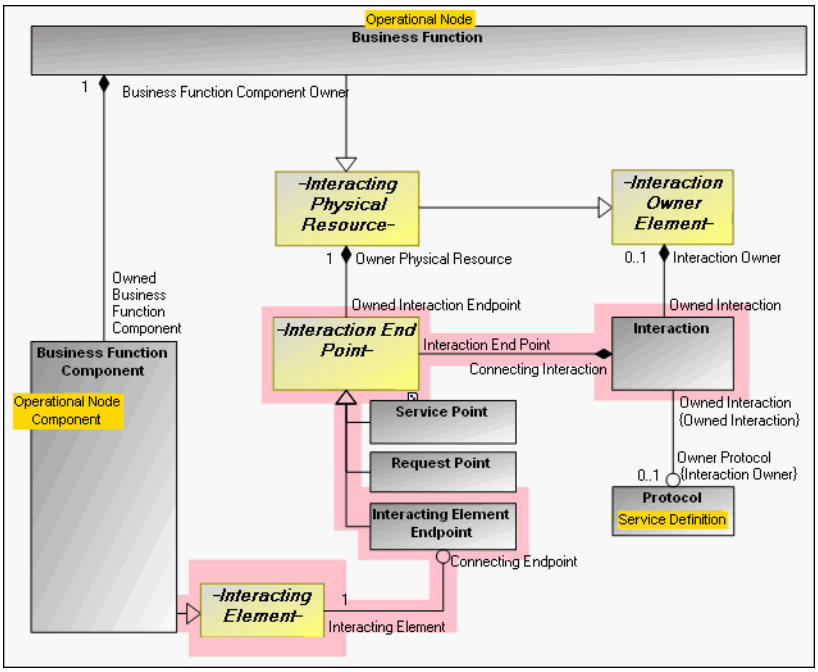


NOV-2 Structure metamodel

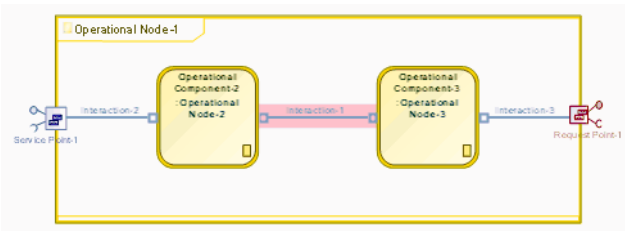
In the example below, one operational node (1) contains two operational components. These components reference sub-operational nodes (2 and 3).

NOV-2 Structure example

The NOV-2 metamodel is useful in defining the connectivity between operational components via interactions.



NOV-2 Metamodel for Interactions



NOV-2 Interaction Example

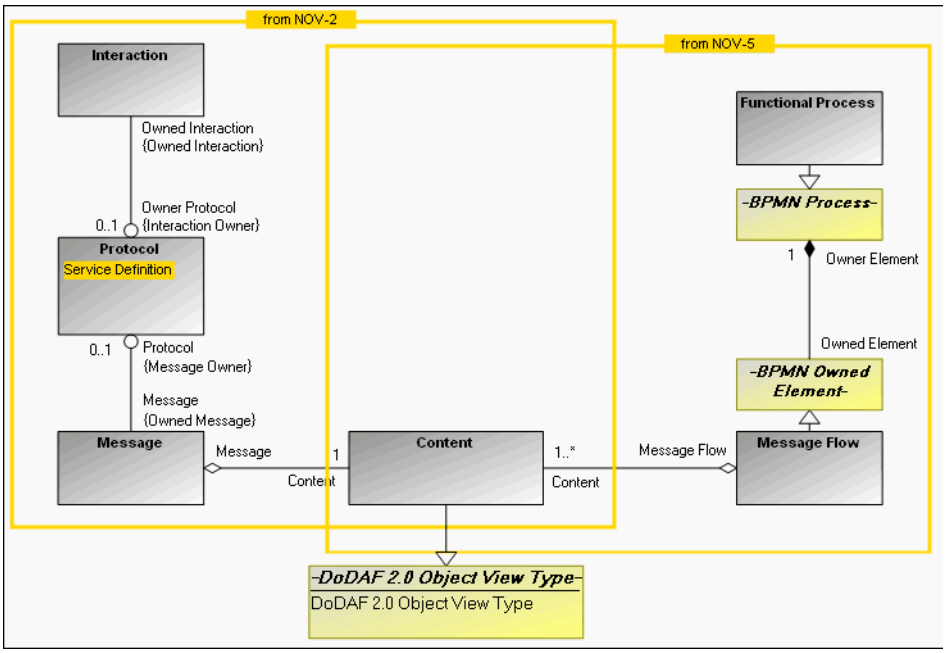
NOV-3 Operational Information Requirements

The NOV-3 view shows the matrix of the contents developed from the NOV-2 and those defined in the NOV-5 views.

The NOV-2 view defines interactions having protocol definitions. The definitions are refined by messages exchanged between different protocol roles. Each message carries an information element; the content.

The NOV-5 view describes the functional behaviors of the operational nodes. From the activities performed by these nodes result some messages (modeled by BPMN message flows that also carry information elements.

One NOV-3 report shows a matrix of the shared content between the NOV-2 and NOV-5 views.

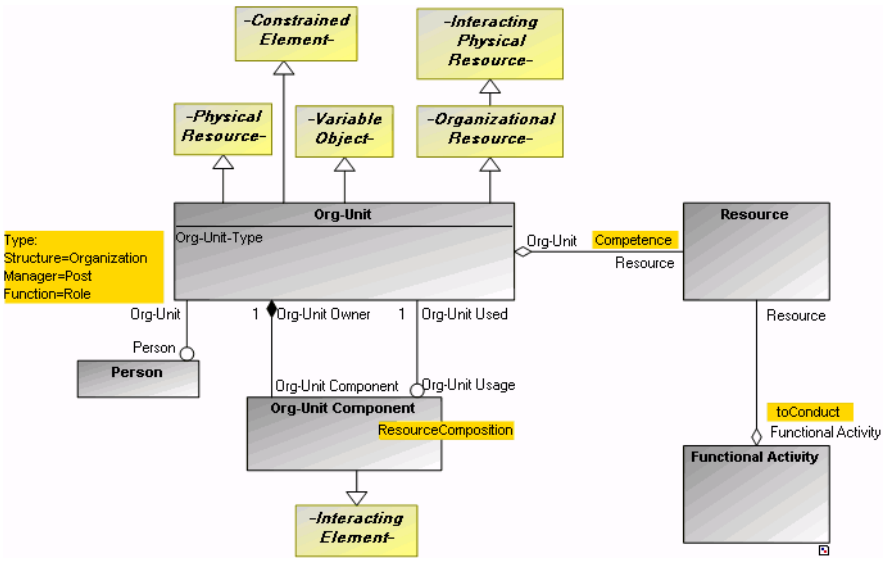


It is also possible to create content directly from the NOV-3 view. In this case, the content is stamped at the operational level thanks to the NAF Architecture View Type property.

NOV-4 Organizational Relationships Chart

Organizational units are modeled thanks to the Org-Unit metaclass. In the NAF metamodel implementation, org-units are composed via a whole-part model that

means there are contextual intermediate items at each hierarchy level to bring the contextual dependencies (in this case information interactions).



Three types of org-units can be used: Organization, Post and Role. These are types usually delivered by HOPEX with other names but these names appear if the NAF Metamodel Customization add-on has been imported.

Organizational-units inherit from several abstract notions that support the following features:

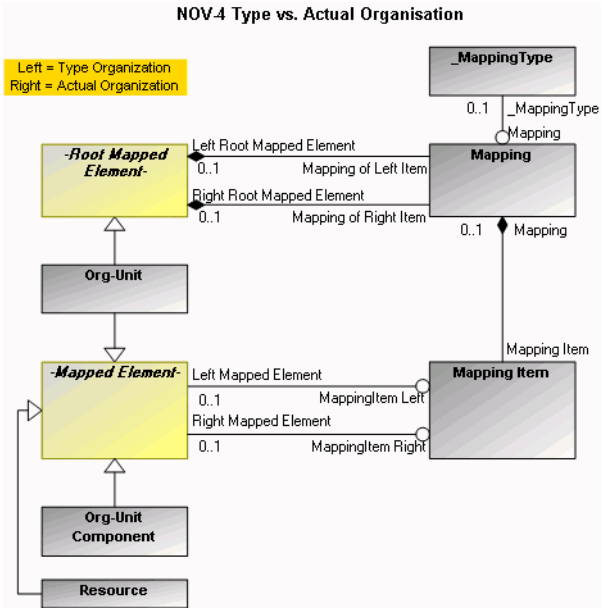
- **Variable Object**: enables the creation of variant of the organization-units. Each variant inherits from the components that can be replaced or excluded.
- **Constrained Element**: enables the adding of constraint to the organizational units. Constraints are defined in the
- **Physical Resource**: enables the connection of process performance items and interaction scenario performance items. These items are the links between the organizational structure and the processes or interaction scenarios that describe the behavior of the structure.
- **Organizational Resource**: enables the ownership of interactions and communication channels between the components of an organizational unit.

