HOPEX UAF

User Guide

HOPEX Aquila

Bizzdesign

CONTENTS

Contents

INTRODUCTION TO HOPEX UAF

The Unified Architecture Framework (UAF) is based on the Unified Profile for DoDAF and MODAF™ (UPDM™). UAF defines ways of representing an enterprise architecture that enables stakeholders to focus on specific areas of interest in the enterprise while retaining sight of the big picture.

UAF meets the specific business, operational and systems-of-systems integration needs of commercial and industrial enterprises as well as the U.S. Department of Defense (DoD), the UK Ministry of Defence (MOD), the North Atlantic Treaty Organization (NATO) and other defense organizations.

UAF was initially developed as UPDM 3.0 in response to needs from the UML®/SysML® and military communities to develop standardized and consistent enterprise architectures based on the U.S. Department of Defense Architecture Framework (DoDAF) and the UK Ministry of Defence Architecture Framework (MODAF). Requirements were derived from both military frameworks as well as the NAF (NATO Architecture Framework). When these requirements were combined with requirements from the business sector (because 90% of concepts and themes captured in the military frameworks are equally applicable in the commercial domains), UAF, as a commercial framework that supports the needs of the commercial sector as well as the military, was born. Participants included a broad spectrum of interested parties, covering industry, tool vendors, and end users as well as representatives of the DoD and MOD.

For detailed guidance, see the official UAF web site.

HOPEX UAF supports the **UAF Specification 1.2 version**.

HOPEX UAF enables to generate UAF deliverables. It is based on the standard features of the **HOPEX** modeling tool. However, dedicated features have been introduced to guide the UAF expert to use the product with the vocabulary with which he/she is accustomed.

The points covered in **HOPEX UAF** guide:

- ✓ HOPEX Implementation of UAF;
- ✓ Strategic Viewpoint;
- ✓ Operational Viewpoint;
- ✓ Services Viewpoint;
- ✓ Personnel Viewpoint;
- ✓ Resources Viewpoint;
- ✓ Security Viewpoint;
- ✓ Projects Viewpoint;
- ✓ Standards Viewpoint;
- ✓ The UAF Terminology.

For more details on the interface and functions of **HOPEX** in general, see:

- ✓ Presentation of HOPEX UAF;
- ✓ Connecting to HOPEX UAF;
- ✓ The HOPEX UAF Desktop;
- ✓ About This Guide.

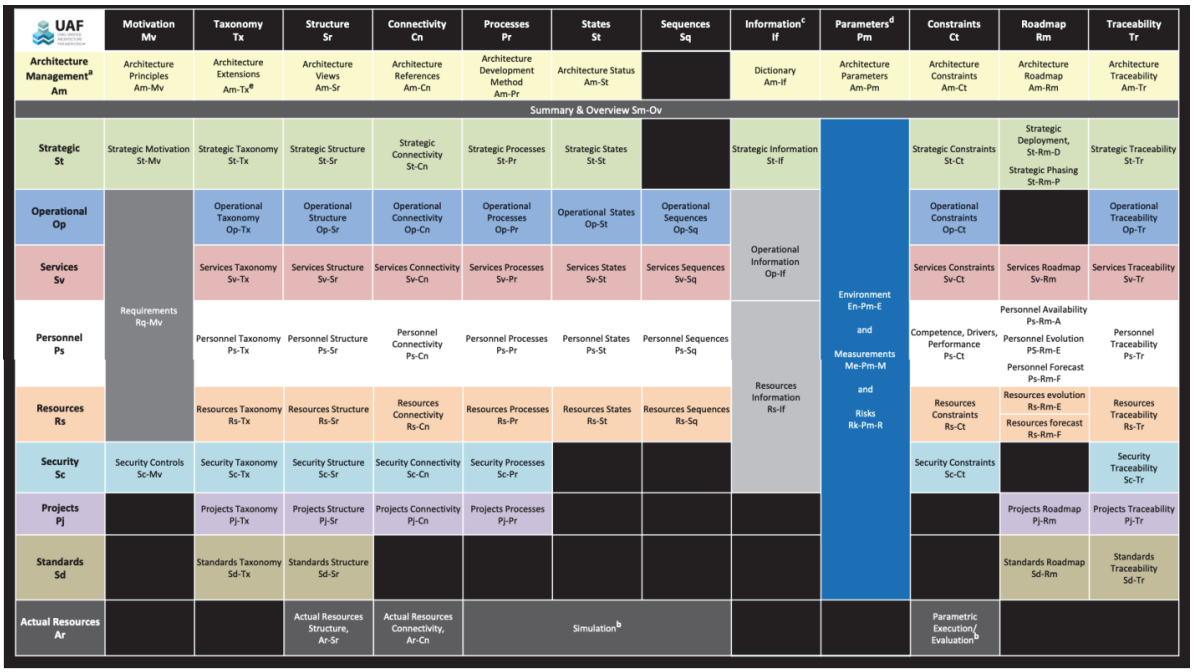
PRESENTATION OF HOPEX UAF

HOPEX UAF features summary

UAF Grid viewpoints and aspects

The grid is a way of showing how the various view specifications (cells) correspond to ***viewpoints*** (prev. known as domains) (horizontal rows) and the ***aspects*** (prev. known as model kinds) (the columns) that describe the view specification.

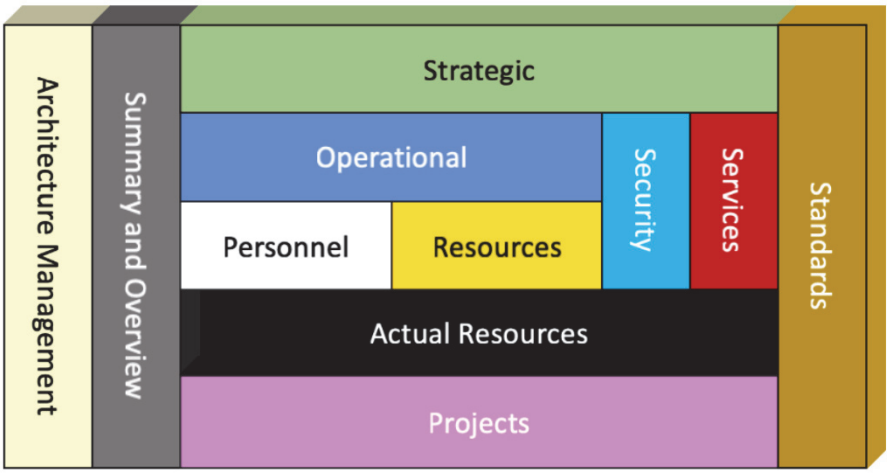
The intent of the grid is not to be complete, but to capture the information that is present in the frameworks that contributes to the UAF. Consequently, some gaps are evident.



UAF	Motivation Mv	Taxonomy Tx	Structure Sr	Connectivity Cn	Processes Pr	States St	Sequences Sq	Information[c] If	Parameters[d] Pm	Constraints Ct	Roadmap Rm	Traceability Tr
Architecture Management[a] Am	Architecture Principles Am-Mv	Architecture Extensions Am-Tx[e]	Architecture Views Am-Sr	Architecture References Am-Cn	Architecture Development Method Am-Pr	Architecture Status Am-St		Dictionary Am-If	Architecture Parameters Am-Pm	Architecture Constraints Am-Ct	Architecture Roadmap Am-Rm	Architecture Traceability Am-Tr
Summary & Overview Sm-Ov												
Strategic St	Strategic Motivation St-Mv	Strategic Taxonomy St-Tx	Strategic Structure St-Sr	Strategic Connectivity St-Cn	Strategic Processes St-Pr	Strategic States St-St		Strategic Information St-If	Environment En-Pm-E and Measurements Me-Pm-M and Risks Rk-Pm-R	Strategic Constraints St-Ct	Strategic Deployment, St-Rm-D Strategic Phasing St-Rm-P	Strategic Traceability St-Tr
Operational Op	Requirements Rq-Mv	Operational Taxonomy Op-Tx	Operational Structure Op-Sr	Operational Connectivity Op-Cn	Operational Processes Op-Pr	Operational States Op-St	Operational Sequences Op-Sq	Operational Information Op-If		Operational Constraints Op-Ct		Operational Traceability Op-Tr
Services Sv		Services Taxonomy Sv-Tx	Services Structure Sv-Sr	Services Connectivity Sv-Cn	Services Processes Sv-Pr	Services States Sv-St	Services Sequences Sv-Sq			Services Constraints Sv-Ct	Services Roadmap Sv-Rm	Services Traceability Sv-Tr
Personnel Ps		Personnel Taxonomy Ps-Tx	Personnel Structure Ps-Sr	Personnel Connectivity Ps-Cn	Personnel Processes Ps-Pr	Personnel States Ps-St	Personnel Sequences Ps-Sq	Resources Information Rs-If		Competence, Drivers, Performance Ps-Ct	Personnel Availability Ps-Rm-A Personnel Evolution PS-Rm-E Personnel Forecast Ps-Rm-F	Personnel Traceability Ps-Tr
Resources Rs		Resources Taxonomy Rs-Tx	Resources Structure Rs-Sr	Resources Connectivity Rs-Cn	Resources Processes Rs-Pr	Resources States Rs-St	Resources Sequences Rs-Sq			Resources Constraints Rs-Ct	Resources evolution Rs-Rm-E Resources forecast Rs-Rm-F	Resources Traceability Rs-Tr
Security Sc	Security Controls Sc-Mv	Security Taxonomy Sc-Tx	Security Structure Sc-Sr	Security Connectivity Sc-Cn	Security Processes Sc-Pr					Security Constraints Sc-Ct		Security Traceability Sc-Tr
Projects Pj		Projects Taxonomy Pj-Tx	Projects Structure Pj-Sr	Projects Connectivity Pj-Cn	Projects Processes Pj-Pr						Projects Roadmap Pj-Rm	Projects Traceability Pj-Tr
Standards Sd		Standards Taxonomy Sd-Tx	Standards Structure Sd-Sr								Standards Roadmap Sd-Rm	Standards Traceability Sd-Tr
Actual Resources Ar			Actual Resources Structure, Ar-Sr	Actual Resources Connectivity, Ar-Cn	Simulation[b]					Parametric Execution/ Evaluation[b]		

UAF Viewpoint Interrelationships

The following diagram is an indication of the how the viewpoints are interrelated.



HOPEX UAF implementation in HOPEX

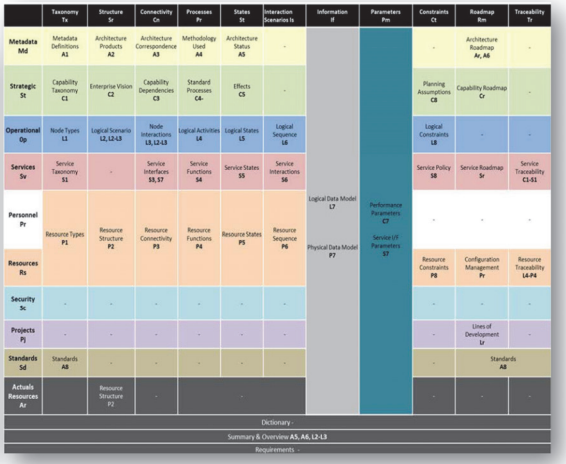
We use the diagram indicating how the viewpoints are interrelated to present which **HOPEX** Solutions are used to provide **HOPEX UAF**.

- ***Core***: supports the Architecture Management. Its implementation is based on **HOPEX** standard features.
- ***ITBM***: supports aspects of Strategic, Operational, Projects and Actual Resources viewpoints. Those features are based on **HOPEX IT Business Management** , which provides a methodology and tools to describe and plan your business transformation.
- ***ITPM***: supports aspects of Resources viewpoint. Those features are based on **HOPEX IT Portfolio Management**, which enables the description of all components involved in management of the enterprise application assets.
- ***ITA***: supports aspects of Resources and Standards viewpoints. Those features are based on **HOPEX IT Architecture** product which is dedicated to the application, technology and implementation layers. It offers possibilities to model the information system architecture according to a number of analysis perspective.
- ***BPA***: supports aspects of Personnel viewpoint. Those features are based on **HOPEX Business Process Analysis,** which supports a methodology and tools to describe a business organization and manage change.
- ***ITRM***: supports aspects of Security viewpoint. Those features are based on **HOPEX IT Risk Management** product which offers tools that simplify creation and analysis of risks to identify the most important of these and set up adapted corrective or preventive action.

HOPEX UAF and NAF V4 framework

UAF specification version 1.2 provides a mapping for **NAF V4.**

NAF v4 grid



UAF and NAF views mapping

Comparing the both grids, the NAF viewpoints can be mapped to a subset of the UAF viewpoints

HOPEX UAF solution provides a **NAF V4** desktop whose menus and submenus are organized according to the NAF grid.

For more details on profils accessing to NAF desktop, see HOPEX UAF Profiles.

For more details on NAF desktop, see The NAF desktop provided by HOPEX UAF.

CONNECTING TO HOPEX UAF

The menus and commands available in **HOPEX** depend on the profile with which you are connected.

Connecting to the solution

To connect to **HOPEX UAF**, see **HOPEX Common Features**, "Accessing HOPEX".

HOPEX UAF Profiles

In **HOPEX UAF**, there are default user profiles with which specific rights and accesses are associated. These profiles are:

- UAF Architect
- UAF Functional Administrator
- NAF Architect
- NAF Functional Administrator

> ☛ *For information regarding Administration management see "Users" chapter in HOPEX Administration - Supervisor guide.*

Profiles	Tasks
UAF Functional Administrator	The **UAF Functional Administrator** is responsible for managing all the administration tasks, the profile can create the working environments from Enterprises,. For more details, see Pre-requisites to HOPEX UAF.
UAF Architect	The **UAF Architect** is the end user profile, entitled to create Models and View Diagrams according to the defined viewpoints.
NAF Functional Administrator	The **NAF Functional Administrator** is responsible for managing all the administration tasks. For more details, see Pre-requisites to HOPEX UAF.
NAF Architect	The **NAF Architect** is the end user profile, entitled to create Models and View Diagrams according to the defined viewpoints.

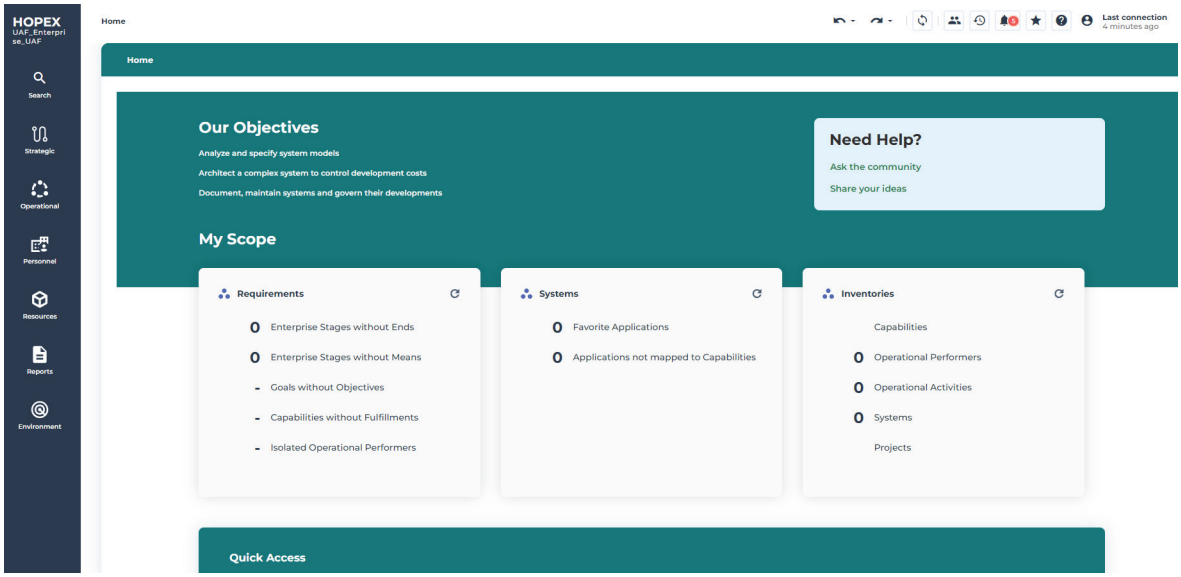
THE HOPEX UAF DESKTOP

The menus available in **HOPEX UAF** depend on the profile with which you are connected.

For a detailed description of the HOPEX interface, see PLATFORM - Common Features > HOPEX Desktop > Interface Presentation.

HOPEX UAF Home Page

The **HOPEX UAF** homepage is divided into the following sections.



HOPEX UAF Home page

- The header presents some information of general interest, see HOPEX UAF Header homepage.
  - **My priorities**: indicates the main strategic themes of interest to solution users.
  - **Help**: points to user documentation and the user community.
- The **My Scope** provides useful indicators of the repository content. See Scope Indicators below.
- The **Quick Access** provides useful shortcuts:
  - **Recently viewed**: last objects and diagrams accessed by the user
  - **Favorites**: user favorites and shared favorites
  - **Actions**: quick access to the creation of architecture elements, see Quick Access Action.
- **My favorite report**: displays the user-defined or administrator-predefined report, which can be used as an entry point into the repository.

HOPEX UAF Header homepage

The header homepage is made of the following standard objectives:

- **Analyze and specify system models**: Stakeholders can use standardized, view-based concepts and models to inter operate effectively and facilitate communication.
- **Architect a complex system to control development costs**: UAF is an architectural framework that describes complex systems and the interactions that exist between them and with organizations. It helps define and understand the overall needs and capabilities to be achieved.
- **Document, maintain systems and govern their developments**: Describing systems in a repository helps to document them for shared understanding and knowledge between stakeholders.

Scope Indicators

The **My Scope** section provides useful indicators on application assets. Clicking the indicator takes you to all the corresponding objects. There are three groups of indicators:

- Requirements,
- System,
- Inventories.

Requirements

This tile lists the following objects:

- **Enterprise stage without Ends**: Number of Enterprise Stage, sub-stage of current enterprise (included the current enterprise) that don't have Ends (Visions, Goals, or Objectives).
  This displays of list of these objects (name of stage, description, etc.)
- **Enterprise stage without Means**: Number of Enterprise Stage, sub-stage of current enterprise (included the current enterprise) that don't have Means (Missions, Strategies, Tactics).
  This displays of list of these objects (name of stage, description, etc.)
- **% Goals without objectives**: Percent of Goals of current enterprise, its stage and their sub-stages that don't have linked objectives.
  This displays of list of these objects (name of Goal, description, multi-combox of stages of current Enterprise).
- **% Capabilities without fulfillments**: Percent of Capabilities of current enterprise that aren't fulfilled by a system.
  This displays of list of these objects (name of Capability, description, etc.).
- **% Isolated Operational Performer**: Percent of Operational Performer of current enterprise without any interactions.
  This displays of list of these objects (name of Operational Performer, description, etc.)