An organizational unit can also be connected to some resources that define the competencies he can develop.

Finally, organizational units are embodied by persons that have the role defined in the organizational structure.

The NOV-4 metamodel also introduces the notion of a mapping between organizations defined at a type level and those defined as the actual organizations. These mapping allows for example to associate generic definitions to concrete organizations deployed in different geographical sites.

The following figure shows the metamodel that supports this mapping. A Mapping Type defines the mappings that can be created. In this situation the mapping is between root organizational units. A notion of left and right mapped element is introduced that is a generic term to differentiate between both set of mapped items. Indeed, the mapping relationship is considered as an equivalence association and there are no preferred directions.



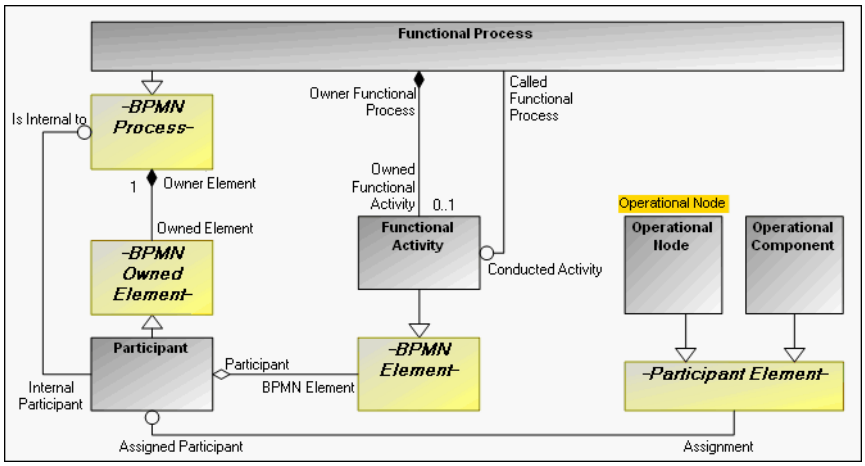
The Organizational Structure

Once a mapping is created, the sub items can be mapped. There are connected thanks to Mapping Items. In this mapping, the org-unit components and the resources (competences) can be mapped.

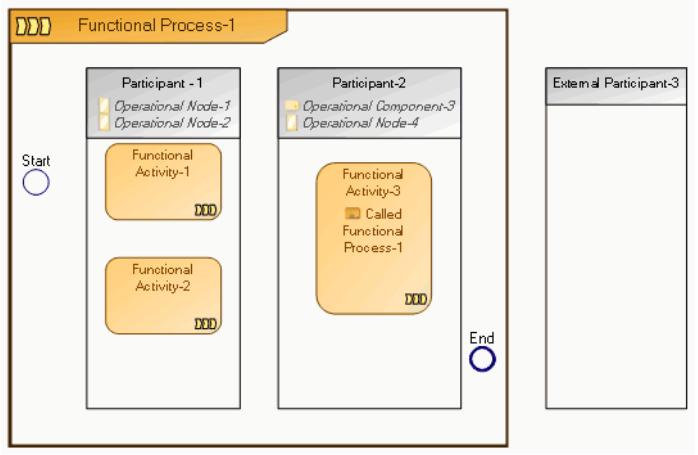
NOV-5 Operational Activity Model

Activity models are described through **HOPEX** functional processes. These models are composed of participants (roles involved in the process) and functional activities (tasks performed by participants). Functional processes describe scenarios of operational node structures. The participants can therefore be assigned by either operational nodes or operational components (the contextualized embodiment of an

operation node). Functional activities can rely on sub-processes (called functional processes).

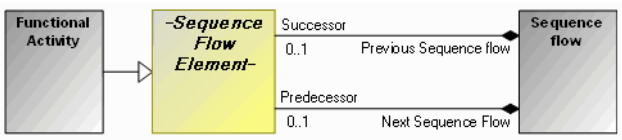


NOV-5 Structure Metamodel

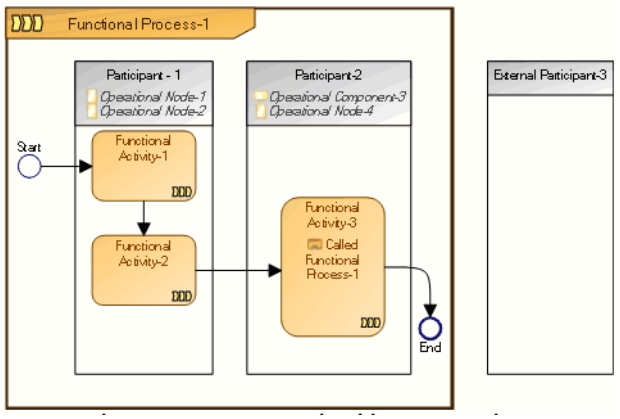


NOV-5 Sequence Flow metamodel

Each functional activity can be sequenced in a functional process via a Sequence Flow.



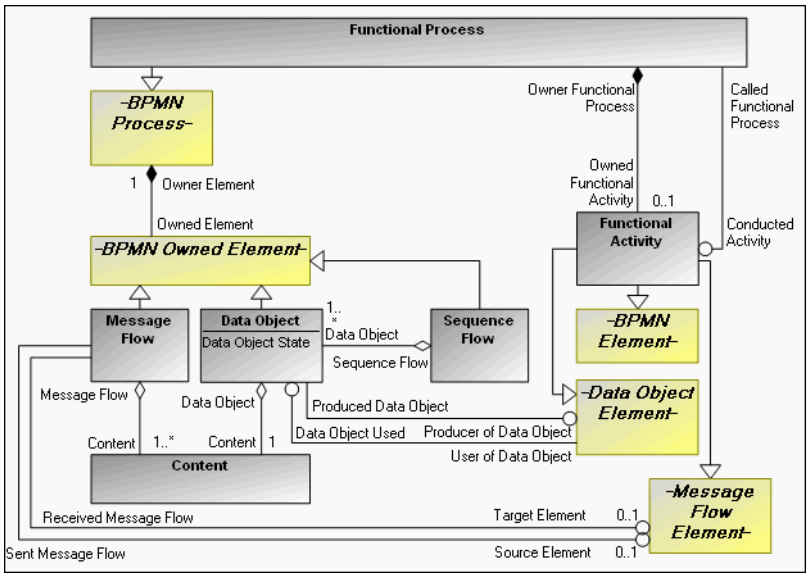
NOV-5 Sequence Flow metamodel



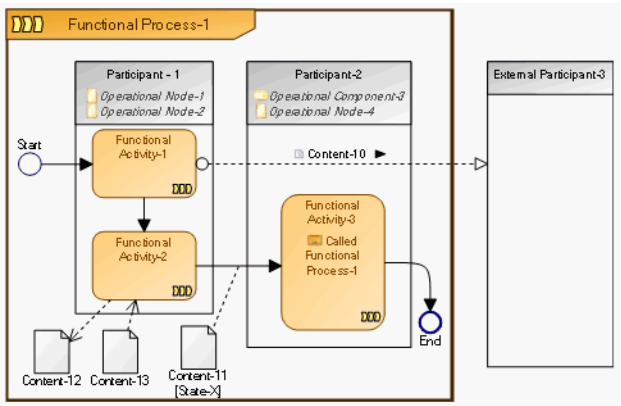
NOV-5 Example with Sequence Flow

Functional activities may exchange information through message flows. Information can also be retrieved from/created in data objects. Data objects attached to sequence flows are actually shortcuts to the source in the data object while the

target retrieves the information to perform the activity. Data objects can be defined with states.



NOV-5 Message Flow and Data Objects

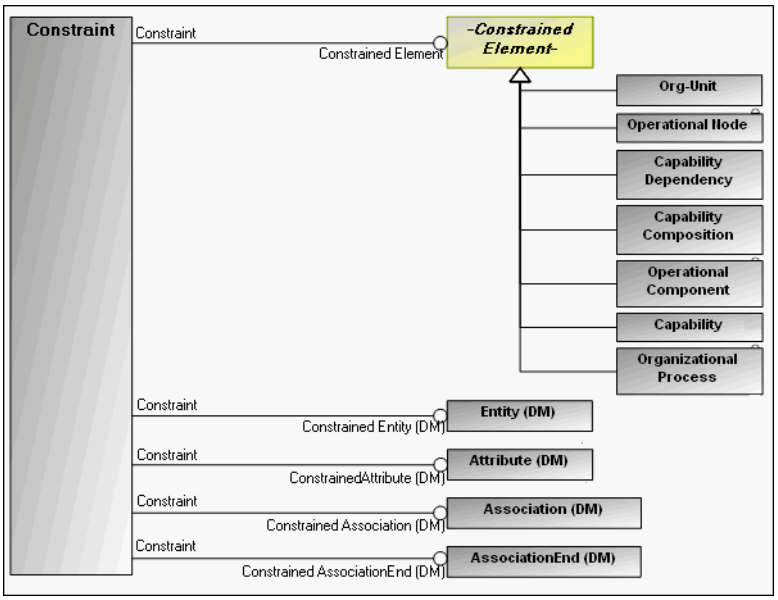


NOV-5 Structure with Sequence and Message

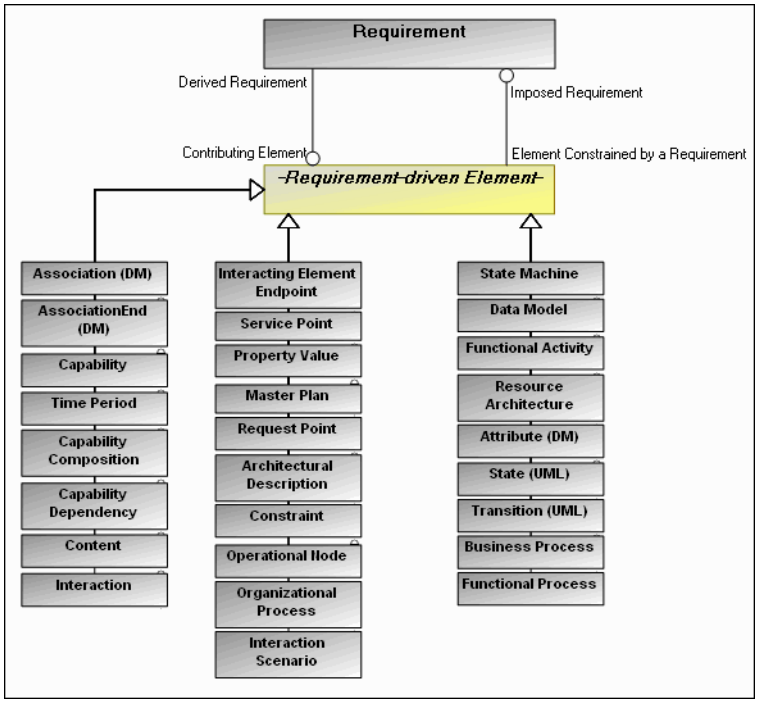
NOV-6a Operational Rule Model

Operational rules are modeled in relation to two notions: constraints that apply to the operational items as external restrictions to the potential means for the project achievement and requirements that are external requests at the capability level.

This Metamodel enables the linking of operational items to constraints and requirements in order to indicate what cannot be done and what must be fulfilled.

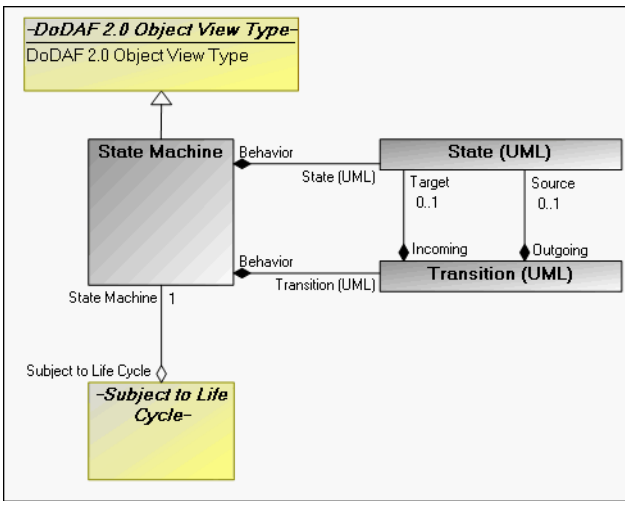


Constraints are Applied to Operational Items



NOV-6a Imposed and Derived Operational Requirements

NOV-6b Operational State Transition Description

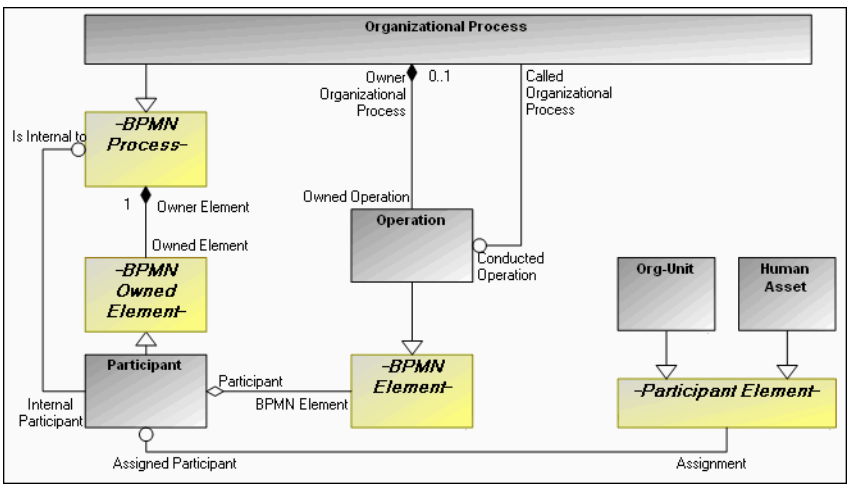


NOV-6b State and Transition Metamodel

NOV-6c Operational Event Trace Description

The NOV-6c metamodel can be divided into two parts: a process description and a specific scenario where generic roles are instantiated.

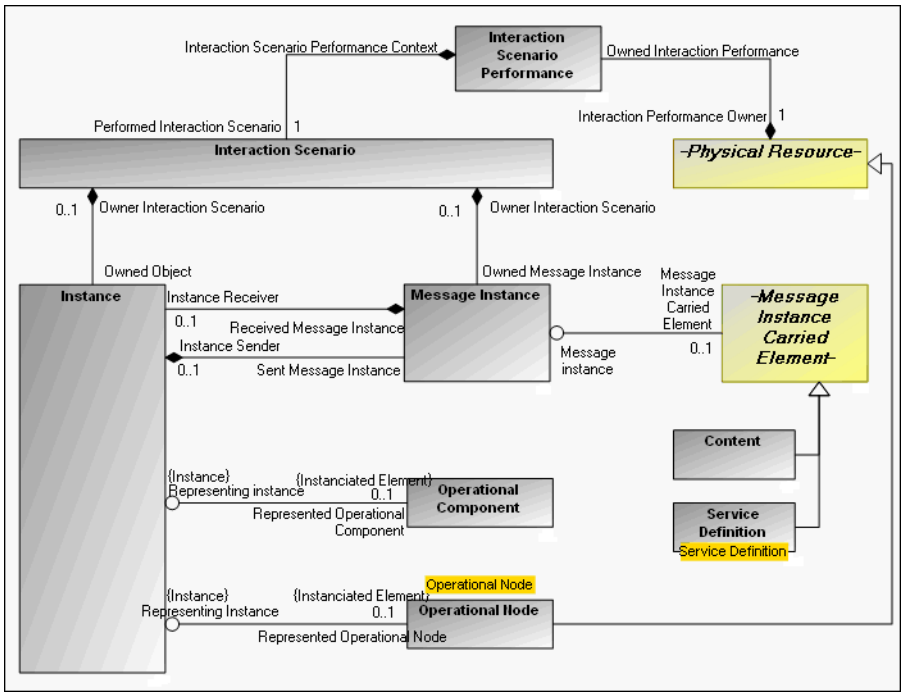
The process metamodel is similar to the NOV-5 metamodel. NOV-6c is BPMN-compliant, however, for this view participants are assigned to org-units and human assets (org-units defined within the context of a resource architecture).



NOV-6c Organization Process Metamodel

Another alternative for the event-trace description is the operational interaction scenario that allows the creation of operational node instances (or more accurately operational components) and message instances.

Interaction scenarios describe examples of processes through the interaction scenario performances that link functional processes to operational nodes.

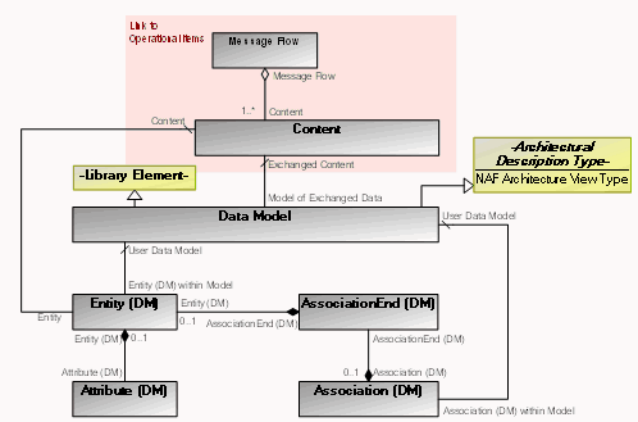


NOV-6c Operational Interaction Scenario

NOV-7 Information Model

The NOV-7 metamodel describes the data model and the entities and associations that formalize the information transported between operational items. This metamodel is based on a common entity-association metamodel. The link to the operational model is established through the content transported by the messages.

The NAF Architecture View Type property is set to Operational to identify system data models.



NOV-7 Data Model Metamodel

SERVICE-ORIENTED VIEWS METAMODELS

NSOV-1 Service Taxonomy

This metamodel supports the definition of services. The Business Services is so named to establish the distinction with the IT services. Even though business services can be supported by the IT, at this level the metamodel addressed the business perspective of the services and not the technical aspects.

A category property enables the classification of NAF Services (operational, information, application).

Services can be derived from existing objects to focus on the differences instead of replicating all the existing sub-items (variable object).