Systems

This tile lists the following objects:

- **Favorite Applications**: Number of Applications and Application Systems that are added in different Favorite folders.
  This displays of list of these objects (name, description, etc.)
- **Applications not mapped to Capabilities**: Number of Applications of current Enterprise that do not fulfill any Capability.
  This displays of list of these objects (name, description, etc.)
- **Applications without exposed Exchanges**: Number of Applications of current Enterprise that do not exchange any Message.
  This displays of list of these objects (name, description, etc.)
- **Applications with sensitive or critical data**: Number of Applications of the current Enterprise that respect one of the following criteria:
  - Have Data Stores referencing a Data Domain using sensitive or critical Data categories
  - Send or receive flows with content referencing sensitive or critical Data categories.

  This displays of list of these objects (name, description, etc.)

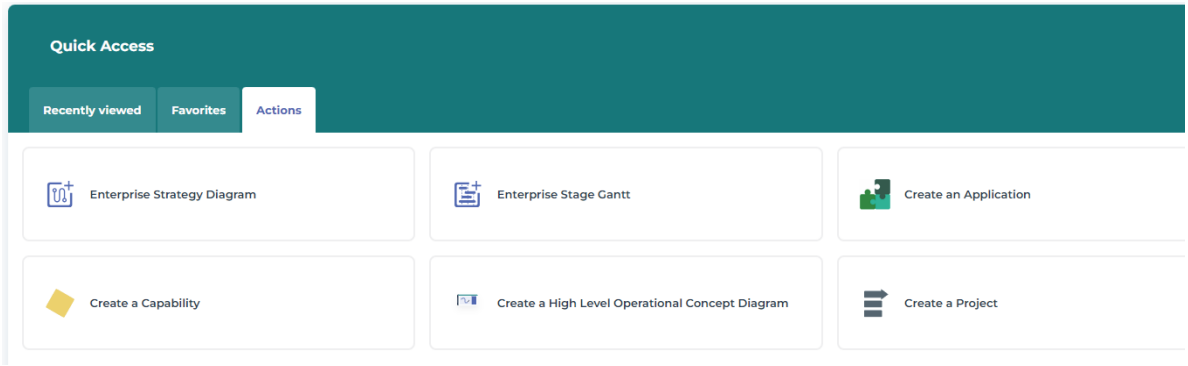
Inventories

This tile lists the following objects:

- **Capabilities**: Number of Capabilities that are owned and imported by the current Enterprise.
  This displays oof list of these objects (name, description, owned/imported flag)
- **Operational Performers**: Number of Performers agents that are owned and imported by the current Enterprise.
  This displays oof list of these objects (name, description, owned/imported flag)
- **Operational Activities**: Number of Activities that are owned and imported by the current Enterprise.
  This displays of list of these objects (name, description, etc.)
- **Systems**: Number of Systems (applications, Applications Systems) that are owned and imported by the current Enterprise.
  This displays of list of these objects (name, description, owned/imported flag)
- **Projects**: Number of Projects that are owned by Project Domains contained in the current Enterprise.
  This displays of list of these objects (name, description, etc.)

Quick Access Action

Quick Access contains shortcuts facilitating your work in **HOPEX UAF**.



The list of **Actions** proposed with **HOPEX UAF** is presented below.

- **Enterprise Strategy Diagram**, see Creating an Enterprise Strategy Diagram,
- **Enterprise Stage Gantt**, see Enterprise Stage Gantt Report,
- **Create an Application**, see Describing an Application with HOPEX UAF,
- **Create a Capability**, see Describing a Capability,
- **Create a High Level Operational Concept Diagram**, see Using High Level Operational Concepts with HOPEX UAF,
- **Create a Project**, see Creating a Project Demand.

UAF Architect Desktop

The **HOPEX UAF** the main navigation menus correspond to UAF Viewpoints:

Strategic

The **Strategic** menu is dedicated to the UAF Strategic Viewpoint. It allows the description of the capability taxonomy, composition, dependencies, and evolution. It mainly addresses the Capacity portfolio managers.

See Strategic Viewpoint.

Operational

The **Operational** menu is dedicated to the UAF Operational Viewpoint.

See Operational Viewpoint.

Services

The **Services** menu is dedicated to the UAF Service Viewpoint.

See Services Viewpoint.

Personnel

The **Personnel** menu is dedicated to the UAF Personnel Viewpoint.

See Personnel Viewpoint.

Resources

The **Resources** menu is dedicated to the UAF resources Viewpoint.

See Resources Viewpoint.

Security

The **Security** menu is dedicated to the UAF Security Viewpoint for the transformation project management.

See Projects Viewpoint.

Projects

The **Projects** menu is dedicated to the UAF Strategic Viewpoint for the transformation project management.

See Projects Viewpoint.

Standards

The **Standards** menu is dedicated to the UAF Standards Viewpoint, it mainly addresses the specification of the protocol stack used in the architecture.

See Standards Viewpoint.

Parameters

The **Parameters** menu provides access to features availables in several usage contexts.

- **Environment**, provides the list of Measurable Properties connected to a list of Environments specified in the report parameters.

  ☛ *For more details on measurable properties with **HOPEX UAF**, see Describing a Measurable Property*

- **Measurements**, to access the list of *measurable properties*.

  🕮 *A measurable property expresses the nature of indicators (duration, mass, cost, etc.) and defines the unit used to measure them (minutes, kilograms, euros, etc.). Measurable properties are used to define indicators, they can be elementary or composite. Elementary measurable properties are described by measurement units: kg, Liter, Gallon, Hour, Minute.*

  ☛ *For more details on measurable properties with **HOPEX UAF**, see Describing a Measurable Property*

- **Risk Measures,** to access the list of *risks*.

  🕮 *A risk is a hazard of greater or lesser probability to which an organization is exposed.*

  ☛ *For more details on risks definition with **HOPEX UAF**, see Defining Risks*

Reports

The **Reports** menu provides a search tool for all report templates and saved reports.

☛ *For more details on reports, see PLATFORM - Common Features > Documentation > Generating Documentation > Generating Reports.*

Environment

The **Environment** menu gives access to the following submenus:

- **Containers**, to access the features of library and environment management.

  ☛ *For more details on **Containers**, see Using a Working Environment.*

- **Report DataSets**, to access all the reports DataSets of your repository.

  ☛ *For more details on report DataSetss, see PLATFORM - Common Features > Documentation > Generating Documentation > Generating Reports.*

- **Sketches**, to access all the skectches of your repository.

  🕮 *A sketching diagram is a drawing that enables you to exchange with your coworkers without an issue of methodology or formalism.*

  ☛ *For more details on the use of sketches with **HOPEX IT Architecture**, see Using High Level Operational Concepts with HOPEX UAF.*

- **Tags**

  🕮 *A tag is a classifying description used to characterize objects.*

  ☛ *For more details on the use of tags, see Platform - Common Features > Collaboration Tools > Communicating in HOPEX.*

UAF Functional Administrator Desktop

The activities offered only to users connected with the **UAF Functional Administrator** profile are:

- Administration, via the **Administration** menu,
- the creation of objects from the **Environment** menu, see Environment.

The **Administration** menu gives access to the following submenus:

- **Assessment**, giving access to the following components.

  ☛ *For more details on Assessments, see chapter "Assessment" in the **HOPEX Common Features** guide.*

  - **Assessment Templates**,
  - **Assessment Characteristics**,
- **Categorization**, giving access to the Categorization Schemes.

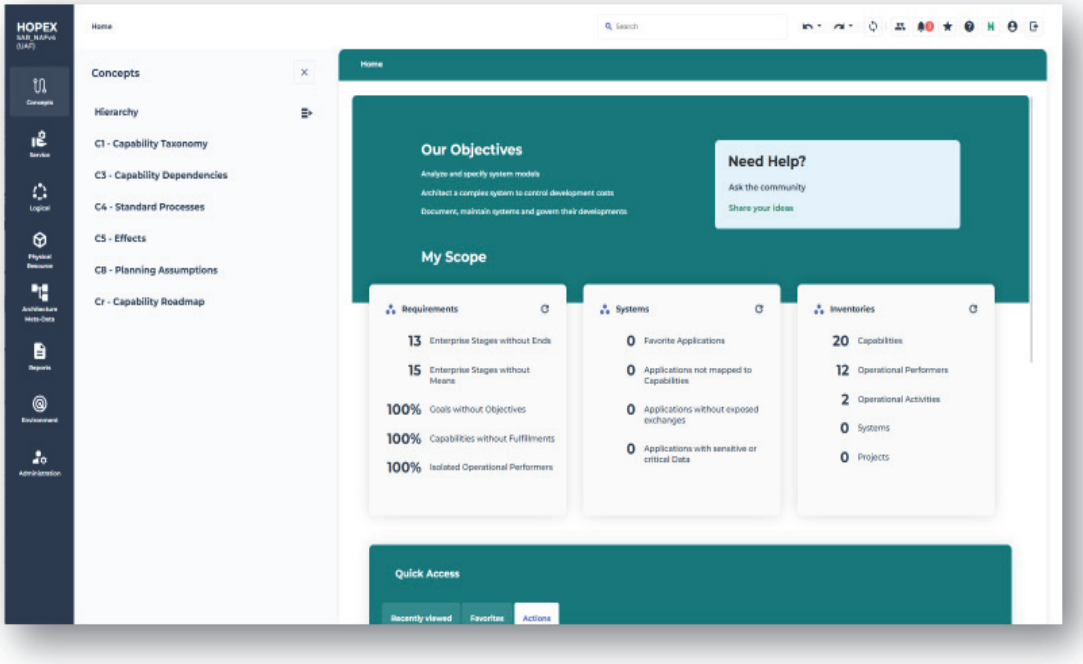
  ☛ *For more information on Data Categories see the **HOPEX Data Governance** guide.*

  ☛ *For more information on Data Categories, see chapter "Defining Categorization Schemas" in the **HOPEX IT Business Management** guide.*
- **IT Templates** giving access to templates associated with the following components:
  - **Service Interfaces**, see Describing an Interface,
  - **Service Operations**, see Describing a Service Operation,
  - **Contents**, see Using Signals,
  - **Deployment Architecture**,

    ☛ *For more information on Deployment Architecture Template, see chapter "Deployment Architecture Template" in the **HOPEX IT Architecture** guide.*
- **Methodological Domains**, each **HOPEX** Solution offers its own methodological domains defining all Solution users' common goals.

The NAF desktop provided by HOPEX UAF

The **HOPEX UAF** solution provides a **NAF V4** desktop whose menus and submenus are organized according to the NAF grid.



About This Guide

This guide presents how to make best use of **HOPEX UAF** to assure efficient management of your environment.

Guide Structure

The **HOPEX UAF** guide comprises the following chapters:

- HOPEX Implementation of UAF: describes the main principles that govern the ArchiMate® implementation of **HOPEX UAF**.
- Strategic Viewpoint: presents the Capability management process offered by **HOPEX UAF**.
- Operational Viewpoint: presents the Logical Architecture modeling with **HOPEX UAF**.
- Services Viewpoint: presents the Service management process offered by **HOPEX UAF**.
- Personnel Viewpoint: presents defines and explores organizational resource types offered by **HOPEX UAF**.
-
- The HOPEX implementation of ArchiMate: describes the **HOPEX** metamodel used to implement **HOPEX UAF**.

Additional Resources

This guide is supplemented by:

- The **HOPEX Common Features** guide, which describes basic functions common to **HOPEX** solutions.

  ☛ *It can be useful to consult this guide for a general presentation of the interface.*

- The administration guide **HOPEX Power Supervisor**.
- more advanced technical functions are described in the **HOPEX Power Studio** guide.

Conventions Used in the Guide

Styles and formatting

Remark on the preceding points.

Definition of terms used.

A tip that may simplify things.

Compatibility with previous versions.

Things you must not do.

Very important remark to avoid errors during an operation.

Commands are presented as seen here: **File > Open**.

Names of products and technical modules are presented in bold as seen here: **HOPEX**.

HOPEX IMPLEMENTATION OF UAF

HOPEX proposes an implementation based on the UAF specification which aims at ensuring continuity with other **HOPEX** products.

This section deals with **HOPEX** implementation of UAF. 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 UAF deliverables.

- ✓ The UAF Viewpoints and Aspects implementation
- ✓ Before starting with HOPEX UAF

The UAF Viewpoints and Aspects implementation

UAF is composed of a set of deliverables, called viewpoints, which address different parts of an enterprise. These viewpoints are grouped into models. Models that focus on the same perspective are placed in the same viewpoint.

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

UAF Viewpoints

Viewpoints	Acronym	Description
Architecture Management	Am	Identifies the metadata required to develop a suitable architecture that is fit for its purpose.
Strategy	St	Capability management process. Describes the capability taxonomy, composition, dependencies, and evolution.
Operational	Op	Illustrates the Logical Architecture of the enterprise. Describes the requirements, operational behavior, structure, and exchanges required to support (exhibit) capabilities. Defines all operational elements in an implementation/solution-independent manner.
Services	Sv	The Service-Orientated View (SOV) is a description of services needed to directly support the operational domain as described in the Operational View. A service within MODAF is understood in its broadest sense, as a unit of work through which a provider provides a useful result to a consumer. DoDAF: The Service Views within the Services Viewpoint describe the design for service-based solutions to support operational development processes (JCIDS) and Defense Acquisition System or capability development within the Joint Capability Area.
Personnel	Pr	Defines and explores organizational resource types. Shows the taxonomy of types of organizational resources as well as connections, interaction and growth over time.

Viewpoints	Acronym	Description
Resources	Rs	Captures a solution architecture consisting of resources, e.g., organizational, software, artifacts, capability configurations, and natural resources that implement the operational requirements. Further design of a resource is typically detailed in SysML or UML
Security	Sc	Security assets and security enclaves. Defines the hierarchy of security assets and asset owners, security constraints (policy, laws, and guidance), and details where they are located (security enclaves).
Projects	Pj	Describes projects and project milestones, how those projects deliver capabilities, the organizations contributing to the projects, and dependencies between projects.
Standards	Sd	MODAF: Technical Standards Views are extended from the core DoDAF views to include non-technical standards such as operational doctrine, industry process standards, etc. DoDAF: The Standards Views within the Standards Viewpoint are the set of rules governing the arrangement, interaction, and interdependence of solution parts or elements.
Actual Resources	Ar	The analysis, e.g., evaluation of different alternatives, what-if, trade-offs, V&V on the actual resource configurations. Illustrates the expected or achieved actual resource configurations.

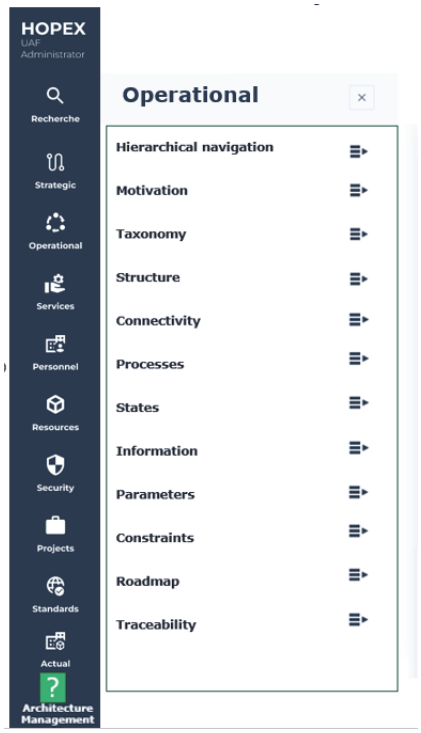
UAF Aspects

Aspect	Acronym	Description
Motivation	Mv	Captures motivational elements e.g., challenges, opportunities, and concerns, that pertain to enterprise transformation efforts, and different types of requirements, e.g., operational, services, personnel, resources, or security controls.
Taxonomy	Tx	Presents all the elements as a standalone structure. Presents all the elements as a specialization hierarchy, provides a text definition for each one and references the source of the element
Structure	Sr	Describes the breakdown of structural elements e.g., logical performers, systems, projects, etc. into their smaller parts

Aspect	Acronym	Description
Connectivity	Cn	Describes the connections, relationships, and interactions between the different elements
Processes	Pr	Captures activity based behavior and flows. It describes activities, their Inputs/Outputs, activity actions and flows between them.
States	St	Captures state-based behavior of an element. It is a graphical representation of states of a structural element and how it responds to various events and actions.
Sequences	Sq	Expresses a time-ordered examination of the exchanges as a result of a particular scenario. Provides a time-ordered examination of the exchanges between participating elements as a result of a particular scenario.
Information	If	Address the information perspective on operational, service, and Capacity Configurations. Allows analysis of an architecture's information and data definition aspect, without consideration of implementation specific issues.
Parameters	Pm	
Constraints	Ct	Details the measurements that set performance requirements constraining capabilities. Also defines the rules governing behavior and structure.
Roadmap	Rm	Addresses how elements in the architecture change over time.
Traceability	Tr	Describes the mapping between elements in the architecture. This can be between different viewpoints within domains as well as between domains. It can also be between structure and behaviors.

Access to Viewpoints and Aspects from the Navigation Bar

The **HOPEX UAF** navigation bar provides access to all UAF Viewpoints.



*Navigation menus available with **HOPEX UAF***

From any UAF Viewpoint, **HOPEX UAF** navigation bar provides access the corresponding UAF Aspects. For more details, see UAF Architect Desktop

BEFORE STARTING WITH HOPEX UAF

Pre-requisites to HOPEX UAF

Importing UAF Module

If you want to use **HOPEX UAF**, you must import the "UAF" module in your environment.

To work with certain HOPEX Solutions, you might need to import modules in **HOPEX**, see **Modules > Importing a Module** documentation.

You must import the UAF module only once even if you have several repositories.

Assigning UAF Profiles to users

Before starting with **HOPEX UAF**, users must be assigned to **HOPEX UAF** profiles.

*For information about **HOPEX UAF** profiles, see HOPEX UAF Profiles.*

For information regarding users profiles, see HOPEX Administration - Supervisor Web > Assigning a Profile to a Person

Using a Working Environment

So that users can work in **HOPEX UAF**, the **UAF Functional Administrator** has to create an **Enterprise**.

An Enterprise is a purposeful undertaking, conducted by one or more organizations, aiming at delivering goods and services, in accordance with the enterprise mission in its changing environment. During its development over time, an enterprise has to adapt to its environment and sets up transformation goals and objectives along with course of action to achieve these objectives. The design and realization of the resulting transformation stages may transcend organizational boundaries and consequently require an integrated team working under the direction of a governing body to involve stakeholders in transformation initiatives.

t*Libraries are collections of objects used to split repository content into several independent parts. They allow creation of virtual partitions of the repository. In particular, two objects owned by different libraries can have the same name.*

In the context of the **HOPEX UAF** solution, a library can hold all the elements of your project.

*For more details on managing libraries, see the "Enterprises and Libraries" chapter in the **HOPEX Common Features** guide.*

The **Environment > Containers** navigation menu provides access to the list of libraries and Enterprises of your repository. The content of Containers element are trees.

☛ *To get more information about the use of folders in **HOPEX** trees, see **HOPEX**, "Handling Trees" chapter of the **HOPEX Common Features** guide.*

Creating an Enterprise

To allow architects to start working with **HOPEX UAF**, the **UAF Functional Administrator** needs to create an UAF **Enterprise**.