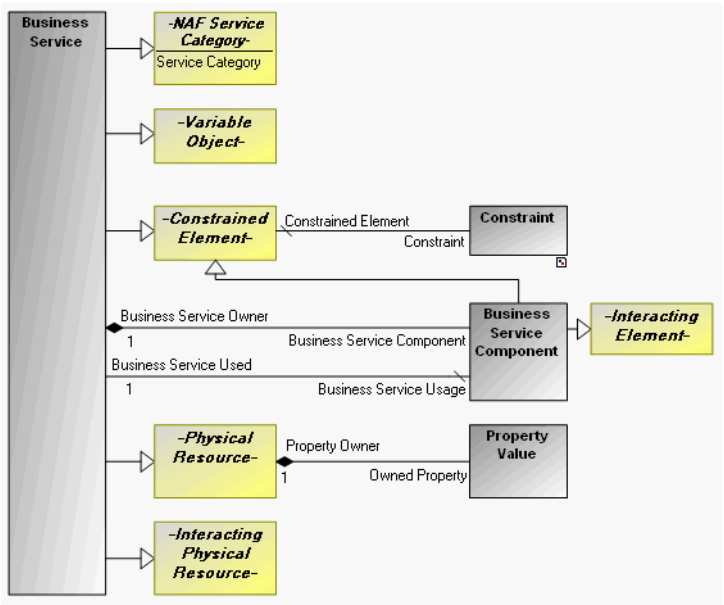
A service can be linked to the constraints that must be supported.

A service can be decomposed into sub-services (business service composition).

As an interacting element the business service compositions can be linked to each other through interactions that express the data exchanged.

As an Interacting Physical Resource, the business service can expose services and request points that match the accessible service interfaces.

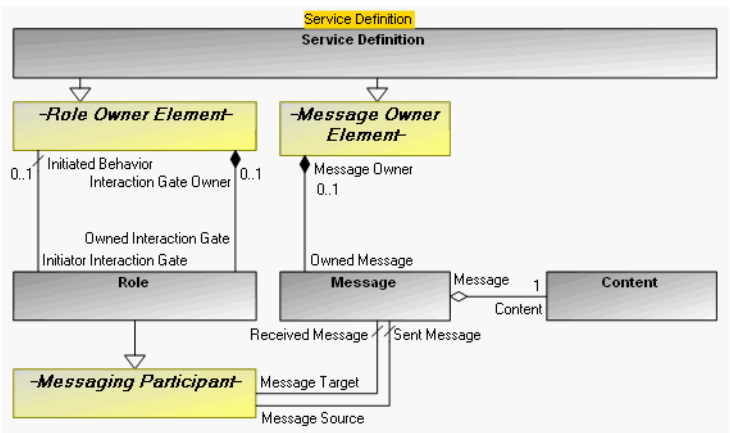
Finally, important property values can be added to emphasize some characteristics of the services.



NSOV-1 Metamodel

NSOV-2 Service Definitions

Service definitions rely on the Protocol notion in **HOPEX**. Each protocol defines two or more roles that are involved in the service. One role corresponds to consumers while the other corresponds to providers. The consumer is connected to the service definition by the initiation association.



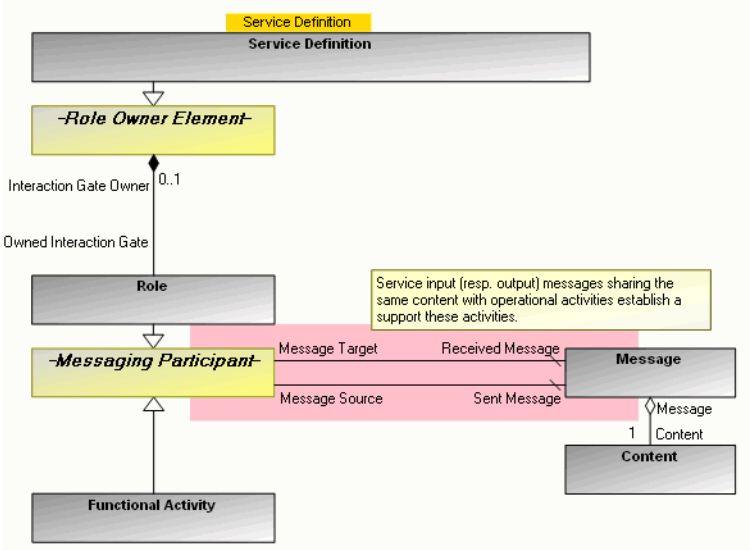
NSOV-2 Metamodel for Service Definition

NSOV-3 Services to Operational Activities Mapping

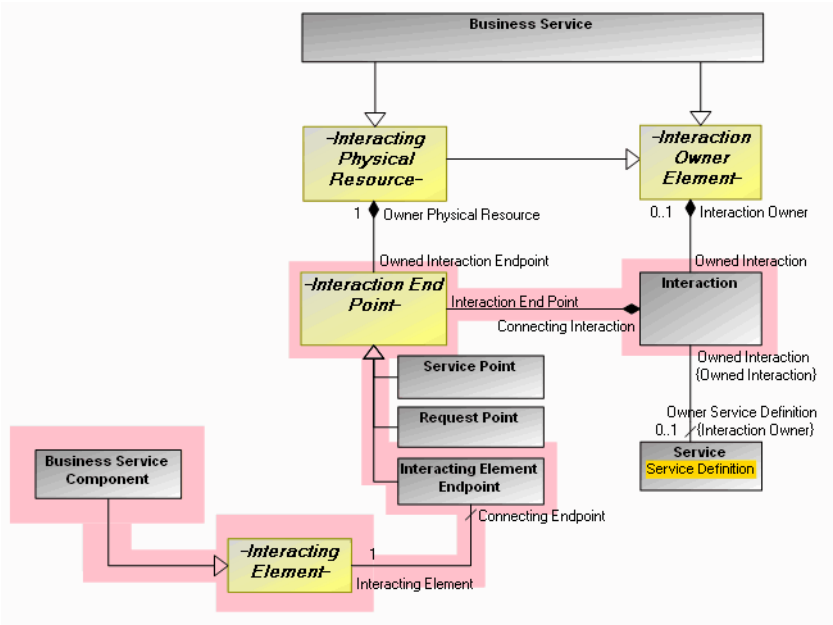
This metamodel adds the link that describes the support of functional activities through the service.

NSOV-3 Metamodel

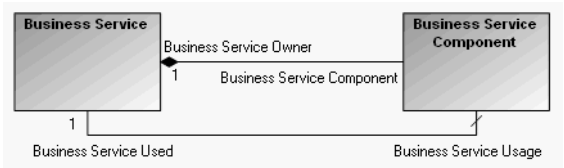
NSOV-4 Service Orchestration



NSOV-4 Metamodel of Service Definition that Supports Functional Activities



NSOV-4 metamodel for Service Interactions



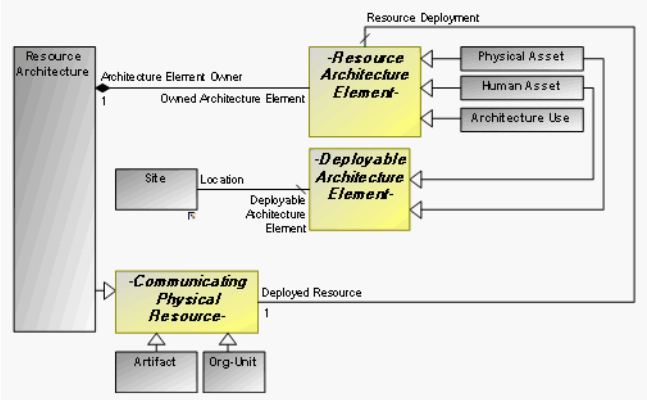
NSOV-4 Metamodel for Service Structure

SYSTEM VIEWS METAMODELS

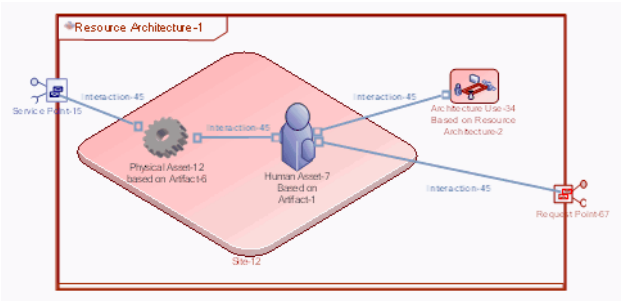
NSV-1 System Interface Description

The NSV-1 metamodel is used to describe resource architectures, artifacts and applications. Resource architectures and artifacts are described using a component structure approach. Resource architectures are composed of physical assets (typed by artifacts), human assets (typed by org-units) and architecture uses (typed by

resource architecture). The objects are typed through the Deployed Resource association. Only physical assets and human assets can be localized (linked to Site).



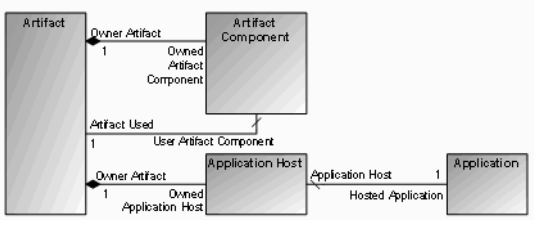
NSV-1 Metamodel for Resource Architectures



NSV-1 Example Including All Kind of Objects in a Resource Architecture

The metamodel that describes artifacts is similar to the resource architecture metamodel. The items included in this metamodel are artifact components (Artifact type) and application host (Application type).

Artifacts are deployed within the context of resource architectures. Their items are deployed in the same manner therefore there are no sites in this metamodel.



NSV-1 Metamodel for Artifact Structure

The example below is of an artifact that contains a sub-artifact (via an Artifact Component) and an application (via an Application Host).

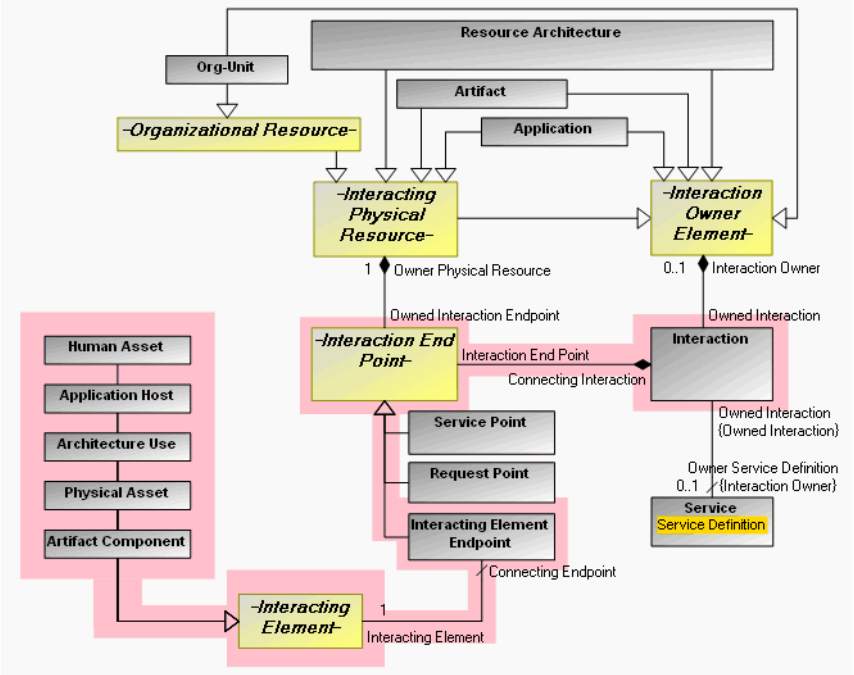
NSV-1 Example Showing an Artifact Assembly Diagram

System Interface

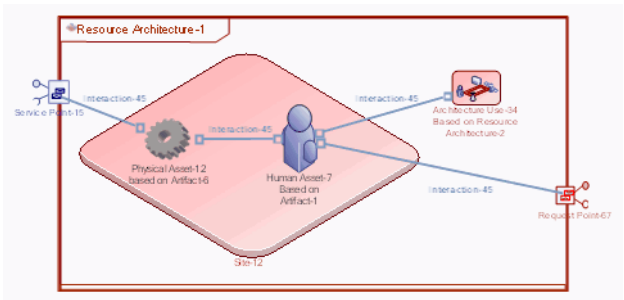
The interfaces between NSV-1 items are described via service and request points. Information is exchanged via interaction-connected service definitions (protocol in HOPEX vocabulary). The NSV-1 metamodel is similar to the NOV-2 metamodel.

In the figure below, the pink highlighted sections indicate the notions used to model interactions.

The NAF Architecture View Type property is set to System to identify system data models.



NSV-1 Metamodel for Interactions



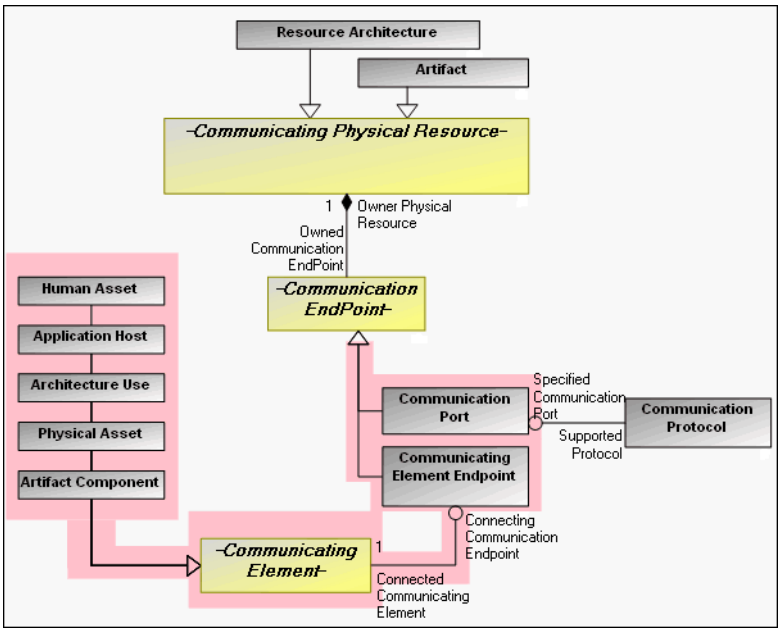
NSV-1 Interaction Example

NSV-2a System Port Specification

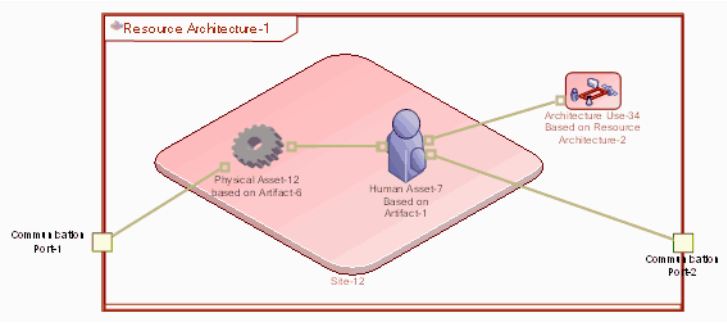
In the NSV-1 metamodel, information exchanges are modeled through interactions and service definitions as in the NOV-2 metamodel. It is also possible to model the means set to support interactions that are not within the scope of an operational view. This information is modeled in the NSV-2a view.

The introduced notions are similar to interactions; these notions are communication ports, communication channels and communication protocols.

The pink highlighted sections in the metamodel indicate the relationships used to create a communication path which is composed of two communication end points (either a communication port or any communicating element endpoint).



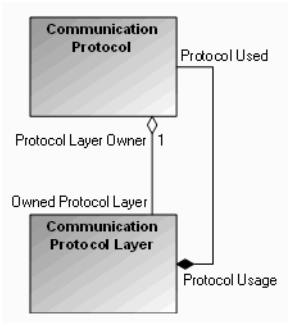
NSV-2a Metamodel for Communication Means



NSV-2 Exchange Means Example

Communication Protocols

Communication protocols can be decomposed into layers, which are then detailed via sub-protocols (one or more per layer). The corresponding metamodel is indicated below.



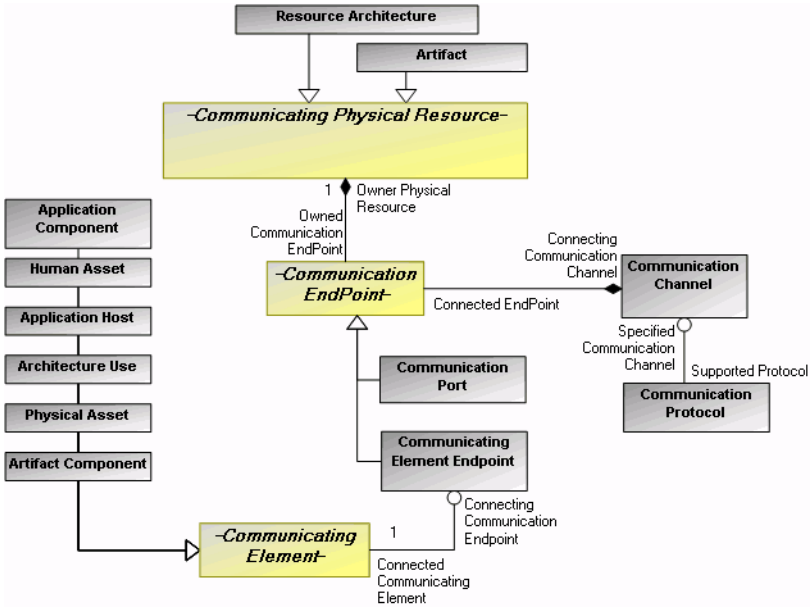
NSV-2a Metamodel for Communication Protocols

NSV2b System Port Connectivity

This view is dedicated to the communication channels that connect either the communication port defined for a artifact or a resource architecture, or the communicating elements included in an artifact or in a resource architecture.