To create and initialize an UAF *Enterprise*:

1. Select the **Environment > Containers** and click the **Enterprise** tab.
2. Click **New**.
   An enterprise creation dialog box opens.
3. Specify the name of the Enterprise.
4. Check the box **Working Environment Automatic Creation**.
5. In the **Working Environment Template** section, select **UAF**.
6. Click **OK**.
   The work environment for the enterprise is automatically created.

Accessing your Working Environment

To be able to start working with **HOPEX UAF**, you first need to select your work environment.

If you are connected with an **UAF Architect** or an **UAF Functional Administrator** profile, you must have already defined a work environment for you.

☛ *If not, please contact your UAF functional administrator.*

For more information, see Creating an Enterprise.

To start working, you have to specify your work environment:

- From the main menu, click **Switch work environment** and select the work environment of interest to you.
  Your work environment will be memorized at later connections.

  ☛ *You may have one or several work environments. If you have several work environments, you may have to switch between them through this menu.*

Enterprise Properties

To display the properties of an Enterprise:

1. From the **Environment > Containers** navigation menu, expand the **Enterprise** folder.
2. Click the Enterprise that interests you to open its properties.

With **HOPEX UAF**, an Enterprise is described by several property pages.

The **Characteristics** properties page of an enterprise provides access to:

- its **Local Name**
- its **Owner**, which is a library or another enterprise.
- its **Enterprise Operating Model**, which can be of one of different types.

The **Roadmap** property page of an enterprise provides access to the list of Transformation Stages of the Enterprise.

> ☛ *For more details on transformation stages, see Defining transformation stages.*

The **Strategy** property page of an enterprise provides access to the following information:

- the **Ends** section that is used to access to the Ends involved in the transformation stage.

  > ☛ *For more details on ends elements, see Creating Enterprise goals.*

- the **Means** section that is used to access to the Means involved in the transformation stage.

  > ☛ *For more details on means elements, see Defining Means.*

- the **exhibited capabilities** page that is used to access to the Capabilities involved in the transformation stage.

  > ☛ *For more details on strategic elements, see Managing Exhibited Capabilities.*

The **Architecture Description** page that is used to access to the different elements of the Enterprise Architecture.

> ☛ *For more details on architecture description, see Architecture Description.*

The **Timeline** page provides access to the Enterprise Stage Gantt.

> ☛ *For more details on this report, see Enterprise Stage Gantt Report.*

The **Working Environment Assignment** page provides access to the Enterprise Working Environment Template selected during the Enterprise creation.

> ☛ *For more details on an UAF Enterprise creation, see Creating an Enterprise.*

The **Reporting** page provides access to the Enterprise reports.

> ☛ *For more details on this report, see Enterprise Reports.*

The **Diagrams** page provides access to the Enterprise diagrams.

> ☛ *For more details on entreprise diagrams, see Accessing Enterprise Diagrams.*

Accessing Enterprise Diagrams

Quick Access contains shortcuts facilitating your work in **HOPEX UAF**.

> ☛ *For more details on Quick Access, see Quick Access Action.*

This shortcut provides access to the following Enterprise diagrams:

- Creating an Enterprise Strategy Diagram,
- Enterprise Stage Gantt Report.

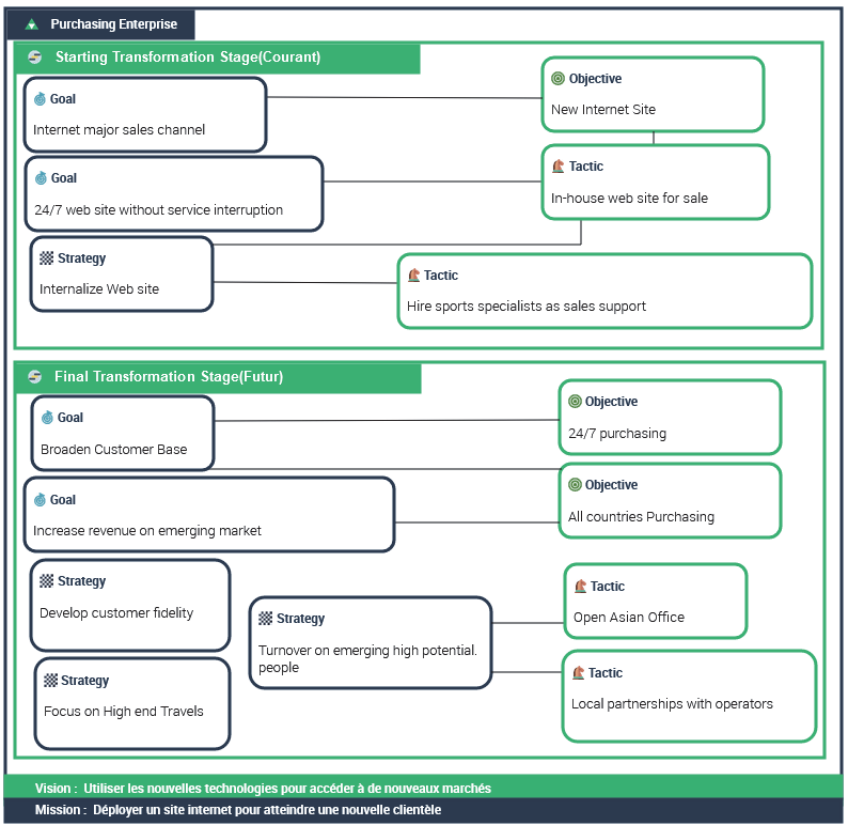
Creating an Enterprise Strategy Diagram

The enterprise strategy diagram describes the links between missions, goals, strategies, tactics and exhibited Capabilities.

To draw this diagram:

1. Select the enterprise and click **New > Diagram**.
2. Select **Enterprise Strategy Diagram**.

☛ *You can also create this type of diagram from an Enterprise Stage.*



The components represented in an enterprise diagram are the strategic elements connected to the enterprise and to its *Transformation stages*.

☛ *For further details, see Defining Enterprise strategic elements.*

The components represented in an enterprise diagram are :

- The enterprise ends, described by the *enterprise goals* and the *enterprise objectives* (that are described at the *transformation stages*)).

  📖 *A Goal tends to be longer term, and defined qualitatively rather than quantitatively. It should be sufficiently narrow-focused that Objectives can be defined for it.*

  📖 *An objective is a goal that a company/organization wants to achieve, or the target set for a Personnel Function or an operation. An*

objective allows you to highlight the features in a Personnel Function or operation that require improvement.

- The enterprise means that are described by *strategies* at the enterprise level and by *tactics* at the transformation stage level.

  📖 *A Strategy is one component of the plan for the Mission. A Strategy represents the essential Course of Action to achieve Ends (Goals in particular). A Strategy usually channels efforts towards those Goals. A Strategy is accepted by the enterprise as the right approach to achieve its Goals, given the environmental constraints and risks.*

  📖 *A Tactic is a Course of Action that implements part of a Strategy. Tactics generally channel efforts towards Objectives.*

Architecture Description

To define the architecture of your enterprise:

1. From the **Environment > Containers** navigation menu, select the **Enterprises** tab.
2. Click the Enterprise that interests you to open its **Architecture Description** property page.

Three sections provides access to information regarding:

- Capability architecture,
- Operating Architecture,
- Information Architecture.

Capability architecture

You can specify:

- the enterprise capability map,
- the enterprise resource capability maps
  - Competence Map,
  - Functionality map,
  - Hardware Capability Map
  - Technology Functionality Map

Operating Architecture

You can specify:

- the enterprise operating model
- the enterprise operating architectures
  - Application Environment
  - Application System Environment
  - Logical Application System Environment
  - Organization Environment
  - Resource Architecture Environment
- the Business Operating Scenarios
  - Global Application Flow Map
  - Scenario of Application Environment Flow
  - Scenario of Application System Environment Flow
  - Scenario of Logical Application System Environment Flow
  - Scenario of Personal Environment Flow

Information Architecture

You can specify:

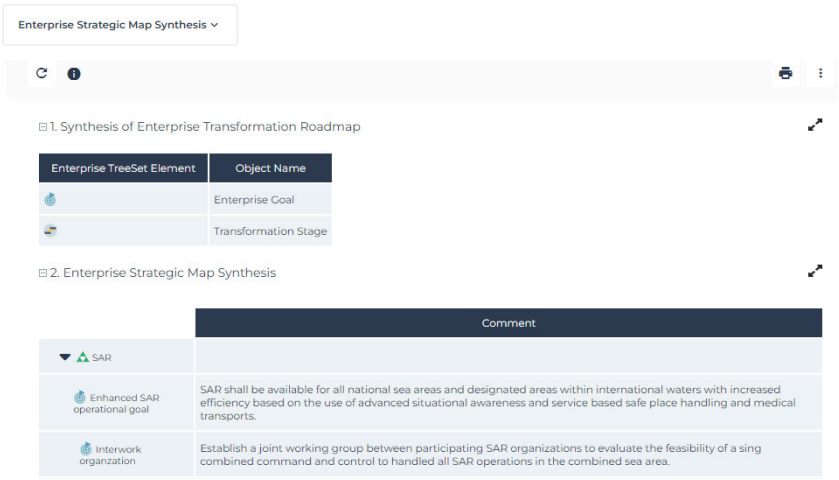
- the Concept Domain Map and the corresponding
  - Data Domain Map
  - Relational Schema Map

Enterprise Reports

Several reports can be accessed from an Enterprise **Reporting** property page.

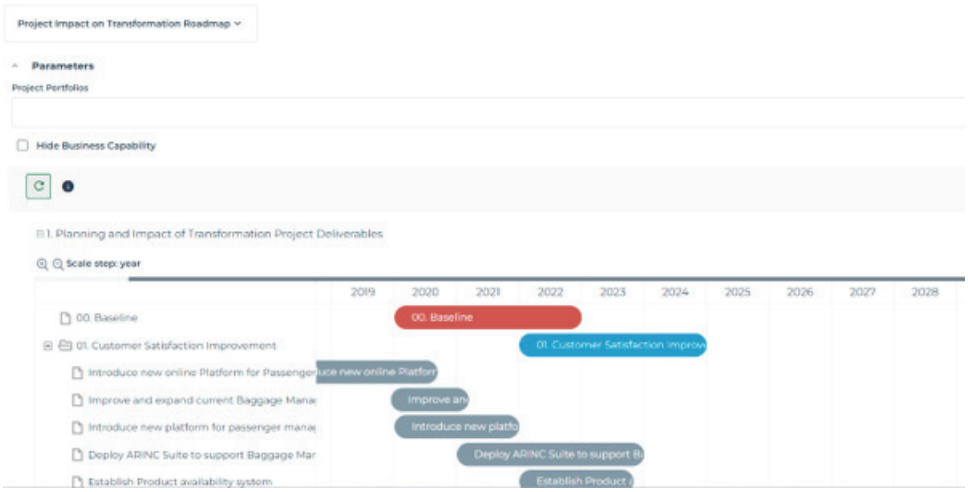
Enterprise Strategic Map Synthesis

This report gives the tree structure of the objects of the analyzed Enterprise.



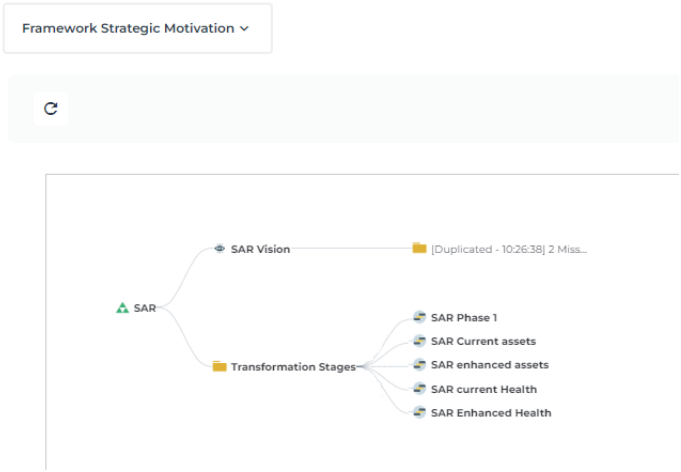
Project Impact on Transformation Roadmap

This report enables the user to have a clear and synthetic vision of the transformation stages of his enterprise, link to each goal and objective.



Framework Strategic Motivation

This report describes the Enterprise Vision and its transformation stages.



Describing a Measurable Property

Measurable Properties and *Qualifying Values* are used to define the performance constraints that must be complied with by the building blocks making up the enterprise, at the forefront of which are the business capabilities and the exhibited business capabilities in a transformation stage.

The nature of a *Qualifying Value* is defined by a *Measurable Property*.

> 🕮 *A measurable property expresses the nature of indicators (duration, mass, cost, etc.) and defines the unit used to measure them (minutes, kilograms, euros, etc.). Measurable properties are used to define indicators, they can be elementary or composite. Elementary measurable properties are described by measurement units: kg, Liter, Gallon, Hour, Minute.*
>
> 🕮 *A qualifying value (key indicator) defines how much of something can be quantified, either as a singular value or as range of values, according to a Measurable Property. Key indicators are valued. Example: Response Time < 20 secondes.*

Measurable Properties can be connected to the some objects such as:

- business capabilities,
- value streams,
- Business Skill,
- application environment

Qualifying Values can be connected to exhibited business capabilities; that is, a capability highlighted within the context of a transformation stage. A *Qualifying Value* is attached to a *Measurable Property*.

```
For example, the Qualifying Values of the "Command
management" exhibition of the exhibited business
```

capability) in a given transformation stage (existing or future) can be "Deliver a pizza in less than 20mn" or "Take the order in less than 3mn".

Accessing the list of Measurable Properties of an Enterprise

To access the list of *Measurable Properties*:

- Expand the **Parameters > Measurements** navigation menu.
  The list of measurable properties appears in the **Measurable Properties** tab.

Creating a Measurable Property from a Business Capability

To create a *Measurable Property* from a Business Capability, for example:

1. Open the **Measurable Property** page of the Business Capability that interests you.
2. In the **Measurable Property** section, click **New**.
   A Measurable Property creation dialog box opens.
3. Specify:
   - its **Name**,
   - the text that describes its **Unit**,
   - the text of its **Description**.
4. Click **OK**.
   The new Measurable Property appears in the section. It's connected to the current library.

The properties of a measurable property

The **Characteristics** property page of a measurable property provides access to:

- its **Name**,
- the text that describes its **Unit**,
- the text of its **Description**.

The **Usage** property page of a measurable property provides access to:

- the **Qualifying Value** section provides the list of Qualifying Values that are based on this Measurable Property.
- the **Architectural Composite** section which provides the list of components associated to the Measurable Property.

☛ *For further details, see Creating a Measurable Property from a Business Capability.*

Using Measurable Properties Reports

A *Measurable Properties Reports* provides the list of Measurable Properties connected to a list of Assets specified in the report parameters.

To access the list of *Measurable Properties Reports*:

- Expand the **Parameters > Measurements** navigation menu.
  The list of measurable properties appears in the **Measurable Properties Reports** tab.

Using the Constraints aspects

Constraints represent obligations imposed on assets. Depending on the type of asset to which it applies, a constraint can be of different types:

- Constraints which type is therefore different depending on the type of objects associated with the viewpoints,
- Performance measures,
- Directing Regulation.

An object constraints are specified in dedicated property pages.

The Constraints reports

To create a Constraint report:

1. Select the tab corresponding to the type of report the interests you,
2. Click **New**.
3. Select the constrained objects.
4. Click **OK**.
   The matrix corresponding to the report is displayed.

Constraint Reports

The Constraint reports available from the **Contraints** aspect display in columns the **Name** and the **Description** of the objects connected in the **Constraints** properties of the constrained objects.

The type of the constraining objects depend on the type of the constrained objects, so on the viewpoint.

Measure Reports

The Measure reports available from the **Contraints** aspect display in columns some characteristics of the objects connected in the **Measurable Property** and **Qualifying Values** of the constrained objects.

For more details on Measurable Properties, see Describing a Measurable Property.

The columns of Measure reports are:

- **Measurable Property**, name of the selected constraining object,
- **Unit**, description of the selected constraining object,
- **Description**, description of the selected constraining object.
- **Measurement Schema**.,
- **Qualifying Value**.,
- **Logical Operator**.,
- **Threshold Value**.,
- **Measure Rank**.,

Politicy Reports

The Policy reports available from the **Contraints** aspect display in columns some characteristics of the objects connected in the **Regulations** properties of the constrained objects.

For more details on Directing Regulation structure, see Security - Taxonomy.

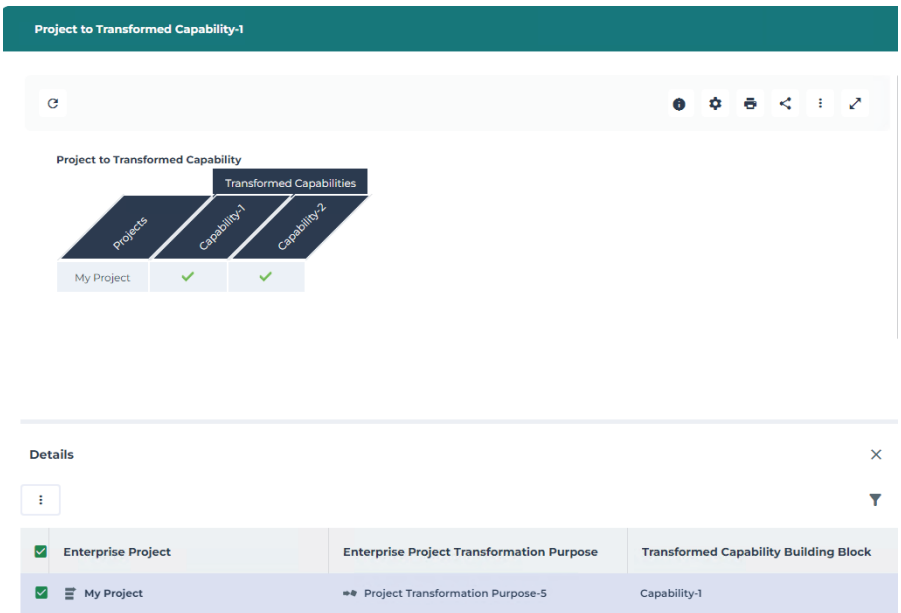
The columns of Policy reports are:

- **Directing Regulation**, name of the selected constraining object,
- **Policy description**, description of the selected constraining object,
- **Parent Policy block**.

Using Traceability Aspect

In several viewpoints, the **Traceability** aspect provides matrix and reports presenting the dependencies between some objects specific to the viewpoint.

For example, the **Project to Transformed capability** report presents a matrix of **capabilities** impacted by specifiic **Projects**.



The **Matrixes** of the **Traceability** aspect enable the UAF Architect to specify dependencies between the elements specified in the rows and in the columns.

The **Reports** of the **Traceability** aspect provide a report presenting in the columns the elements with dependencies with the elements specified by the UAF Architect as parameters of the report. A type of a report parameters corresponds to the type of elements expected in the matrix rows.