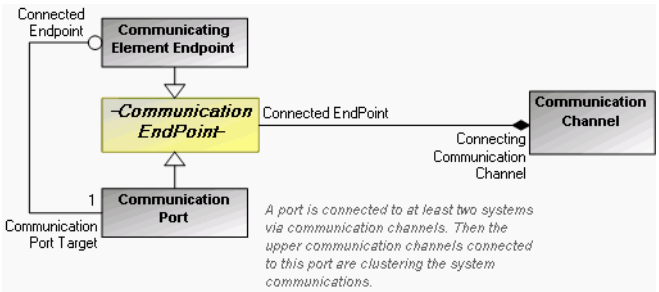
Communicating elements can be:

- Human assets: typed by org-units in a resource architecture,
- Application hosts: typed by applications in an artifact,
- Application components: typed by applications in an upper application,
- Architecture uses: typed by resource architecture in a resource architecture,
- Physical assets: typed by artifact in a resource architecture,
- Artifact components: typed by artifact in an upper artifact.



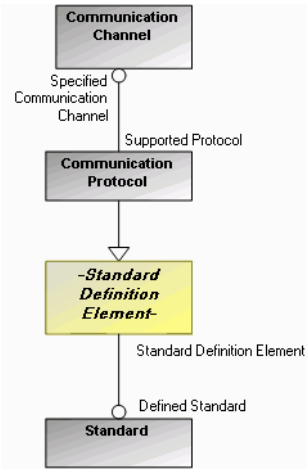
NSV-2c System Connectivity Clusters

This view collects all the communication channels connected to every communication port that dispatches the information elements to more than one sub-system.

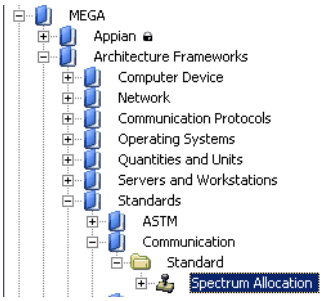


NSV-2d System Communication Quality Requirements

This view shows the communication channel based on protocols that match the Spectrum Allocation standard.



The ***Spectrum Allocation*** standard can be found importing the HOPEX Architecture add-on into the current repository. The protocol used in a communication channel is then linked as a definition element of this standard.



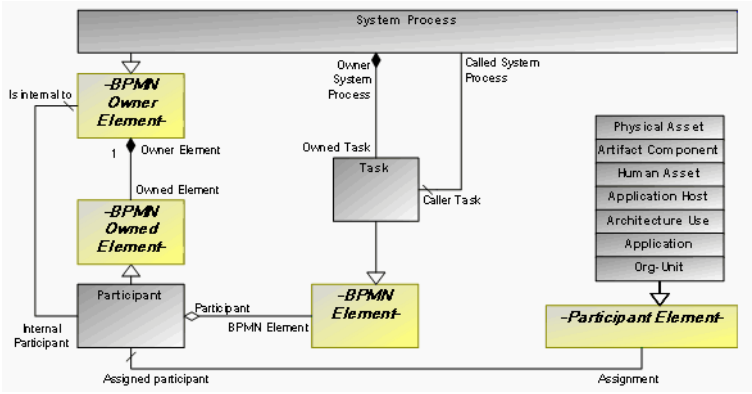
NSV-3 Resource Interaction Matrix

This view shows two matrices between communicating elements: one matrix with a checkmark in the cells when a communication channel exists between the row and the column item, one other matrix with a checkmark in the cells when an interaction exists between the row and the column item.

NSV-4 System Functionality Description

The NSV-4 metamodel is similar to the process metamodel used for NOV-5 and NOV-6a. In this metamodel, participants can be assigned by any of the items modeled in the NSV-1 view. This assignment can be done at the general level (Application, Artifact, Resource Architecture) or at a more accurate level within the context of an architecture (Application Host, Physical Asset, Architecture Use).

Note: Since the organizational part defined in the NOV-4 view is included in system models, both org-units (general level) and Human Assets (contextual level) can be assigned to participations.



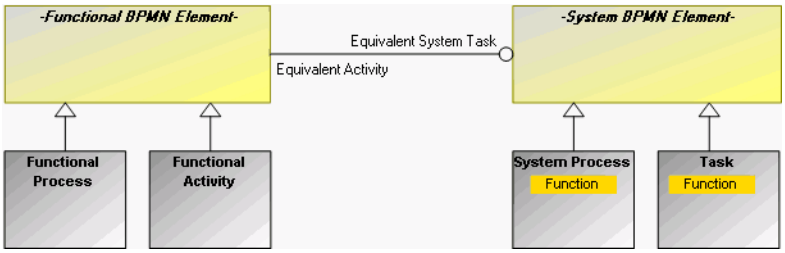
NSV-4 Metamodel Describing System Processes

NSV-5 System Function to Operational Activity Traceability Matrix

This matrix allows to link functional behavioral elements (functional processes or functional activities) to system behavioral elements (system processes or tasks).

In both the functional and system scopes, there are some processes that can be assessed and hierarchically organized. Such organizations are obtained either using the inheritance mechanism (variation) or the composition mechanism. In the last situation, components are respectively functional activities and tasks.

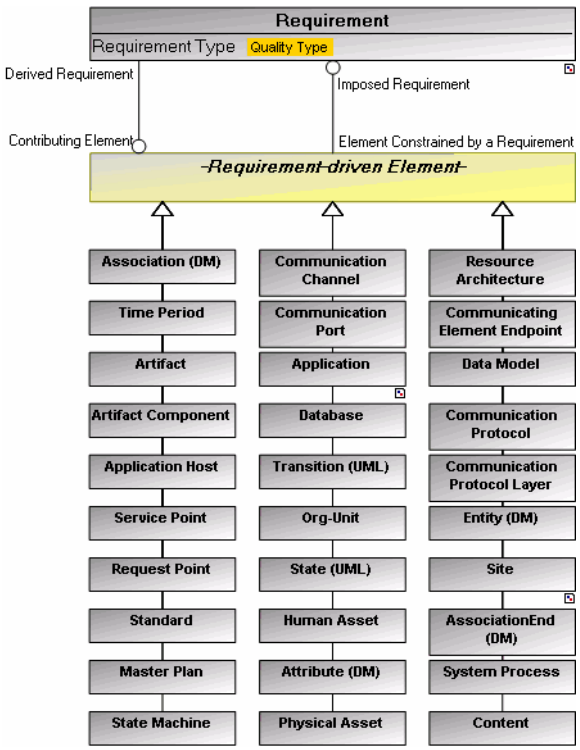
Thus, it is possible to link the independent behavioral elements (processes) or the contextualized elements corresponding to those including in a process (activities and tasks).



NSV-7 System Quality Requirements Description

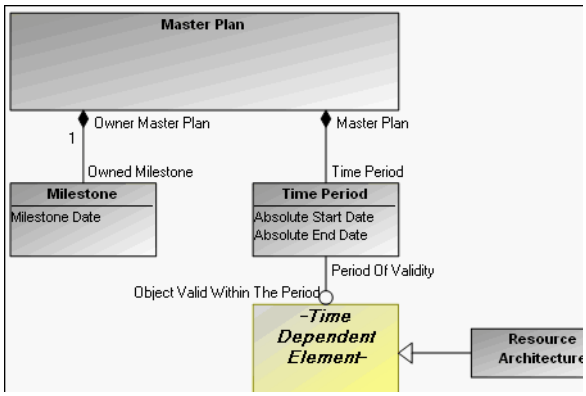
This view assesses all the system requirements (those having the View Type property set to "System") and having the Requirement Type property set to "Quality".

These requirements can constrain all the architecture elements defined in the system views.



NSV-8 Systems Configuration Management

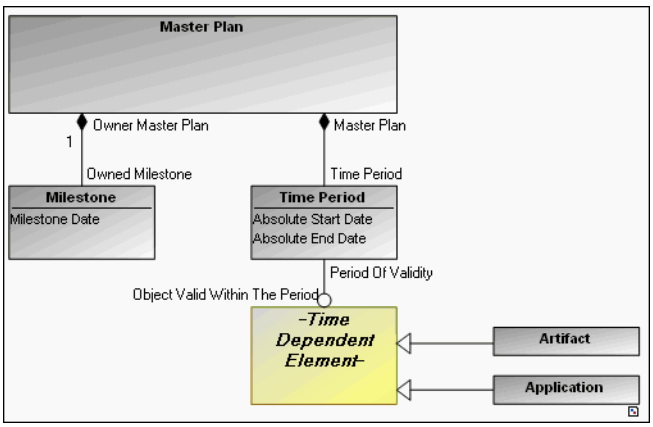
The NSV-8 metamodel can be used to plan systems (resource architectures connected, or not, to a capability).



NSV-8 Metamodel

NSV-9 Technology Forecast

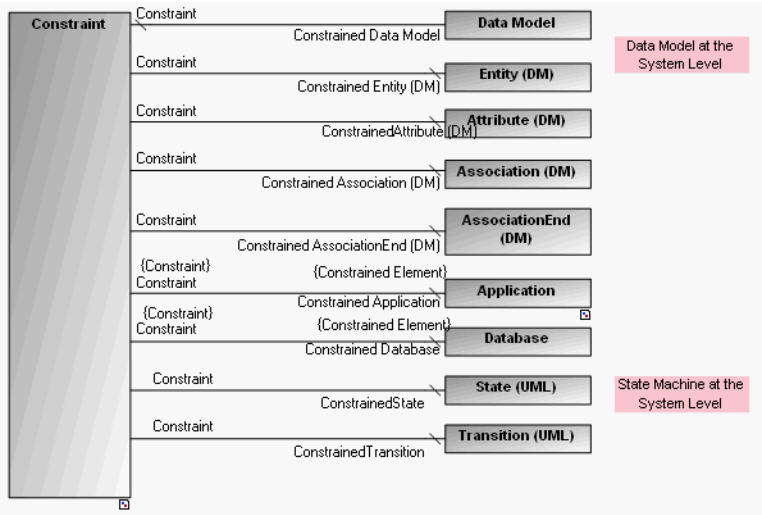
The NSV-9 metamodel is similar to the NSV-8 metamodel, however, it can be used to plan technology item (artifacts). To plan architecture items, the Infrastructure Planning property of the master plan must be activated.



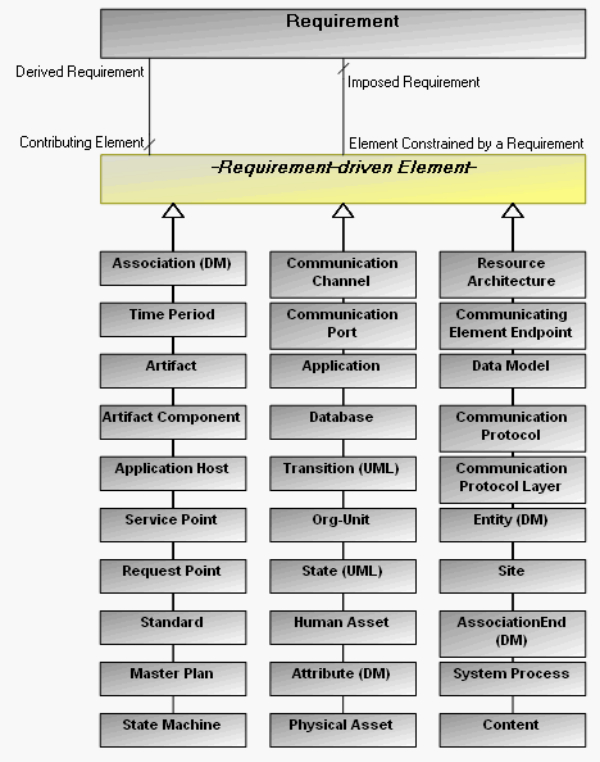
NSV-9 Metamodel

NSV-10a Systems Rule Model

System rules are modeled in relation to two notions: constraints that apply to the system items as external restrictions to the potential means for the project achievement and requirements that are external requests at the capability level.



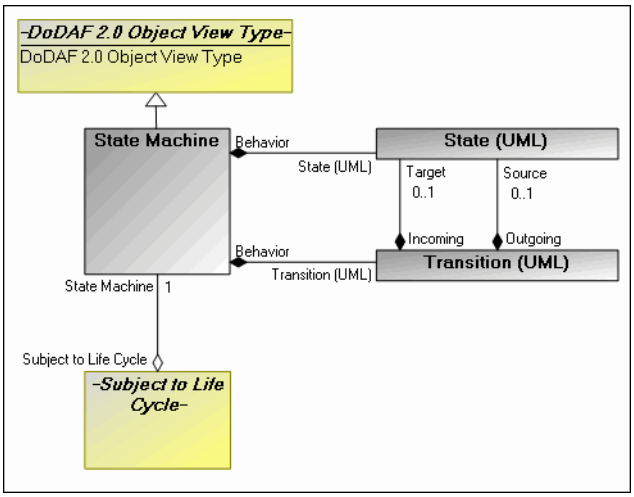
NSV-10a Metamodel for Constraints



NSV-10a Metamodel for Requirements

NSV-10b Resources State Transition Description

The NSV-10b Metamodel is similar to the NOV-6b Metamodel. In this Metamodel the View Type property is set to System to categorize the behavior at the system level.

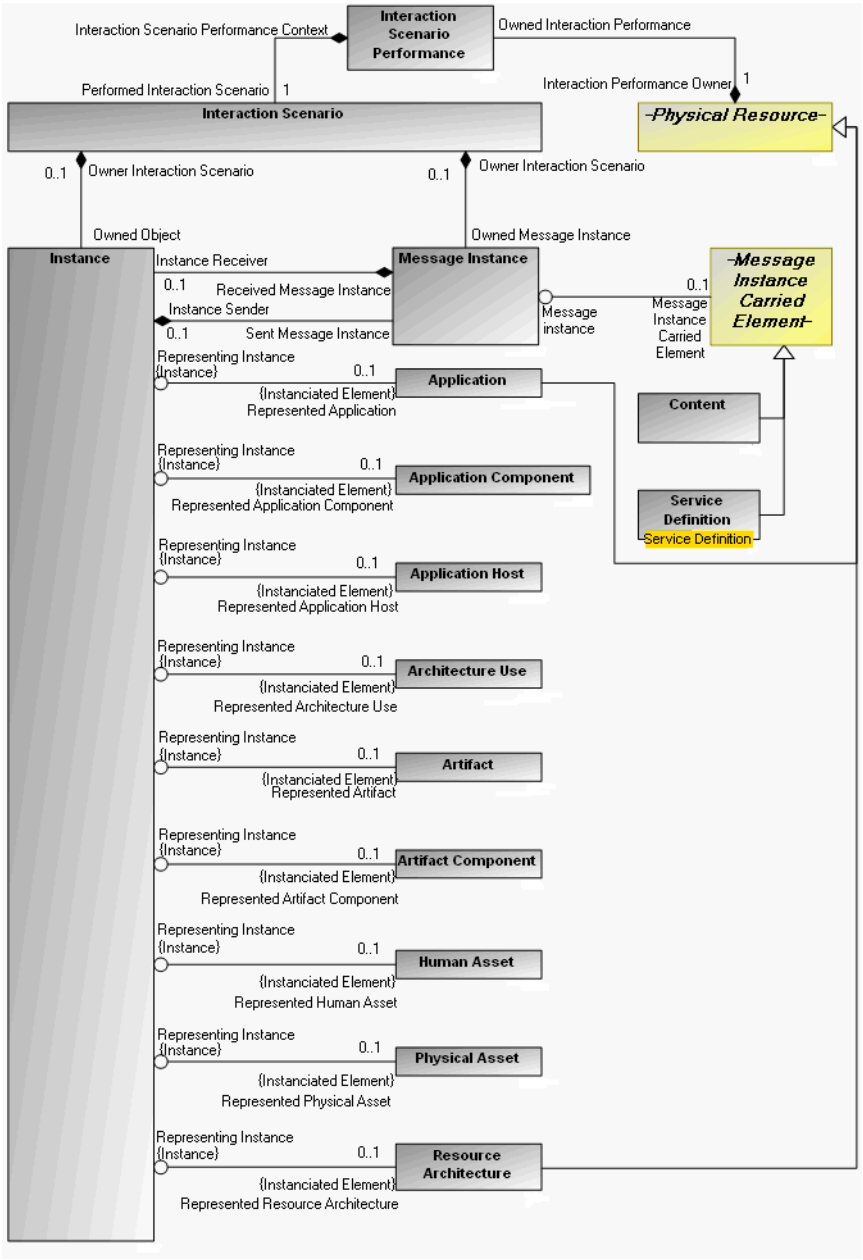


NSV-10b Metamodel

NSV-10c Resources Event-Trace Description

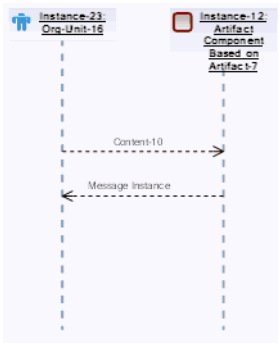
NSV-10c Metamodel supports the time-ordered description of events in a system. This is supported by the interaction scenario concept that contains instances and message instances.

An interaction scenario describes an example of a process through the interaction scenario performance that links a system process to a resource architecture.



NSV-10c Metamodel

In the example below, the first message instance is based on content (content-10) while the second message instance is not yet specified.

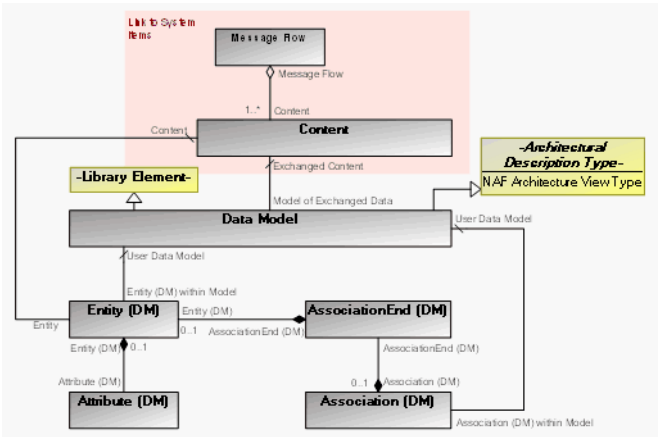


A System Interaction Scenario Diagram

NSV-11a Logical Data Model

The logical data model metamodel is similar to the information model metamodel defined in the NOV-7 view. The sole difference is that in the logical data model metamodel the message flow connected to the content must exchange information between system items.