Accessing the Traceability matrixes

Depending on the viewpoint several matrix types may be proposed.

If only one matrix type is proposed, you must select the **Matrixes** tab to access the existing matrixes.

If several matrix types are proposed, you must select the tab corresponding to the matrix that interests you.

To access the traceability matrixes:

1. From the navigation menu, select the viewpoint that interests you and the **Traceability** submenu.
2. Click the matrix tab that interests you.
   The list of existing matrixes is displayed.

Creating a Traceability matrix

To create a Traceability matrix:

1. From the navigation menu, select the viewpoint that interests you and the **Traceability** submenu.
2. Click the matrix tab that interests you.
3. Click the **New** button.
   An empty matrix is displayed.
4. Click **Add row** to select the objects that interest you for the matrix lines.
5. Click **Add Column** to select the objects that interest you for the matrix columns.
   The matrix is displayed and recorded.

In a matrix, dependency between an object in a row and an object in a column is represented by check marks. You can add or remove such a dependendy by clicking in the corresponding box.

Accessing the Traceability reports

Depending on the viewpoint several reports types may be proposed.

If only one report type is proposed, you must select the **Reports** tab to access the existing reports.

If several report types are proposed, you must select the report that interests you in a drop-down.

Creating a Traceability report

To create a traceability report:

1. From the navigation menu, select the viewpoint that interests you and the **Traceability** submenu.
2. Click the **Reports** tab.
3. Select the type of matrix that interests you.
4. Click the **New** button.
5. Select the objects that will be presented in rows and click **OK**.
   The report is created and displayed. The objects presented in columns have dependencies witj objects presented in rows.

STRATEGIC VIEWPOINT

The **Strategic** viewpoint is dedicated to Capability management process. It allows the description of the capability taxonomy, composition, dependencies, and evolution. It mainly addresses the Capacity portfolio managers.

To manage the risks of complex procurements, there is a need to provide visualizations of the evolving capabilities so that Portfolio managers can synchronize the introduction of capability increments across a portfolio of projects.

Another justification for the capability viewpoint is the increasing importance of transformational programs within the DoD.

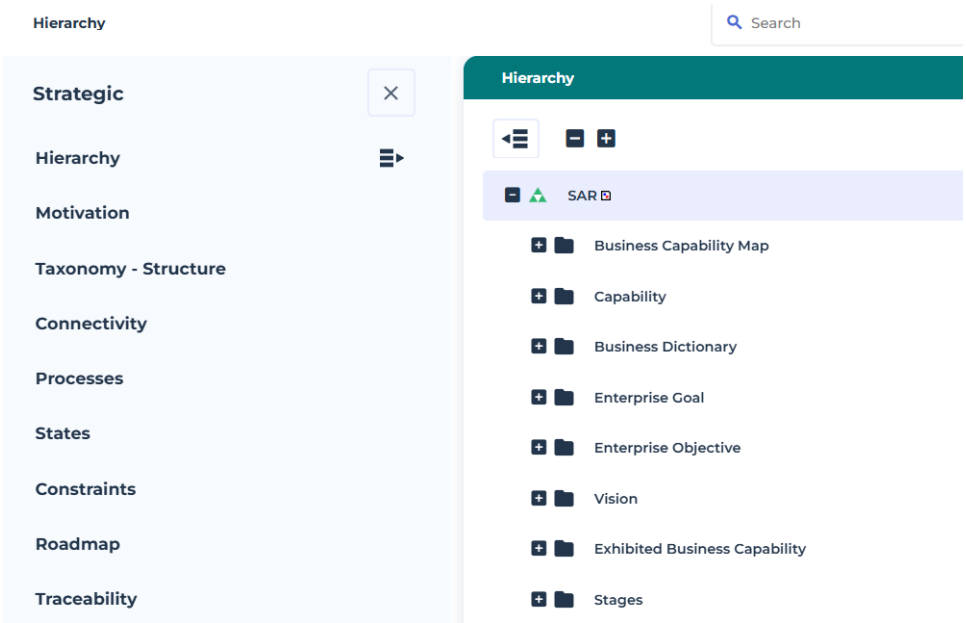
The concept of capability allows one to answer several questions such as:

- ✓ How does a particular capability or capabilities support the overall mission/vision?
- ✓ What outcomes are expected to be achieved by a particular capability or set of capabilities?
- ✓ What services are required to support a capability?
- ✓ What is the functional scope and organizational span of a capability or set of capabilities?
- ✓ What is our current set of capabilities that we are managing as part of a portfolio?

This chapter comprises the following sections:

STRATEGIC - HIERARCHY

The **Hierachy** aspect of the **Strategic** view helps UAF Architect to manage goals and objectives of the current Entreprise. This view allows the description of capabilities: their composition, dependencies and their evolution with a roadmap.



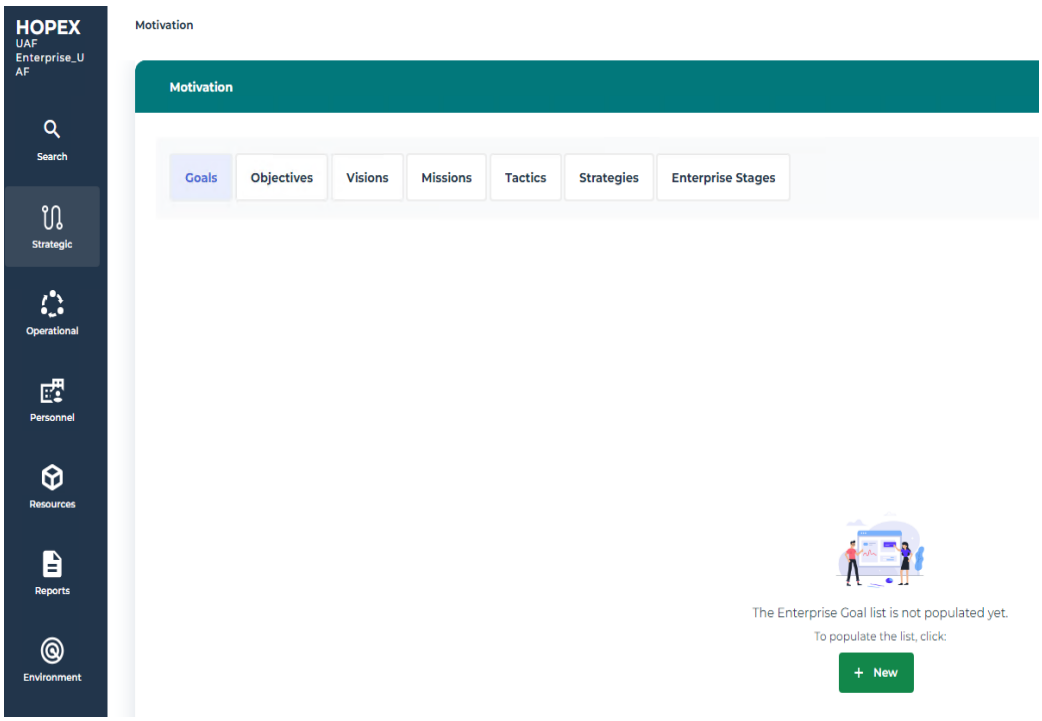
To get more information about the use of folders in **HOPEX** trees, see **HOPEX**, "Handling Trees" chapter of the **HOPEX Common Features** guide.

The hierarchy navigation enables to navigate on the main concepts of the view:

- Capability map, see Describing a Capability Map,
- Capability,see Describing a Capability,
- Business Dictionary,Business Dictionary
- Enterprise goal, see Creating Enterprise goals,
- Enterprise objective,see Creating Enterprise Objectives,
- Vision, see Creating Visions,
- Exhibited capabilities,Managing Exhibited Capabilities
- Stages, see Defining transformation stages.

STRATEGIC - MOTIVATIONS

The **Motivation** aspect of the **Strategic** view helps UAF Architect to manage goals and objectives that are motivated by drivers and opportunities that enable them.



Defining Enterprise strategic elements

Strategic elements of an enterprise are classified in the following categories:

- Ends, see: Identifying enterprise ends,
- Means, see: Defining Means.
- The transformation stages, see Defining transformation stages.

Identifying enterprise ends

Creating Enterprise goals

The *enterprise goals* are determining elements in your enterprise model since they interconnect the ends of the enterprise transformation with the objectives of the transformation stages.

> 🕮 *A Goal tends to be longer term, and defined qualitatively rather than quantitatively. It should be sufficiently narrow-focused that Objectives can be defined for it.*

To create an *enterprise goal*:

1. From the navigation menu, select **Strategic > Motivation**.
2. Click the **Goals** tab.
3. Click **New** button.
   The **Enterprise Goal** is created.

The **Characteristics** property pages of an enterprise goal is used to access:

- its **Name**,
- its **Owner**, by default the current enterprise.
- the **Comment** text.
- the **Supported Visions** section, which specifies the **Visions** connected to the enterprise goal.

> ☛ *For further details, see Creating Visions.*

Creating Enterprise Objectives

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

To create an *Enterprise Objective*:

1. From the navigation menu, select **Strategic > Motivation**.
2. Click the **Objectives** tab.
3. Click **New** button.
   The **Enterprise Objective** is created.

The **Characteristics** property page of an enterprise objective is used to access:

- its **Name**,
- its **Owner**, by default the current transformation stage.
- the **Comment** text.
- the **Supported Goals** section provides the list of *Enterprise goals* covered by the objective.

> ☛ *For further details, see Creating Enterprise goals.*

- The **Supported Objectives** section provides the list of *Enterprise objectives* to satisfy.

Creating Visions

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

To create a *Vision*:

1. From the navigation menu, select **Strategic > Motivation**.
2. Click the **Visions** tab.
3. Click **New** button.
   The **Vision** is created.

The **Characteristics** property page of a vision is used to access:

- its **Name**,
- its **Owner**, by default the current transformation stage.
- the **Comment** text.

Defining Means

To ensure that the *strategies* and tactics implemented in the enterprise correspond to the enterprise goals, you can use **HOPEX IT Business Management** to align the objects representing the ends of the strategy with the means to be implemented.

> 🕮 *A Strategy is one component of the plan for the Mission. A Strategy represents the essential Course of Action to achieve Ends (Goals in particular). A Strategy usually channels efforts towards those Goals. A Strategy is accepted by the enterprise as the right approach to achieve its Goals, given the environmental constraints and risks.*

To check the consistency of the transformation plan, each *strategy* is connected to an *enterprise goal*.

Creating a Mission

> 🕮 *A mission indicates the ongoing operational activity of the enterprise. The Mission describes what the business is or will be doing on a day-to-day basis. A Mission makes a Vision operative; that is, it indicates the ongoing activity that makes the Vision a reality. A Mission is planned by means of Strategies.*

To describe a *Mission*:

1. From the navigation menu, select **Strategic > Motivation**.
2. Click the **Missions** tab.
3. Click **New** button.
   The **Mission** is created.

The **Characteristics** property page of a Mission is used to access:

- its **Name**,
- its **Owner**, by default the current transformation stage.
- the **Comment** text.

The **Ends Support & Means Refinement** property page of a Mission provide access to:

- the **Supported Ends** section provides the list of *Visions* supported by the Mission.

  ☛ *For further details, see Creating Visions.*

- The **Refining Means** section provides the list of *Strategies* to use in the context of the Mission.

  ☛ *For further details, see Creating a Strategy.*

Creating a Tactic

> 🕮 *A Tactic is a Course of Action that implements part of a Strategy. Tactics generally channel efforts towards Objectives.*

To create a *Tactic*:

1. From the navigation menu, select **Strategic > Motivation**.
2. Click the **Tactics** tab.
3. Click **New**.
   The **Tactic** is created.

The **Characteristics** property page of a Tactic is used to access:

- its **Name**,
- its **Owner**, by default the current transformation stage.
- the **Comment** text.
- the list of **Strategies** that the tactic implements in the **Supported Strategies** section.

> ☛ *For further details, see Creating a Strategy.*

The **Ends Support & Means Refinement** property page of a Mission provide access to:

- the **Supported Ends** section provides the list of *Objectives* covered by the Tactic.

> ☛ *For further details, see Creating Enterprise Objectives.*

Creating a Strategy

> 🕮 *A Capability Map is an assembly of Capabilities and their Dependencies that, together, provide a capability scope for an Enterprise Stage.*

To create a *strategy*:

1. From the navigation menu, select **Strategic > Motivation**.
2. Click the **Strategy** tab.
3. Click **New**.
   The **Strategy** is created.

The **Characteristics** properties page of the strategy provides access to:

- its **Name**,
- Its **Enterprise**,
- the **Comment** text.
- the **Business Objective Contribution** can be an *Objective* or a *Goal.*

> ☛ *For further details, see Identifying enterprise ends.*

- the list of **Supported Missions**.

> ☛ *For further details, see Creating a Mission.*

Defining transformation stages

The implementation of an *enterprise* is described by the *enterprise stages* that correspond to its state at a given time.

> 🕮 *An Enterprise is a purposeful undertaking, conducted by one or more organizations, aiming at delivering goods and services, in*

accordance with the enterprise mission in its changing environment. During its development over time, an enterprise has to adapt to its environment and sets up transformation goals and objectives along with course of action to achieve these objectives. The design and realization of the resulting transformation stages may transcend organizational boundaries and consequently require an integrated team working under the direction of a governing body to involve stakeholders in transformation initiatives.

An Enterprise stage End Event is the end event of an Enterprise State.

Thus, when an enterprise is created, the following two *enterprise stages* can also been created:

- The current ('As-Is') stage that concerns existing elements;
- The target 'To-Be' phase that contains the target elements determined by the review of the transformation strategic goals.

The *Capability map* is associated to the enterprise and its transformation stages.

A Capability Map is an assembly of Capabilities and their Dependencies that, together, provide a capability scope for an Enterprise Stage.

The strategic elements of a transformation stage that follow:

- The enterprise objectives and the corresponding tactics, see Identifying enterprise ends,
- the exhibited Capabilities, see: Managing Exhibited Capabilities.

Each transformation stage is scheduled in the enterprise project depending on real or estimated dates. The scheduling is used to buid the enterprise transformation roadmap.

Creating a Transformation Stage

To create a *transformation stage* from an enterprise:

1. From the navigation menu, select **Strategic > Motivation**.
2. Click the **Enterprise Stages** tab.
3. Click **New**.
   A Transformation stage creation dialog box opens.
4. Specify the **Name** of the transformation stage.
5. Specify the **Period** of the transformation stage: As Is, Future or Passed.
6. Specify the **Begin Date** and the **End Date**.
7. Click **OK**.

Transformation stage properties

With **HOPEX UAF**, a transformation stage is described by several property pages.

The **Characteristics** property page of an enterprise stage provides access to the following information:

- **Name**,
- **Owner** the current enterprise,
- **Period**, As Is, Future or Passed. This attribute can be used if the dates are not specified.
- **Begin Date** of the phase,
- **End Date** of the phase,
- the **Description** text.
- the **Enterprise Stage Component** section,

The **Strategy** property page of an enterprise stage provides access to the following information:

- the **Ends** section that is used to access to the Ends involved in the transformation stage.

  ☛ *For more details on ends elements, see Identifying enterprise ends.*

- the **Means** section that is used to access to the Means involved in the transformation stage.

  ☛ *For more details on means elements, see Defining Means.*

- the **exhibited capabilities** page that is used to access to the Capabilities involved in the transformation stage.

  ☛ *For more details on strategic elements, see Managing Exhibited Capabilities.*

The **Architecture Description** page that is used to access to the architecture elements involved in the transformation stage.

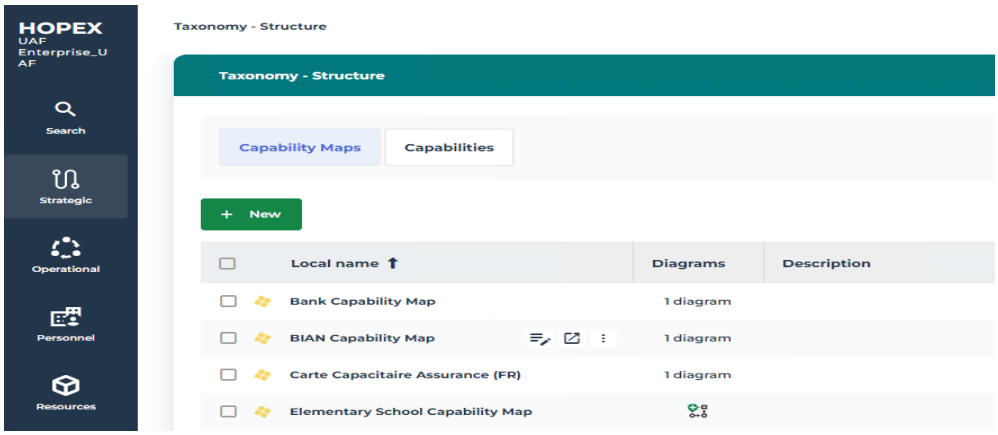
☛ *For more details on architecture elements, see Architecture Description.*

The **Timeline** page provides access to the Enterprise Stage Gantt.

☛ *For more details on this report, see Strategic - States.*

STRATEGIC- TAXONOMY - STRUCTURE

The **Taxonomy - Structure** aspect of the **Strategic** view helps UAF Architect to describe a structured list of the capabilities and sub-capabilities required for the vision established in the Enterprise descriptionl.

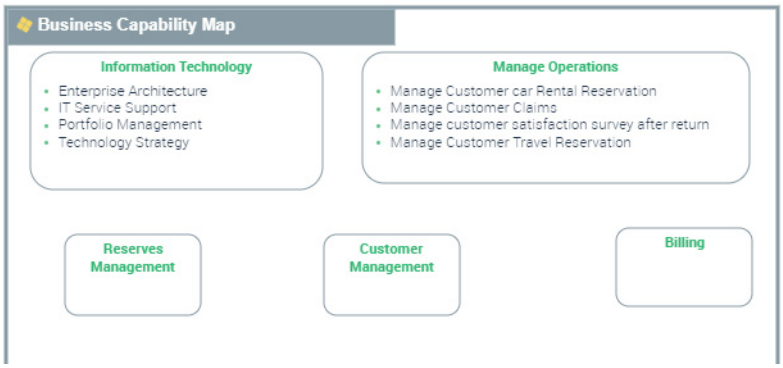


This list of capabilities is to be delivered for a particular timeframe. It has a hierarchical structure where capabilities are sometimes subdivided into sub capabilities and/or functions in order to provide clarity and the appropriate level of granularity required by subsequent processes in the capability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

Describing a Capability Map

A capability map describes what the enterprise is capable of producing for its internal needs or for meeting the needs of its clients. It is thus based on the main Capabilities of its activity at a given moment.

> 🕮 *A Capability Map is an assembly of Capabilities and their Dependencies that, together, provide a capability scope for an Enterprise Stage.*



Accessing capability maps

To access *Capability maps*:

1. From the **Strategic** navigation menu, select **Taxonomy - Structure**.
2. Click the **Capability Maps** tab.
   The list of the Enterprise Capability maps is displayed.

Creating a capability map

To create a *Capability map*:

1. From the **Strategic** navigation menu, select **Taxonomy - Structure**.
   The list of capability maps appears.
2. Click **New**.
   A creation dialog box opens.
3. Specify the **Name** and click **OK**.
   The new capability map appears in the list.

Properties of a capability map

The **Characteristics** properties page of the capability map provides access to:

- its **Owner**, by default on creation of the capability map, the current enterprise.
- its **Name**,
- the text of its **Description**,
- the Capability **Asset Category**. The proposed values are:
  - GDPR,
  - Operational,
  - Steering,
  - Support.

You can describe a capability map as follows:

- the **Structure** page is used to specify
  - the list of Capability components (compositions) that constitute the Capability map described.

    ☛ *You can only create capability components. You cannot reuse them. See Using the capability components*
  - the dependencies between these components,

    ☛ *For more details on capacity structure, see Defining Capability dependencies.*
- The **Fulfillments** page provides access to the components realizing the Capability map.

  ☛ *For more details on capacity fulfillment, see Capability Fulfillment.*
- The **Reporting** page provides access to the **Capability Map Breakdown**, see Reports on Capabilities.
- the **Diagram** page is used to access or to create Capability Maps diagrams.

  ☛ *For more details, see Creating a Capability Structure diagram.*

Creating a Capability Decomposition Tree diagram

A **Capability Decomposition Tree**, this diagram describes the tree structure of a Capability or a Capability map. Focusing on a particular Capability, this type of diagram enables summary representation of Capability breakdown into sub-Capabilities.

Creating a Capability Structure diagram

Using the capability components

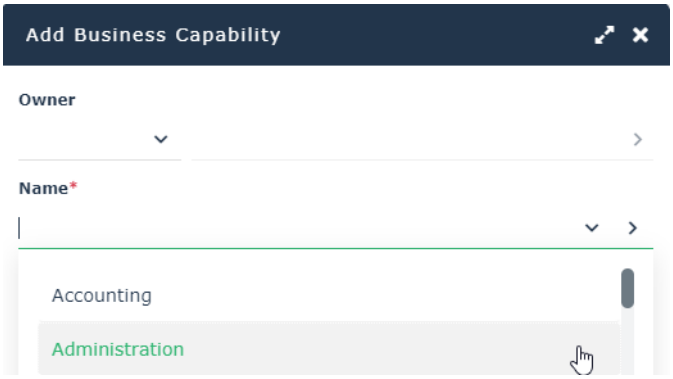
The components represented in a Capability map diagram are **Capability Components**. Each capability composition is associated with a Capability.

📖 *A **Capability component** is the involvement of a Capability in the context of a Capability map (one and only one) linked to an enterprise.*

To add a Capability composition to a Capability map diagram:

1. In the diagram insert toolbar, click **Capability Composition**.

2. Click in the frame of the Capability map.
   The creation window for a capability composition opens.
3. Click on the arrow associated with the **Name** field and select the Capability that interests you.



4. Click **OK**.
   The capability composition appears in the diagram.

   ☛ *If the Capability is associated with functionalities, they also appear. For more details on the capabilities and functionalities associated with Capabilities, see Defining the business skills and functionalities associated with Capabilities.*

Defining Capability dependencies

A dependency link between one capability composition and another is used to specify the elements on which this dependency is based.

For example, "Billing" uses "Order Management". Note that the expected result (business effect) of "Billing" is an "Invoice" and the expected result (business effect) of "Order Management" is a "Order to be delivered".

☛ *A single capability composition can have more than one dependency within a single diagram.*

To create dependency links between two capability compositions:

1. In the insert toolbar, click **Capability Dependency**.
2. Click the user component, and keeping the left mouse button pressed, move the cursor to the assembly used.
3. Release the mouse button.
   The capability composition appears in the diagram.

To enter the results concerned by a dependency between two Capability components:

1. Open the **Characteristics** properties dialog box.
2. Enter the user component result in the **Dependent Business Effect** field.

   For example, "Invoice".
3. Fill the **Desired Business Effect** field with the result of the component used.

   For example, "Order to be delivered".

Capability Map Reports

Two reports are available for a Capability Map:

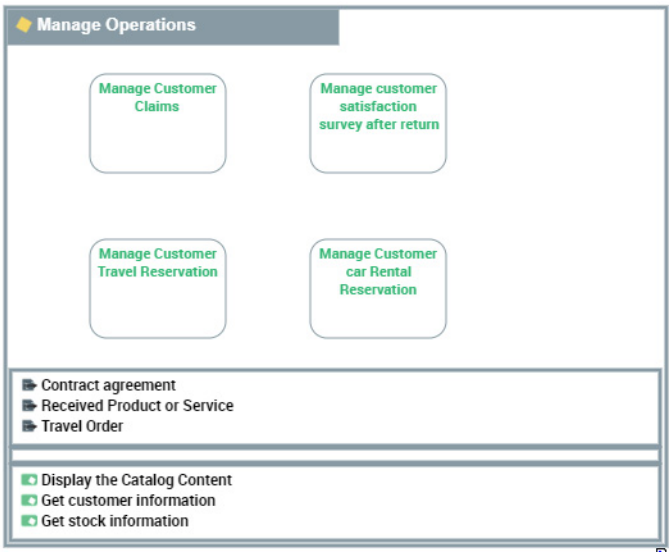
- a dendrogram illustrating the environment of the Capability Map.
- a breakdown analysis report.

> ☛ *For more details on the breakdown map of Capabilities, see The Breakdown Report of Capabilities.*

Describing a Capability

A Capability can be based on business sub-capabilities. The capabilities expected by the Capabilities described are the capabilities associated with each business sub-capability.

With **HOPEX UAF,** the capability structure diagram describes the composition of a Capability.



Creating a Capability

You can create a new Capability in several ways:

- From the Capability map diagram,
- From the properties page of a Capability map,
- From the navigation menu.

To create a *Capability* :

1. From the navigation menu, select **Strategic > Taxonomy - Structure**.
2. Click the **Capabilities** tab.
   The Capability list appears.
3. Click **New**.
   The new Capability appears in the list.

The properties of a Capability

The **Characteristics** property page of the Capability map provides access to:

- its **Owner**, by default on creation of the Capability, the current enterprise.
- its **Name**,
- the text of its **Description**.

With **HOPEX UAF** a Capability is described by the following pages:

- the **Structure** page specifies a part of the list of components that constitute the Capability described, as well as the dependencies between these components,

  ☛ *For more details on the structures of a Capability, see Using the capability components.*

- the **Expected Capabilities** page is used to specify a list of business skills and functionalities that are expected from the Capability.

  ☛ *For more details on the skills and functionalities associated with a Capability, see Defining the business skills and functionalities associated with Capabilities.*

- The **Fulfillments** page provides access to the components realizing the Capability.

  ☛ *For more details on capacity fulfillment, see Capability Fulfillment.*

- The **Reporting** page provides access to the **Capability Breakdown**, see Reports on Capabilities.
- the **Diagram** page is used to create Capability diagrams.

  ☛ *For more details, see Creating a capability structure diagram.*

`Creating a capability structure diagram`

To create a capability structure diagram:

1. Right-click the Capability that interests you and select **New > Diagram**.
2. Select **Capability structure Diagram**.
   The diagram opens in the edit area. The frame of the Capability described appears in the diagram.
   You can construct this diagram in tabular input mode.

   ☛ *Tabular input is available with the **HOPEX Web Front-End** module. For more information on using tabular input, see the "Entering a diagram in tabular mode" in the **HOPEX Common Features** guide.*

The components represented in a Capability structure diagram are **Capability Composition**. Each capability composition is associated with a Capability.

📖 *A **Capability component** is the involvement of a Capability in the context of a Capability map (one and only one) linked to an enterprise.*

☛ *For more details on how to use business components in a diagram, see Using the capability components.*

A dependency link between one capability composition and another is used to specify the elements on which this dependency is based, that is, the effect of one required by the effect of the other.

☛ *For more details on creating dependency links between two capability compositions, see Defining Capability dependencies:*

The capability compositions and their dependencies appear in the **Structure** property page of the Capability described.

Defining the business skills and functionalities associated with Capabilities

> 🕮 *A Business Skill is a Capability of an Organization (human resource) that has been acquired by training.*
>
> 🕮 *A Operational Activity Fulfillment follows the Class of Building Block Realization pattern and materializes the fulfillment of a Operational Activity by a BPMN Personnel Function, for instance.*

Each Capability is associated with functionalities that it is able to provide and that it needs to ensure its functionalities.

To associate a *skill* with a Capability:

1. Open the **Expected Capabilities** properties window of the Capability.
2. In the **Expected Business Skill** section, click **New**.
   An **Expected Business Skill** creation dialog box opens.
3. Click, for the **Connect a Business Skill** check box.
4. Specify the name of the skill.
5. Click **OK**.
   The business skill appears in the list of skills associated with the Capability.

> ☛ *For more information on business skills, see Personnel Constraints.*

To associate a *functionality* with a Capability:

1. Open the property pages of the Capability concerned and select the **Expected Capabilities** page.
2. In the **Expected Functionality** section, click **New**.
   An **Add functionality** window appears:
3. Click the down arrow.
4. Select a functionality.
5. Click **OK**.
   The functionality appears in the list of functionalities associated with the Capability.

The business skills, functionalities and the expected effects appear in the diagrams, at the bottom of the frame of the capability described.

A report is available to check the suitability between the Capability map and the operational environment, for more details, see Reports on Capabilities.

Capability Fulfillment

A capability can be implemented by different types of object such as a Operational Performer, Operational Domain, a Personnel Function or an Application.

To associate an Application to a Capability, you must create a capability fulfillment.

> 🕮 *A Fulfillment describes the relationship between a logical entity and a physical entity that implements it. The physical entity gives the list of logical entities that fulfill it.*

To specify that a Capability is fulfilled by an Application:

1. Open the **Fulfillments** property page of the Capability that interests you.
2. Click **New**.
   The Implementation creation dialog box opens.
3. Select **Business Capability Fulfillment** and click **Next**.
4. Select **Reuse objects**.
5. Select **Application**.
6. Select the Application that interests you and click **OK**.
   The Capability fulfillment appears in the list with the name of the selected Application.

Reports on Capabilities

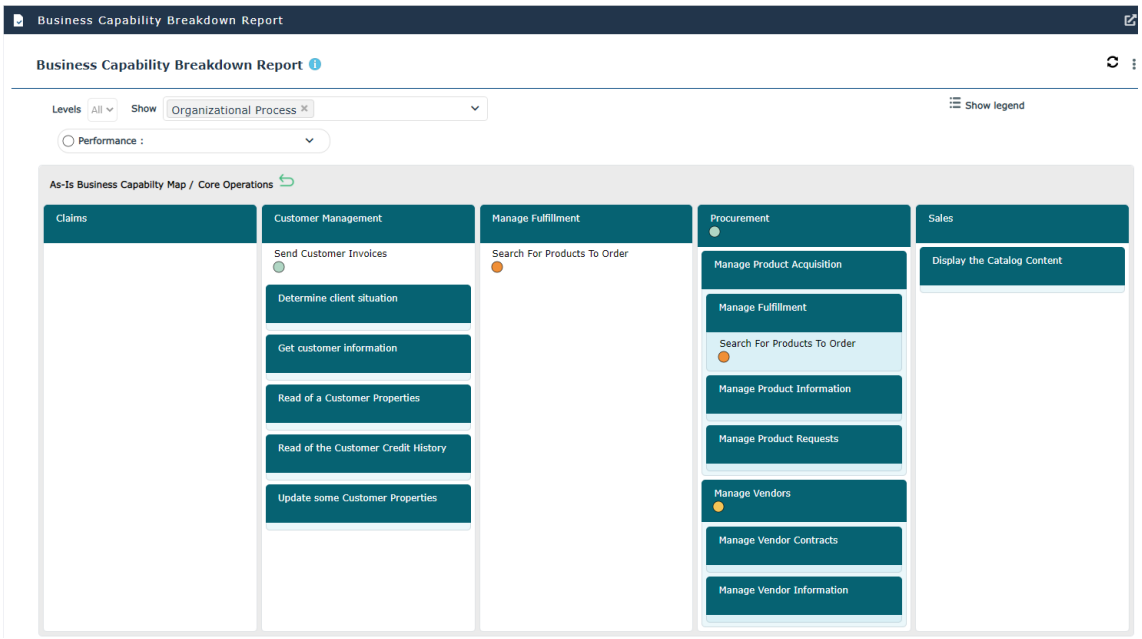
HOPEX UAF offers you the possibility to get a UAF Strategy Taxonomy report component report detailing the ownership between capabilities.

The Breakdown Report of Capabilities

You can use this report to display the realization coverage of Capability elements by operational elements such as logical and physical applications, application systems, etc.

The example below enables viewing of the coverage rate of the Capability map specified as parameters by processes.

☛ *For more details on Capability maps, see Describing a Capability Map.*



Example of a Capability map breakdown report

☛ *For more details on use of a breakdown report, see Handling a Breakdown Report*

☛ *For more details on use of a breakdown report, see the, chapter "Handling a Breakdown Report" in the **HOPEX Common Features** guide.*

Managing Exhibited Capabilities

📖 *An exhibited Capability is a Capability that is exhibited by an Enterprise Stage with quantified measure (KPI) and potential geopolitical scope (Site) for a defined market segment (Operational Partner).*

Accessing the list of exhibited capabilities

To access the list of exhibited capabilities of a transformation phase:

1. From the navigation menu, select **Strategic > Motivation**.

2. Click the **Enterprise Stages** tab.
3. Open the **Strategy** page of the Enterprise Stage that interests you.
4. Expand the **Exhibited Capabilities** section.
   The exhibited Capabilities list is displayed.

Creating an exhibited Capability

To create an *exhibited Capability* from an transformation stage:

1. Access the list of **Exhibited Capabilities** of the transformation stage that interests you. See Accessing the list of exhibited capabilities.
2. Click **New**.
3. Select the Capabilities that interests you and click **Add**.
   The exhibited Capability is created and appears in the **Exhibited Capability** column.
4. Select the exhibited Capability.
   The **Exhibited Capability** properties page opens on the right.
5. In the **Enterprise objectives** column, click the arrow to display the transformation stage objectives.
6. Select the objectives concerning the exhibited capability.
7. In the **Capability Realization** column, click the arrow to display the realizations of the current Capability in transformation stage context.

   ☛ *For more details on implementation of Capabilities, see Figure .*
8. Select the realizations concerning the exhibited capability.

   ☛ *The details of elements connected to an exhibited Capability are displayed in a transformation stage report, see .*

The properties of an exhibited Capability

The **Characteristics** property page of the exhibited Capability provides access to:

- its **Name**,
- its **Owner**, by default the current transformation stage,
- **Desired Capability Effect**, provides access to the exhibited Capability outcomes.

An exhibited Capability is described by the following pages:

- the **Structure** page specifies a part of the list of Capability components that constitute the exhibited Capability, as well as the dependencies between these components,

  ☛ *For more details on Capabilities components, see Using the capability components and Defining Capability dependencies.*
- the **Fulfillments** page provides access to the list of Components implement the Capability.

  ☛ *For more details on implementation of Capabilities, see Capability Fulfillment.*
- the **Transformation** page provides access to transformation stages connected to the exhibited Capability.

STRATEGIC - CONNECTIVITY

The **Connectivity** aspect of the **Strategic** view helps **UAF Architects** to define and analyze **Capability Dependencies**.

> 🕮 *A Capability Dependency is a relationship which asserts that a dependent Capability depends upon a needed capability in the context of a capability dependency owner, with respect to its needed effect (business outcome).*

Capability dependencies matrix

> 🕮 *A Capability Dependency is a relationship which asserts that a dependent Capability depends upon a needed capability in the context of a capability dependency owner, with respect to its needed effect (business outcome).*

This page lists **Capability Dependencies** together with the following information:

- dependency owner

  > ☛ *The dependency owner corresponds to the parent capability (as opposed to the capability components of this capability). It can be a Capability map.*

- capabilities
  - dependent capability
  - required capability
- effects
  - dependent effect
  - required effect

    > 🕮 *The effect is the business outcome required or desired by a capability.*

The name of the dependency is a combination of the name of the dependent capability and of the required capability. For example, [Detection -> Observation]. You cannot change it.

Creating a Capability dependencies matrix

> 🕮 *A Capability Dependency is a relationship which asserts that a dependent Capability depends upon a needed capability in the context of a capability dependency owner, with respect to its needed effect (business outcome).*

To create a **Capability Dependencies Matrix**:

1. From the **Strategic** navigation menu, select **Connectivity**.
2. Select the **Enterprise Stage**.
3. Select the **Capability**.
4. Click **Excel** to get an Excel file click **OK**.
   An Excel file is generated.

STRATEGIC - PROCESSES

The **Processes** aspect of the **Strategic** view shows the relationship between strategic phases and the Capabilities that are intended to be developed during the strategic phases, and the actual organizations involved.

This viewpoint provides a matrix tool to create realizations between organizations and capabilities.

Capability to Organizations Relationships Matrix Description

Matrix enabling creation of realization between organizations and capabilities.

- capabilities in rows

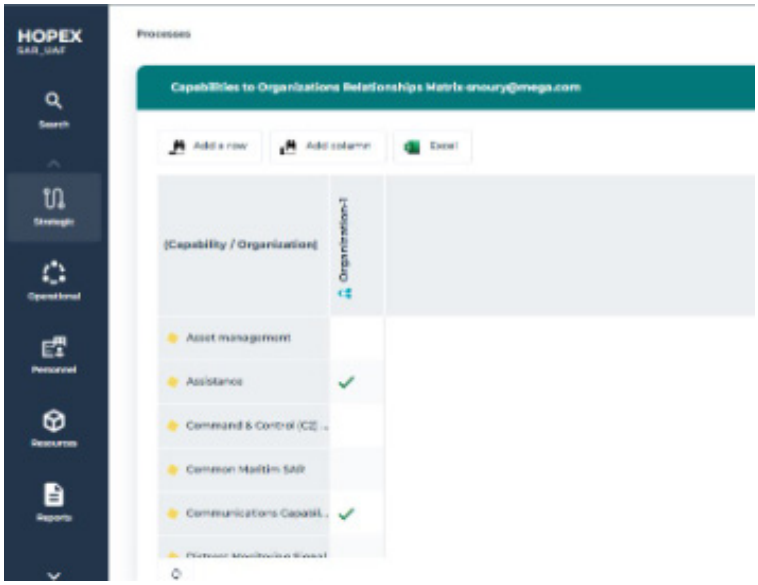
  📖 *A Capability is a set of features that can be made available by an enterprise.*

  ☛ *For more information, see: Transformation stage properties.*

- organizations in columns

  📖 *An organization represents a person or a group of persons that intervenes in the enterprise Personnel Functions or information system. An organization can be internal or external to the enterprise. An internal organization is an organizational element of the enterprise structure such as a department, a service, or a workstation. An internal organization 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 organization is an organization that exchanges flows with the enterprise. Example: Customer, Supplier, Government Office.*

  ☛ *For more information, see: Organizations.*



The matrix describes the mapping between capabilities and organizational resources that enable these capabilities.