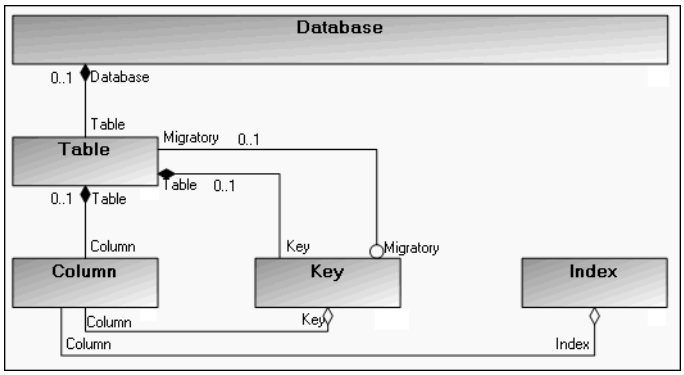
In this metamodel, the NAF Architecture View Type property is set to System to identify system data models.



NSV-11a Metamodel

NSV-11b Physical Data Model

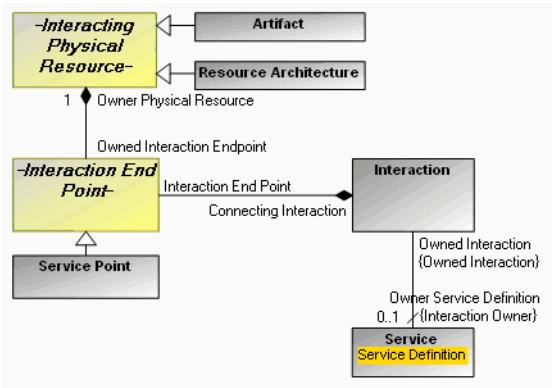
The NSV-11b metamodel supports the creation of databases with tables, columns, keys and indexes.



NSV-11b Metamodel

NSV-12 Service Provision

The NSV-12 metamodel is based on notions previously introduced and established in the matrix of service definitions linked to either resource architectures or artifacts that support an expected service.

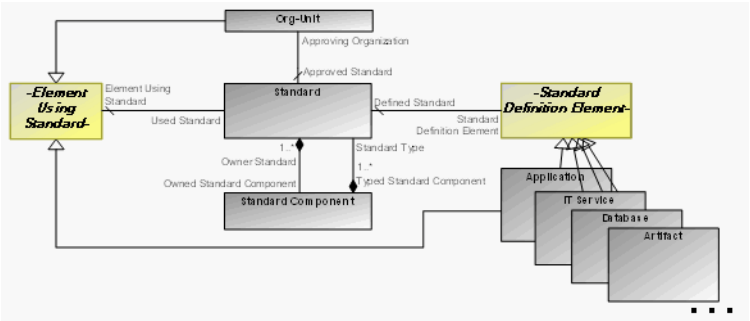


NSV-12 Metamodel

TECHNICAL VIEWS METAMODELS

NTV-1 Standards Profile

The NTV-1 metamodel supports the description of standards and can be used to link standards to the items by which they are defined. At the opposite end of the metamodel, standards can be linked to the elements that use them.

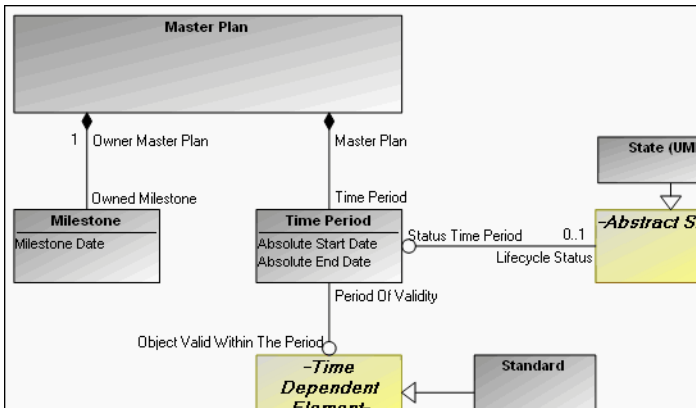


The NTV-1 Metamodel

NTV-2 Standards Forecast

The NTV-2 view is based on master plans that can be used to plan and forecast. As standards are time-dependent objects time periods can be linked to them and placed into master plans of standards. The time periods represent moments in the

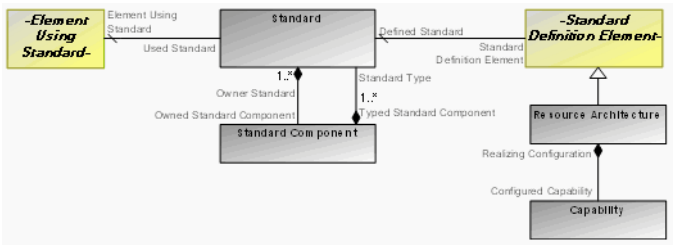
"lifetime" of the standard and so they can be linked to states (UML states) that are appropriate for the standard objects.



NTV-2 Metamodel

NTV-3 Standards Configuration

The NTV-3 view gives a more detailed illustration of how standards are configured. The NAF advice is to use the models established in the NSV-1 view for these illustrations. The **HOPEX** implementation therefore relies on the resource architecture linked to the standards as defining elements.

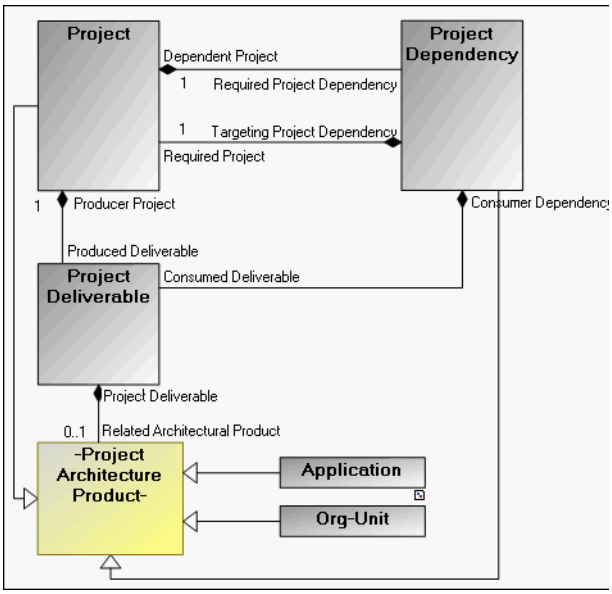


NTV-3 Metamodel

PROGRAMME VIEWS METAMODELS

NPV-1 Programme Portfolio Relationships

The NPV-1 view details the projects that support the design and implementation of architectures. Projects are dependent on each other (Project Dependency). They produce deliverables that are embodied by HOPEX objects such as applications and org-units.

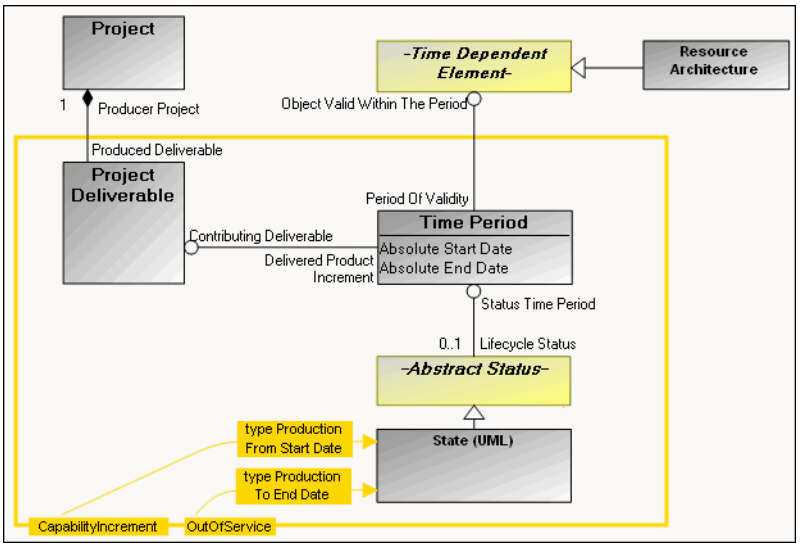


NPV-1 Metamodel

NPV-2 Programme to Capability Mapping

The NPV-2 metamodel makes the link between the project deliverables and periods of a resource architecture (NSV-1) defined to support a capability (NCV-2) and

planned in a solution master plan (NCV-3). A period can be linked to a state that matches one of the available resource architecture states.



NPV-2 Metamodel

METAMODEL RENAMING

Customer feedback have indicated that a difficulty exists in using the NAF application with a mixture of both NAF and **HOPEX** vocabulary. This difficulty is visible in the NAF navigation tree which uses NAF vocabulary while created objects have **HOPEX** metaclass names.

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.

HOPEX METAMODEL RENAMING

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	Initial HOPEX Definition	NAF Definition
Business Function	Operational Node	A business function is a skill or grouping of skills of interest for the enterprise.	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 business functions. This concept enables assembly under the same root of business functions around a common theme when this theme cannot be explained in the method.	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.
Protocol	Service Definition		