To associate an Organization to a Capability, you must create a capability fulfillment. For more details, see Capability Fulfillment.

Creating the list of Capability to Organizations Relationships Matrix

To create a Capability to Organizations Relationships Matrix:

1. From the navigation menu, select **Strategic > Processes**.
   The list of of Capability to Organizations Relationships Matrix is displayed.
2. Click the **New** button.
   The new matrix is created.
3. Click the matrix to open it.
   The list of capabilities is displayed in the first column.
4. Click the **Add column** button to add Organizational Units.
   The list of Organizations of the Enterprise is displayed.
   ☛ *For more information, see: Organizations.*
5. Select the **Organizations** you want to add and click **OK**.

STRATEGIC - STATES

The **States** aspect of the **Strategic** view addresses the evolution of *Capabiliies* objects over the time using *State Machine* objects represented by a *State Machine diagram*.

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

The ***state machine diagram*** enables definition of the behavior of an object in response to internal or external requests it may receive. It indicates each possible object *state*, and the reaction of the object to a given event when in that state.

The ***state machine diagram*** describes the static aspects of an object: the different *states* it can be in and the possible *state tranistions*.

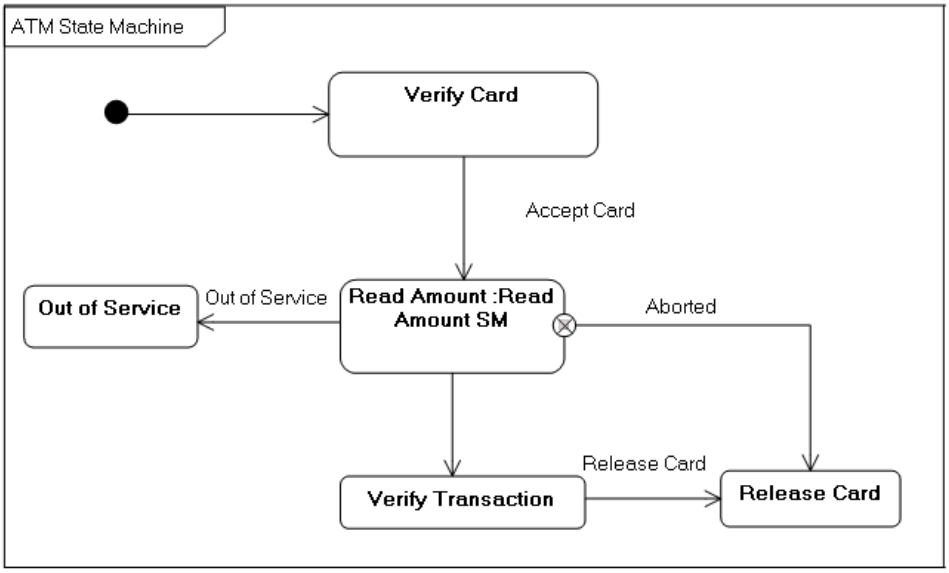
State Machine

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

The state machine diagram enables representation of the sequence of states that an object can take in response to interactions with the objects (internal or external to the studied system) in its environment.

Example of state machine diagram

The diagram below describes possible behaviors of an automated teller machine:



Creating a State Machine

To create a *State Machine* from the **Strategic** navigation pane.

1. Select **States**.
   The list of State Machines connected to Capability objects is displayed.
2. Click the **New** button.
   A creation wizard opens.
3. Enter the **Name** of the new State Machine.
4. Select (or create) a **Capability** in the **Subject of State Machine** box.
5. Click **OK**.
   The state machine is created.

With **HOPEX UAF**, a state machine diagram is created based on a *state machine*.

☛ *The diagram is initialized by creation of a region. A region is part of a composite state or state machine which contains states and transitions and of which execution is autonomous.*

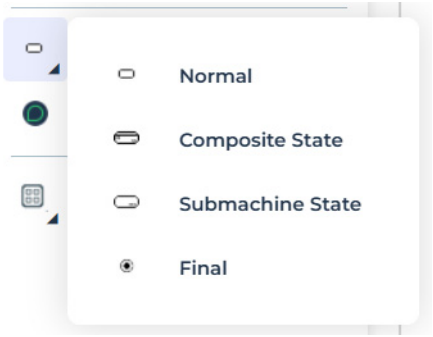
States

Creating a State

📖 *A state is a condition or situation in the life of an object, during which it satisfies some condition, performs some activity, or waits for some event. A state represents an interval of time delimited by two events. It is a phase an object passes through during its life cycle.*

To create a state in a state machine diagram:

1. Click the arrow associated with the **State** button of the object insert toolbar.



2. Select a ***state type***.

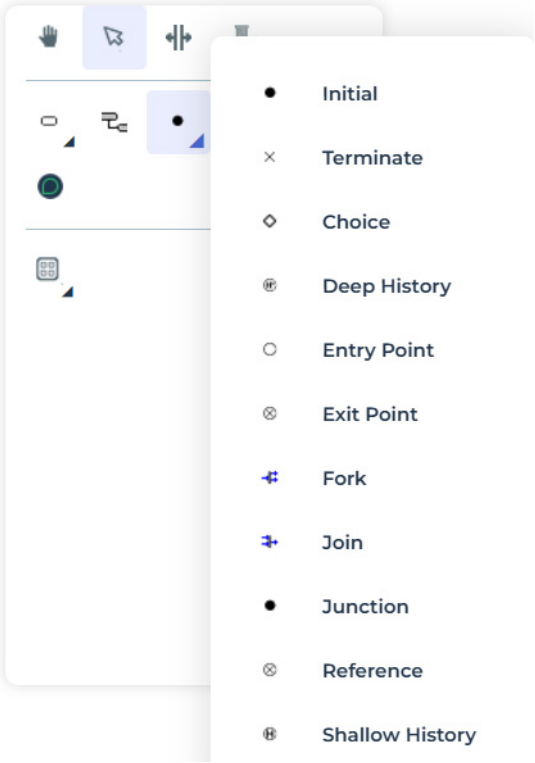
- A **normal** state: has no sub-structure.
- A **composite state**: comprises several states, described in the diagram.
- A **sub-machine state**: calls the descriptor of a state machine described elsewhere.
- A **final** state

3. Click in the diagram.
   The state appears in the diagram.

Pseudo-states

Pseudo-states are used to specify complex paths by combining several transitions between states.

They can be of different types: initial, final, choice, deep history, shallow history, input, output, fork, join, junction or reference.



- An ***Initial*** pseudo-state has a single output transition to the initial state of the object at its creation.
- A ***Deep History*** pseudo-state represents the last active configuration of a composite state containing it; that is the configuration that was active the last time the composite state was exited.
- A ***Simple history*** pseudo-state represents the most recent active sub-state of a composite state (without the sub-states of this sub-state).
- A ***Fork*** separates a transition into several concurrent transitions.
- A ***Join*** is the grouping of several transitions into a single transition.
- A ***Choice*** represents the choice of a transition between several possible transitions.
- A ***Junction*** is used to define paths of complex transitions between several states.
- An ***Input*** is an entry point of a state machine or of a composite state.
- An ***Output*** is an exit point of a state machine or of a composite state.
- A ***Reference*** to an input or output of a state machine or of a composite state.
- An ***Final*** is an input in this pseudo-state involves complete shutdown of the state machine.

State Transition

Passage from one node to another is represented by a *transition*.

> *A transition is passage of an object from one state to another. A transition is the response of an object to an event it receives. When an event occurs and certain conditions are satisfied, the object executes certain actions while still in the first state, before passing to the second state.*

All authorized transitions must be defined. Those that are not defined are prohibited.

Creating a Transition

To create an transition between two states in a State Machine Diagram:

1. In the state machine diagram, click **Transition**  in the insert toolbar.
2. Click the source state and drag the mouse to the target state.
3. Release the mouse button. The association is created.

Transition Types

A transition can be external, internal or local.

You can specify the transition type in the **Characteristics** property page of the transition.

- An *external transition* is a transition that modifies the active state.
- An *internal transition* enables an object to react to the arrival of an event that does not result in a state change but has an effect such as calling an operation or sending a message. For example, when pulling items from inventory, an item may not change state if the quantity remaining in the inventory is sufficient and does not fall below the reorder level or shortage level.
- A *local transition* applies to sub-states of a composite state. It can cause a change of state only within the composite state.

STRATEGIC - CONSTRAINTS

The **Constraints** aspect of the **Strategic** view is dedicated to the constraints imposed to *Capabilities*.

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

List of strategic constraints and reports

Three types of **Constraints** may be imposed to capabilities:

- Constraints, specified in the capability **Constraints** page,

  > ☛ *For more details on Constraints, see Constraints in the Strategic view*

- Mesurable properties, specified in the capability **Measurable Property** and **Qualifying Values** of the capability.

  > ☛ *For more details on Measurable Properties, see Describing a Measurable Property.*

- Directing Regulations, specified in the capability **Regulations** properties.
  - *For more details on Directing Regulation structure, see Security - Taxonomy.*

Three types of contraints reports are available:

- The Constraint Reports,
- The Measure Reports,
- The Politicy Reports.

Constraints in the Strategic view

To access the list of **Constraints** dedicated to Capabilities:

1. From the navigation menu, select **Strategic > Constraints**.
2. Click the **Constraints** tab to access the list of constraints.

To create a *Constraint*:

1. From the **Strategic** navigation menu, select **Constraints**.
   The list of the Constraints appears.
2. Click **New**.
   The new *Constraint* appears in the list.

Properties of a Constraint

The **Characteristics** properties of a constraint provides access to:

- the **Identification** section to get information about:
- the **Definition** section to get information about the constraint.
  - its **Owner**, by default it is the current enterprise.
  - its **Name**,
  - the text of its **Description**.

STRATEGIC - ROADMAP

The **Roadmap** aspect of the **Strategic** view addresses the deployment of capability(ies) to the organizations over the time.

Accessing the Roadmap Reports

To access the current **Enterprise Stages GANTT** report:

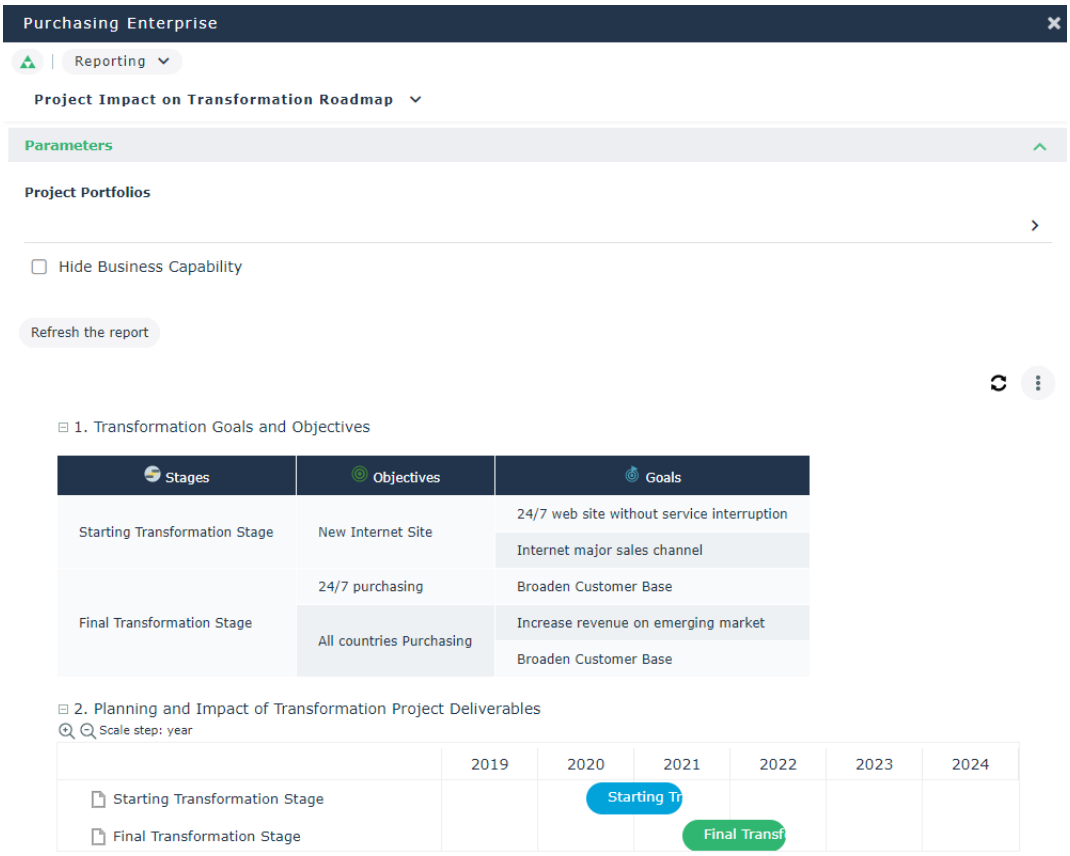
1. From the navigation menu, select **Strategic > RoadMap**.
2. Click the **Stages** tab to access the list of stages of the report.
3. Click the **Enterprise Stages Gantt** tab to access Enterprise Stage Gantt Report

Enterprise Stage Gantt Report

The **Quick Access** shortcut provides access to the current Enterprise Stage Gantt report.

☛ *For more details on Quick Access, see Quick Access Action.*

The transformation stages dates are presented in the colons of the dedicated report of the enterprise.



STRATEGIC - TRACEABILITY

The **Traceability** aspect of the **Strategic** view provides access to the **Business Capability to Transformation stages** matrixes and reports.

Those matrixes and reports are dedicated to the exhibited capabilites of transformation stages.

- Capabilities are presented in rows

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

- transformation stages are presented in columns

  > 🕮 *A transformation stage is a type of enterprise transformation stage aiming at the alignment of the enterprise operating model to its strategy and corresponding exhibited Capabilities.*

For more details on Traceability matrixes and reports, see Using Traceability Aspect.

For more details on Exhibited Capabilities, see Managing Exhibited Capabilities.

OPERATIONAL VIEWPOINT

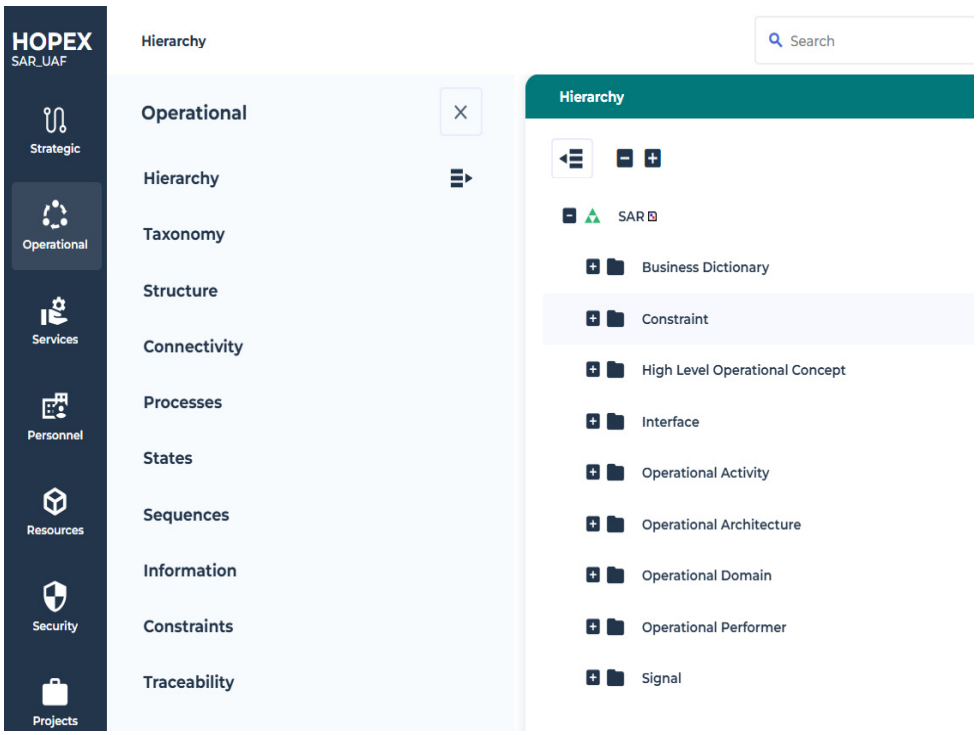
The **Operational** Viewpoint illustrates the Logical Architecture of the enterprise. It describes the requirements, operational behavior, structure, and exchanges required to support (exhibited) capabilities. It defines all operational elements in an implementation/solution ndependent manner.

This chapter comprises the following sections:

- ✓ Operational - Hierarchy;
- ✓ Operational - Taxonomy;
- ✓ Operational - Structure;
- ✓ Operational - Connectivity;
- ✓ Operational - Processes;
- ✓ Operational - States;
- ✓ Operational - Sequences;
- ✓ Operational - Information;
- ✓ Operational - Constraints;
- ✓ Operational - Traceability.

OPERATIONAL - HIERARCHY

The **Hierachy** aspect of the **Operational** view helps **UAF Architect** to manage operational components of the current Entreprise.



To get more information about the use of folders in **HOPEX** trees, see **HOPEX**, "Handling Trees" chapter of the **HOPEX Common Features** guide.

The hierarchy navigation enables to navigate on the main concepts of the view:

- Business Dictionary, see Business Dictionary,
- Constraint, see Operational - Constraints,
- High Level Operational Concept, see Using High Level Operational Concepts with HOPEX UAF,
- Interface, see Describing an Interface,
- Operational Activity, see Using Operational Activities,
- Operational Architecture, see Managing an Operational Architecture,
- Operational Domain, see Describing a Operational Domain
- Operational Performer, see Describing Operational Performers.
- Signal, see Using Signals.

OPERATIONAL - TAXONOMY

The **Taxonomy** aspect of the **Operational** view shows the taxonomy of types of Operational Agents.

HOPEX UAF enables you to draw diagrams freehand and connect the shapes to objects of the repository using **High Level Operational Concept**.

> 🕮 *A High Level Operational Concept is a rough draft of a model; unformal representation.*

High Level Operational Concept with HOPEX UAF

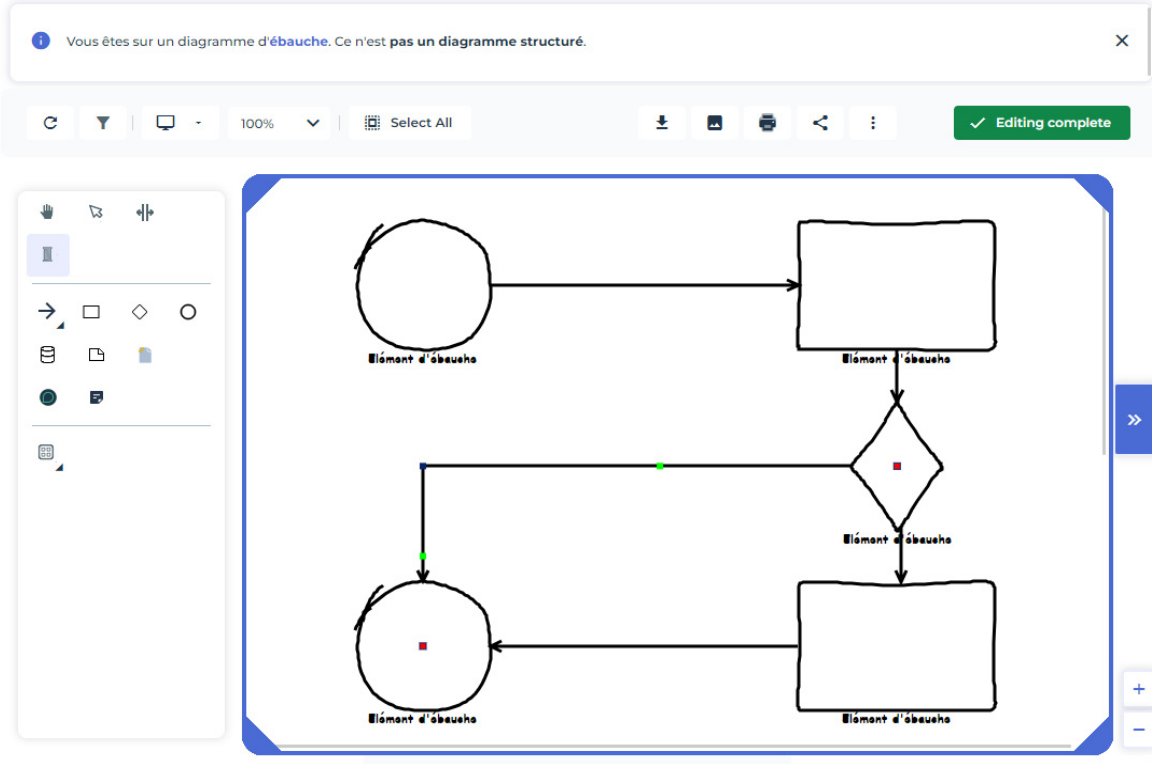
HOPEX UAF allows you to use drawings to create diagrams from shapes without needing to specify the type of object represented from the start. This feature is based on **HOPEX** facility based on Sketches and sketching diagrams. See Using High Level Operational Concepts with HOPEX UAF.

This feature makes teamwork easier by using sketches without any methodological or formalism constraints. Sketching diagrams can then be reworked and transformed into diagrams recognized by an **HOPEX** solution.

> ☛ *For more details ont the use of skecthing diagrams, see "Sketches" in the **HOPEX Common Features** guide.*
>
> ☛

A *Higth Level Operational Concept diagram* consists of *Concept Roles* connected by *Arbitrary Connectors*.

> 🕮 *A High Level Operational Concept is a rough draft of a model; unformal representation.*
>
> 🕮 *An Arbitrary Connector is a link between the two Concept Roles, represented in the form of a stroke.*
>
> 🕮 *A Concept Role part of a High Level Operational Concept represented as a shape.*



Available *Concept Roles* are:

- rectangles
- rombus
- circles
- barrels
- folded corner rectangles

You can connect *Concept Roles* with repository objects representing architecture elements.

☛ *For further details, see Connecting a Concept Role with a repository object.*

Using High Level Operational Concepts with HOPEX UAF

A High Level Operational Concept is a rough draft of a model; unformal representation.

An Arbitrary Connector is a link between the two Concept Roles, represented in the form of a stroke.

A Concept Role part of a High Level Operational Concept represented as a shape.

Accessing the list of High Level Operational Concepts

To access the list of **High Level Operational Concepts** with **HOPEX UAF**:

- Select **Operational > Taxonomy** navigation menu.
  The list of High Level Operational Concepts is displayed.

Creating a new High Level Operational Concept

To create a **High Level Operational Concept** with **HOPEX UAF**:

1. From the **Operational > Taxonomy** navigation menu, click **New**.
   The new High Level Operational Concept appears in the list.
2. (Optional) Modify its **Name**.

Attaching business documents to a High Level Operational Concept

You may need to attach an external document to the High Level Operational Concept (for example a Powerpoint file).

To do this:

1. In the properties of the **High Level Operational Concept** unfold the **Documentation** section.
2. In the **Business Document** tab, drop the file that interests you.

Using High Level Operational Concepts diagrams

A High Level Operational Concept is a rough draft of a model; unformal representation.

An Arbitrary Connector is a link between the two Concept Roles, represented in the form of a stroke.

A Concept Role part of a High Level Operational Concept represented as a shape.

Creating a High Level Operational Concept diagram

To create a High Level Operational Concept diagram with **HOPEX UAF**:

1. Select **Operational > Taxonomy** navigation menu.
   The list of High Level Operational Concepts is displayed.
2. Select the High Level Operational Concept that interests you and click **Create Diagram**.
   A dialog box opens.
3. Select **Create a High Level Operational Concept diagram**.
   The diagram appears.

> ☛ *To insert a Concept Role or a Arbitrary Connector in a High Level Operational Concept diagram, see "Using diagrams" chapter in **HOPEX Common Features** guide.*

Upload an existing picture in a High Level Operational Concept diagram

You can upload a png or jpg file in a High Level Operational Concept diagram and adding then other objects in your diagram.

To upload a png or jpg file in a High Level Operational Concept diagram with **HOPEX UAF**:

1. Select **Operational > Taxonomy** navigation menu.
   The list of High Level Operational Concepts is displayed.
2. Select the High Level Operational Concept that interests you and click **Create Diagram**.
   A dialog box opens.
3. Select **Upload image**.
   A dialog box opens to specify the file you want to upload.
4. Specify the file and click **Create High Level Operational Concept**.
   The diagram opens and the picture is inserted.

> ☛ *To insert a Concept Role or a Arbitrary Connector in a High Level Operational Concept diagram, see "Using diagrams" chapter in **HOPEX Common Features** guide.*

Connecting a Concept Role with a repository object

If a *Concept Role* represents an object already created in the repository, you can connect the *Concept Role* to the architecture element.

To do this:

1. Open the **Characteristics** page of the *Concept Role*.
2. Click the arrow at the far right of the **Represented Object** field and select **Connect**.
3. In the connecting information window, select the type of object you want to connect, then select the adequate object from the referential.
   The Concept Role appears in the form of the object type selected.
4. (Optional) In the **Appearance** section, uncheck the box **Use represented object representation**.
   The Concept Role reappears in the form of the object type selected.

> ☛ *If need be, you can connect another shape to the Concept Role.*

> ☛ *No check is made between the chosen shape and the type of the architectural element.*

OPERATIONAL - STRUCTURE

The **Structure** aspect of the **Operational** view defines operational architecture and exchange requirements necessary to support a specific set of Capability(ies).

One of the most important phases in describing an Operational Architecture is defining and understanding of the enterprise functional architecture.

The operational architecture enables the organization to understand, independently of its physical structure, which capabilities and skills it includes, those it needs, and how these contribute to its processes.

The description of the operational architecture also enables identification of areas of the organization where skills and Personnel Functions are duplicated and where synergies exist. These areas are not necessarily visible from the organizational structure.

Describing Operational Performers

📖 *An Operational Performer is a skill or grouping of skills of interest for the enterprise.*

📖 *A Operational Domain is a grouping of Operational Performers and their associated Operational Activities on the conjunction of two main criteria. 1. Their need in accomplishing one or more Capabilities. 2. The common skills and functionalities required to accomplish these Capabilities.*

Accessing the list of Operational Performers

To access the list of Operational Performers:

1. From the **Operational** navigation menu, select **Structure**.
   The list of *Operational Performers* connected to the current enterprise appears.

Operational Performer properties

The **Characteristics** properties page of a Operational Performer provides access to:

- its **Owner**, by default on creation of the Operational Performer, the current enterprise.
- its **Name**,
- the text of its **Description**.
- its **Owned Realizations**

☛ *For more details on fulfillments, see Capability Fulfillment.*

With **HOPEX UAF**, an operational performer is described by the following pages:

- the **Required Abilities** page is used to specify a list of skills and functionalities required by the business.

  ☛ *For further details, see Personnel Constraints.*

- the **Usage** page, which provides access to the components owning the current object.
- the **Performed Process** page, which provides access to the operational activities executed.
- The **Reporting** page provides access to the **Realizations of an architecture building block and its components**,

  ☛ *For more details on use of a breakdown report, see Handling a Breakdown Report*

- the **Diagram** page is used to manage the Operational Performer diagrams:
  - Operational Performer Structure Diagram,
  - Operational Performer Tree Diagram,
  - Scenario of Operational Flow Diagram.

Operational Performer Structure Diagram

The **Operational Performer Structure diagram** describes the components of the Operational Performer and the connections between components. It thus describes:

- the *Operational Roles*,

  🕮 *An Operational Role represents the configuration of a Operational Performer within another one.*

- access points: *service points* and *request points*.

  🕮 *A service point is a point from which a system receives a request from another system and provides the requested service.*

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

  ☛ *For more details on service points and request points, see Describing Service and Request Points.*

- *service interactions*

  🕮 *A Service Interaction represents an interaction for service purpose between entities in a specific context inside or outside a company. These entities can be enterprise org-units, applications, activities or Personnel Functions, as well as external org-units. The content of this interaction is described in an Interface.*

  ☛ *For more details on service interactions between components, see Creating a Service Interaction from a diagram.*

Operational Performer Tree Diagram

The **Operational Performer Tree diagram** describes the tree structure of the Operational Performer. Such diagram describes the current Operational Performer and its sub-Operational Performers.

Managing an Operational Architecture

A Operational Architecture represents the relationships of a Operational Domain with its partners.

Accessing the list of Operational Architectures

To access the list of *Operational Architectures*:

1. From the **Operational** navigation menu, select **Hierarchy**.
2. Expand the current enterprise folder, as well as its **Operational Architecture** folder.
   A list of the Operational Architectures appears.

Creating a Operational Architecture

To create a *Operational Architecture*:

1. In the **Operational** navigation menu, click **Hierarchy**.
2. From the **Operational Architecture** folder, click **New > Operational Architecture**.
3. Enter its **Name** and click **OK**.
   The new Operational Architecture appears in the list.

The properties of a Operational Architecture

The **Characteristics** properties page of the Operational Architecture provides access to:

- its **Name**,
- its **Owner**, by default on creation of the Operational Architecture, the current enterprise.
- the text of its **Description**.

With **HOPEX UAF**, an Operational Architecture is described by the following property pages:

- the **Structure** page which provides access to the list of components of the Operational Architecture.

  For more details on the components of the Operational Architecture, see Operational Architecture diagram.

- the **Implementation** page, which provides access to the list of environments (Capacity Configuration, organization, applications, application systems or logical application systems) that implement the Operational Architecture.
- the **Diagram** page is used to manage the Operational Architecture diagrams:
  - Operational Architecture diagram,
  - Scenario of Operational Environment Flows Diagram.

Operational Architecture diagram

A Operational Architecture diagram describes the service interactions between the main internal components of the environment described and the external components. It thus describes:

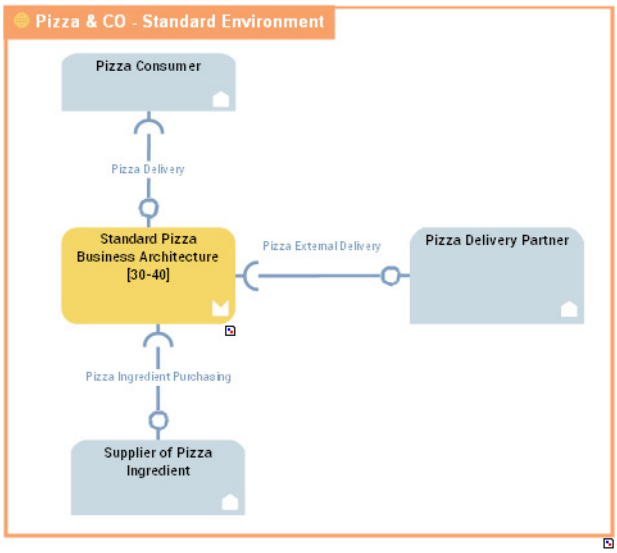
- the internal and external Operational Domains Role,

  An Operational Domain Role is the usage of an Operational Domain which is internal to the considered environment.

- the Operational Partners Roles,

  An Operational Partner is a conceptual stakeholder type that interacts with the enterprise in the context of the enterprise Operational Architecture. Examples : 'Private Sector Customer', 'Regulation Authority', 'Supplier'.

In this example, the Operational Performer architecture environment of company is made up of the historical Operational Performer architecture and its interactions with external partners: clients and suppliers. You can see in the diagram that delivery is outsourced to a third party deliver partner.



Communications between the objects are represented by service interactions that represent requests and service provision.

A Service Interaction represents an interaction for service purpose between entities in a specific context inside or outside a company. These entities can be enterprise org-units, applications, activities or Personnel Functions, as well as external org-units. The content of this interaction is described in an Interface.

For more details on service interactions between components, see Creating a Service Interaction from a diagram.

Describing a Operational Domain

A Operational Domain is a grouping of Operational Performers and their associated Operational Activities on the conjunction of two main criteria. 1. Their need in accomplishing one or more Capabilities. 2. The common skills and functionalities required to accomplish these Capabilities.

Accessing the Operational Domain list

To access the Operational Domain list:

1. From the **Operational** navigation menu, select **Hierarchy**.
2. Expand the current enterprise folder, as well as its **Operational Domain** folder.
   The tree of Operational Domains appears.

The properties of a Operational Domain

The **Characteristics** properties page of a functional area provides access to:

- its **Name**,
- its **Owner**, by default on creation of the Operational Domain, the current enterprise.
- the text of its **Description**.
- its **Owned Realizations**

  ☛ *For more details on fulfillments, see Capability Fulfillment.*

With **HOPEX UAF**, a Operational Domain is described in the following pages:

- the **Structure** page, which provides access to the list of components of the Operational Domain.

  ☛ *For more information on the components of a Operational Domain, see Operational Domain diagram.*
- the **Performed Process** page, which provides access to the Operational Activities executed in the context of the Operational Domain.

  ☛ *For more details on Operational Activities, see Using Operational Activities.*
- the **Diagram** page is used to manage the Operational Domain diagrams:
  - Operational Domain diagram,
  - Scenario of Operational Flow Diagram.

Operational Domain diagram

A Operational Domain diagram describes the service interactions between the main internal components of the architecture described. It thus describes:

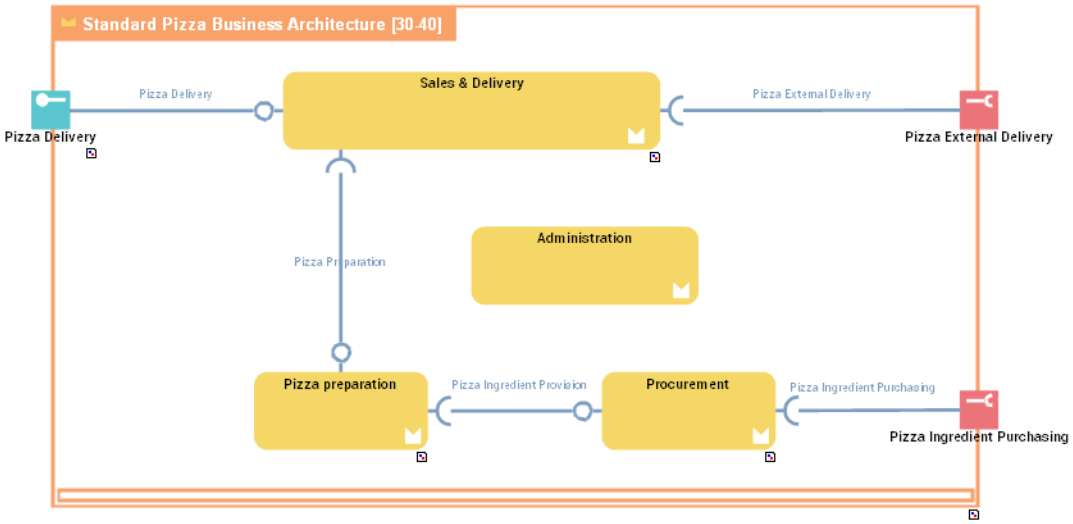
- the *Operational Domain Roles*,

  An Operational Domain Role is the usage of an Operational Domain which is internal to the considered environment.

  In this example, the history Operational Domain is based on the Operational Domains for selling, delivering and command.

- the *Operational Roles*.

  An Operational Role represents the configuration of a Operational Performer within another one.



With **HOPEX UAF**, communications are based on:

- access points: *service points* and *request points*.

  A service point is a point from which a system receives a request from another system and provides the requested service.

  A request point is a point by which an agent requests a service from potential suppliers.

  ☛ *For more details on service interactions between components, see Describing Service and Request Points.*

- *service interactions*

  A Service Interaction represents an interaction for service purpose between entities in a specific context inside or outside a company. These entities can be enterprise org-units, applications, activities or Personnel Functions, as well as external org-units. The content of this interaction is described in an Interface.

  ☛ *For more details on service interactions between components, see Creating a Service Interaction from a diagram.*

OPERATIONAL - CONNECTIVITY

The **Connectivity** aspect of the **Operational** summarizes logical exchanges between Operational Performers of information, systems, personnel, energy etc. and the logical activities that produce and consume them. Measurements can optionally be included.

Using Signals

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

Accessing Signals

To access the list of *Signal*:

1. From the **Operational** navigation menu, select **Connectivity**.
2. Click the **Signals** tab.
   The list of the Signals is displayed.

Creating a Signal

To create a *Signal*:

1. From the **Operational** navigation menu, select **Connectivity**.
2. Click the **Signals** tab.
   The list of the Signals is displayed.
3. Click **New**.
   A creation dialog box opens.
4. Specify the **Name** and click **OK**.
   The new Signal appears in the list.

   ☛ *You can also creating a signal when you create a Flow.*

Properties of a Signal

The **Characteristics** properties page of the Signal provides access to:

- its **Owner**, by default the owner is the current Enterprise.
- its **Name**,
- the text of its **Description**.
- the **Business Information Used** section provides access to the list of concept elements. See Concept.

Using Service Interactions

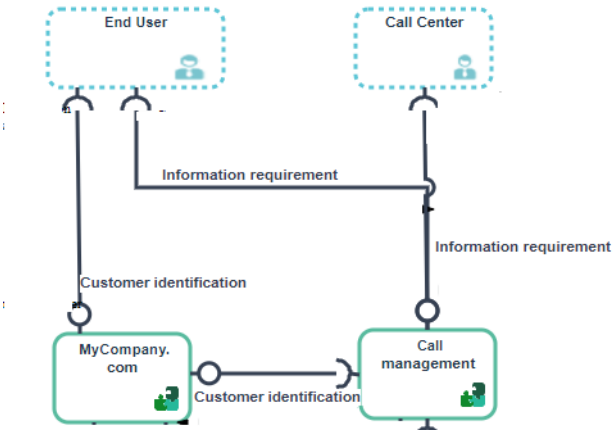
A *Service Interaction* represents the exchange of information between architecture components.

> *A Service Interaction represents an interaction for service purpose between entities in a specific context inside or outside a company. These entities can be enterprise org-units, applications, activities or Personnel Functions, as well as external org-units. The content of this interaction is described in an Interface.*

The content of a service interaction is described by an *interface*.

> *An Interface is a template of a contract between entities (organizational, IT …). The contract is described by available operations which can be triggered trough messages exchanged by roles (vendor, buyer..).*
>
> ☛ *For more details on interfaces, see Describing an Interface.*

In a "Purchasing Requests Processing" application system structure diagram, two interfaces are used by the different service interactions.



Adding a service interaction to an application system structure diagram

```
The clients must be identified before entering an order.
They can enter orders directly from "MyCompagny.com"
application or by using a Call Center. The Call Center uses
the "Call Management" application which uses the client
identification service offered by the "MyCompagny.com"
application.
```

Creating a Service Interaction from a diagram

To create a service interaction from a diagram:

1. In the objects toolbar for a diagram, click **Service Interaction**
2. Click the entity requesting the service and draw a link to the entity providing the service.

3. In the add service interaction dialog box, specify the interface you wish to use.

   ☛ *You can also use a new interface. For more details, see Creating an Interface.*

4. Click **Add**.

Describing Service and Request Points

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

📖 *A service point is a point from which a system receives a request from another system and provides the requested service.*

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

Service points

An application system, for example, is created to ensure one or more services. These services are represented by *service points*.

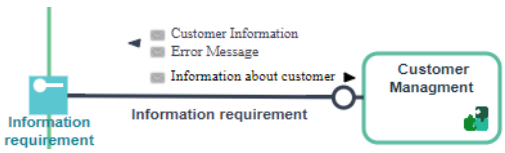
📖 *A service point is a point from which a system receives a request from another system and provides the requested service.*

The service is requested according to precise terms defined by an *interface* assigned to the service point.

📖 *An Interface is a template of a contract between entities (organizational, IT …). The contract is described by available operations which can be triggered trough messages exchanged by roles (vendor, buyer..).*

☛ *For further detail on interfaces, see Describing an Interface.*

Components activated to assure a service are linked to the service point by service interactions. If it is necessary to activate several components, you have to create several service interactions between the service point and the system components.



In the example presented here, the IT Service "Customer Management" is activated by the interaction service "Information request".

☛ *To create a service point, see Creating a Service Point or a Request Point.*

Request points

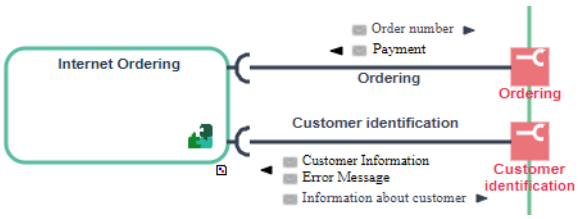
A *request point* enables representation of use of a service external to the described entity.

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

The service is requested according to precise terms defined by an Interface assigned to the request point.

> *An Interface is a template of a contract between entities (organizational, IT …). The contract is described by available operations which can be triggered trough messages exchanged by roles (vendor, buyer..).*
>
> ☛ *For further detail on interfaces, see Describing an Interface.*

Components that issue a request are linked to the request point by a service interaction.



In the example, request points represent requests for service executed by the "Email Order Management" IT service to identifier a customer and issue an order.

> ☛ *To create a request point, see Creating a Service Point or a Request Point.*

Creating a Service Point or a Request Point

The process for creating a *service point* or *request point* is identical.

> *A service point is a point from which a system receives a request from another system and provides the requested service.*
>
> *A request point is a point by which an agent requests a service from potential suppliers.*

To create a service point in a diagram:

1. In the diagram insert toolbar, click **Service Point** .
2. Position the object at the edge of the frame of the described object.
   A creation dialog box opens.
3. Click the arrow to the right of the **Interface** field to define the interface enabling activation of this service point, and select, for example, **Connect Interface**.
   A query window opens.
4. Select the interface associated with this service point and click **Connect**.
5. Click **Next**.
   A dialog box opens proposing a list of the interface roles that can be associated with the service point.

   > ☛ *This dialog box is not proposed if there is only one candidate role that can be associated with the service point.*
6. Select the role that interests you and click **OK**.
   The service point appears in the diagram.

Describing an Interface

An Interface represents the exchange of information between architecture components.

> 🕮 *An Interface is a template of a contract between entities (organizational, IT …). The contract is described by available operations which can be triggered trough messages exchanged by roles (vendor, buyer..).*

The content of a service interaction is described by an *interface*.

> 🕮 *A Service Interaction represents an interaction for service purpose between entities in a specific context inside or outside a company. These entities can be enterprise org-units, applications, activities or Personnel Functions, as well as external org-units. The content of this interaction is described in an Interface.*

An service contract is described by a sequence of operations which are represented:

- By *interfaces used*,

  > 🕮 *An Interface use is associated to an interface. It enables representation of complex exchanges.*

- Or by *service operations used*.

  > 🕮 *A service operation use represents the usage of a service operation in an service interface.*
  >
  > ☛ *For further detail on service operations, see Describing a Service Operation.*

Examples of Interface Diagrams (BPMN)

An Interface is described by a sequence of steps which are represented:

- By *service operations used*,
- By *Interfaces used*.

  > 🕮 *A service operation use represents the usage of a service operation in an service interface.*
  >
  > 🕮 *An Interface use is associated to an interface. It enables representation of complex exchanges.*

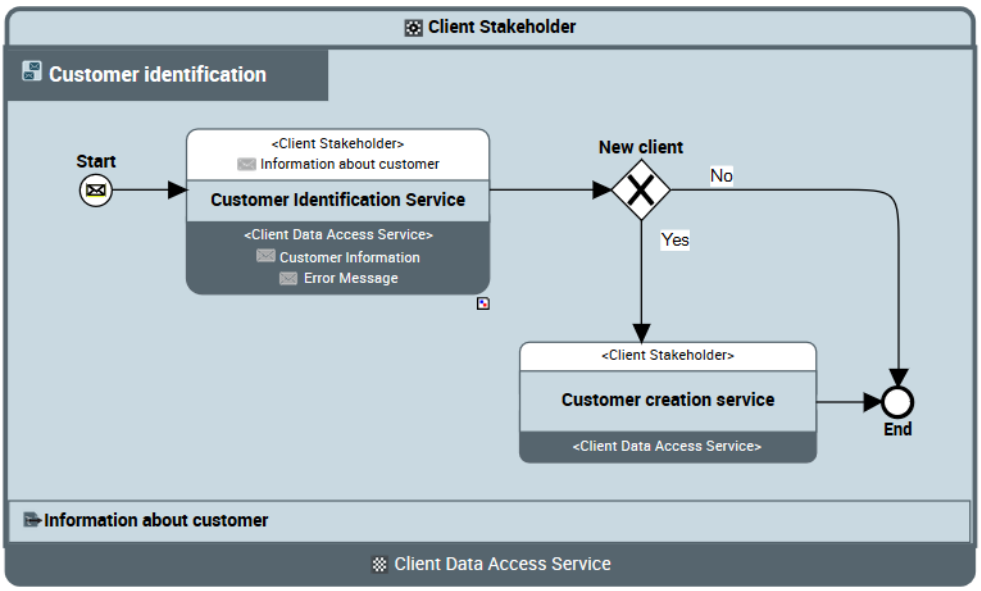
The Interface roles, presented at the border of the frame, represent participants:

- customer/supplier, or
- sender/recipient

An Interface can be described by involving more than two participants. In this case, a role is consumer of the Interface and the others are providers.

Example of Interface Diagram (BPMN)

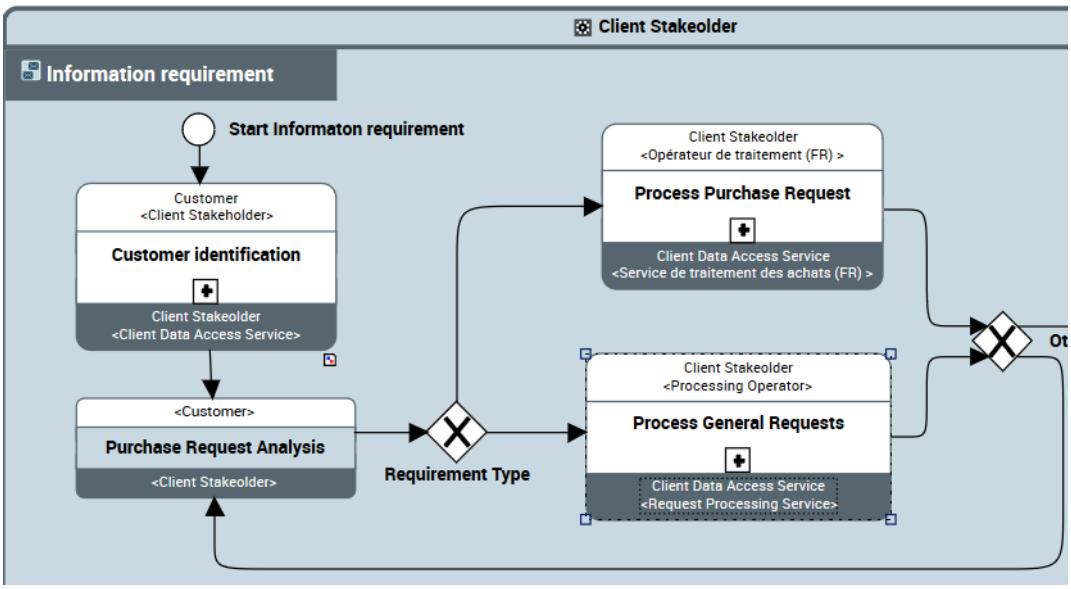
The Interface diagram associated with the "customer identification protocol " describes in BPMN formalism the operations executed.



Interface Diagram (BPMN) "Customer Identification"

Customer identification protocol starts with a customer identification step. If the customer is found the Interface returns customer information, if not, a "customer creation" Interface is activated.

Example of an advanced Interface communication



"Information Requirement" Interface diagram (BPMN)

The "Information Request" Interface is used by Center call center to take account of a customer request online. There are therefore three participants in this Interface: the customer, the IT applications and the customer representative who is the effective requester of the service (in this case the call center).

This Interface consists of identifying the customer, then analyzing the request. The request is then processed as a purchase request or as another request if it is an information request for example.

☛ *The **Roles** property page provides access to the list of contributor roles and to the initiator role of an Interface.*

Accessing the list of Interfaces

To access the list of Interfaces:

1. From the **Operational** navigation menu, select **Connectivity**.
2. Click the **Interfaces** tab.
   The list of the Interfaces is displayed.

Creating an Interface

You can create an Interface:

- from a library,
- from a diagram using service interactions, for example.

Whatever the point of origin, you can create Interface in standard mode or using an Interface *template*.

☛ *For more details on Interfaces, see Using an Interface Template.*

To create an Interface in standard mode, in a diagram, from an service interaction:

1. In the objects toolbar for a diagram, click **Service interaction**
2. Draw a link between the two communication entities.
3. In add service interaction window, click the arrow at the right of the field **Interface** and select **Create an Interface**.
   The creation window appears.
4. Select the **Creation Mode**: **Standard Creation**.

   ☛ *For more details on Interface template use, see Creating an Interface from an Interface template.*
5. Enter the Interface name in the **Name** field.
6. Click **OK**.
7. In the service interaction creation dialog box, enter the name of the service interaction using the name of the Interface and click **Add**.
   The service interaction and the Interface are created.

Properties of a Interface

The **Characteristics** properties page of the Interface provides access to:

- its **Owner**, by default the owner is the current Enterprise.
- its **Name**,
- the text of its **Description**.
- the text of its **Description**.
- the **Business Information Used** section provides access to the list of concept elements. See Concept.

Creating an Interface Diagram (BPMN)

An Interface is represented by an **Interface Diagram (BPMN)**.

To create an Interface Diagram (BPMN) from an interaction service:

1. From the **Environment** navigation menu, open the exploration area **Container > Libraries**.
2. Unfold the desired library, and then the **Interfaces** folder.
   The list of Interfaces accessible from the library appears.
3. Select the associated Interface and, in its pop-up menu, click **Create Diagram**
4. In the dialog box, select **Interface Diagram (BPMN)**
   The diagram opens with Interface frame and the two *roles* representing consumer and the supplier.

   A role is a participant in an interaction service, workflow or Personnel Function. It can be the initiator, that is the requester of a service, or it can represent a sub-contractor carrying out processing outside the service. A role is an integral part of the object that it describes, and is not reusable. It can subsequently be assigned to an org-unit internal or external to the organization or to an IT component. Examples: client, traveler.

The *events*, *gateways* and *sequence flows* of your diagram follow the BPMN standard.

> ☛ *For more details on events, gateways and sequence flows, see Managing events, gateways and sequence flows*

Defining a Service operation or an Interface

In an Interface diagram (BPMN), operations are described by:

- *Interfaces used*
- *Service operations used*

> 🕮 *An Interface use is associated to an interface. It enables representation of complex exchanges.*
>
> 🕮 *An Interface is a template of a contract between entities (organizational, IT …). The contract is described by available operations which can be triggered trough messages exchanged by roles (vendor, buyer..).*

To create a used *Interface used*:

1. Select the **Interface Used** button and click in the diagram within the Interface frame.
   The Interface appears in the diagram.
2. Open the **Characteristics** property page of the Interface.
3. Click the arrow to the right of the **Specification of an Interface used** box.
4. Select **Connect Interface** from the drop-down list and choose the Interface that you want to use.

> ☛ *The **Service operations** page provides access to the list of components of the exchange contract.*

Describing a Service Operation

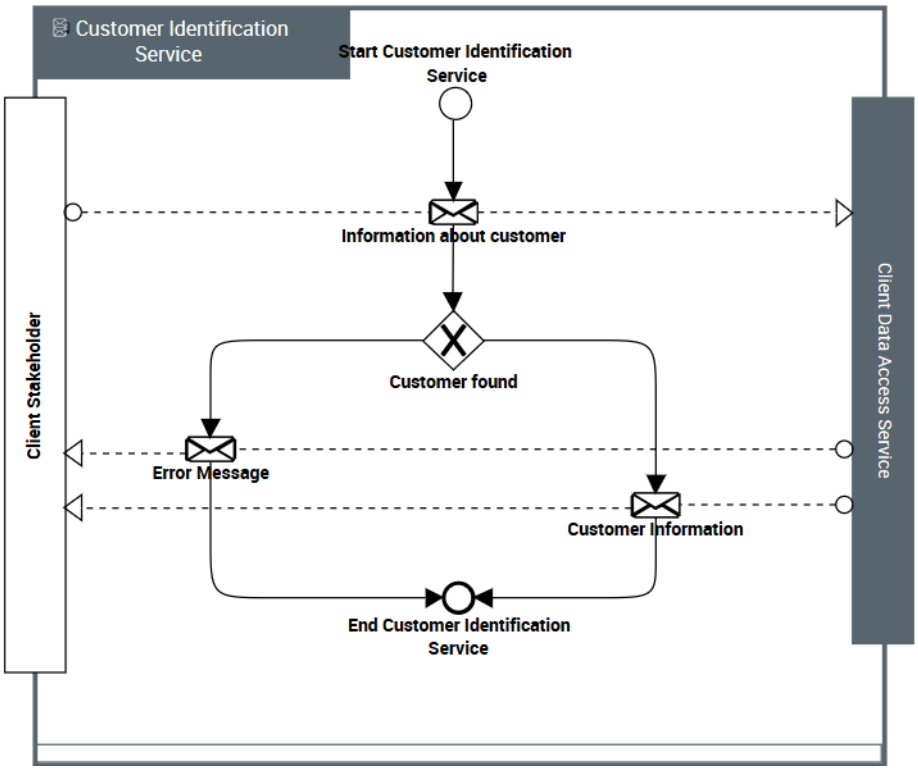
The content of a service interaction is described by an *Interface*.

> 🕮 *An Interface is a template of a contract between entities (organizational, IT …). The contract is described by available operations which can be triggered trough messages exchanged by roles (vendor, buyer..).*
>
> ☛ *For further detail on Interfaces, see Describing an Interface.*

An Interface is described by a sequence of service operations or Interfaces.

> 🕮 *A service operation specifies exchanges between participants.*

A service operation diagram describes the sequence flows of a *service operation*.



"Customer Identification Service" Service Operation Diagram

The customer identification service protocol begins by sending information enabling identification of the customer. An error message appears if the customer is not found, otherwise customer information is sent (customer identification, status of orders, etc.).

Accessing the list of service operations

To access the list of service operations of a library:

1. From the **Environment** navigation menu, open the exploration area **Container > Hierarchy**.
2. Unfold the desired library, and then the **Service operations** folder. The list of service operations accessible from the library appears.

Creating a service operation

You can create a *service operation* an Interface diagram (BPMN).

☛ *For more details on service operation templates, see Using an Interface Template.*

To create a *service operation* from an Interface diagram (BPMN):

1. Select the **Service Operation Used** button and click in the diagram within the Interface frame.

   A service operation use represents the usage of a service operation in an service interface.

   The service operation appears in the diagram.
2. Open the **Characteristics** property page of the service operation.
3. Click the arrow at the right of the **Service operation specification** and select **Create a service operation**.
   The Creation of Operation Joint Action dialog box opens.
4. Enter the **Name** of your service operation click **OK**.
   The service operation is automatically created.

Creating a Service Operation Diagram (BPMN)

A *service operation* is described by an service operation diagram presenting the sequence flow of messages exchanged.

To create an service operation diagram:

1. From the **Environment** navigation menu, open the exploration area **Container > Libraries**.
2. Unfold the desired library, and then the **Service operations** folder.
3. Select service operation that interests you and click **Create Diagram**.
4. In the dialog box, select **Service operation Diagram (BPMN)**
   The diagram opens. The frame of the service operation is positioned and the two roles (Consumer and Provider) are created.

Creating a message flow with signal

You must specify the *message flows* and their *signal* exchanged between the two service operation roles.

A Message Flow represents circulation of information within a service interface. A message flow transports its content.

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

To create a message flow and its signal:

1. In the service operation diagram, click the **Flow With signal** button.
2. Click the role that represents the message flow sender and, holding the mouse button down, draw a link to the message flow recipient.
   The **Creation of Flow** dialog box opens.
3. In the **signal** drop-down list, select the signal you wish to associate with the flow.
   The message flow is displayed with its signal in the diagram.

Managing events, gateways and sequence flows

"Start" and "End" *events* are required in description of the service assured by the Interface.

> *An Event represents a fact or an action occurring in the system, such as updating client information. It is managed by a broker. An application indicates that it can produce the event by declaring that it publishes it. If an application is interested in an event, it declares that it subscribes to the event.*

In compliance with the BPMN standard, in the object toolbar, several *gateway* types are available to you.

> *A Gateway represents elements that are used to control how sequence flows interact as they converge and diverge within a process.*

A *sequence flow* is a directional link that represents the chronological organization of the different processing steps.

> *A Message Flow represents circulation of information within a service interface. A message flow transports its content.*
>
> *For more details on events, gateways and sequence flows, see* *Managing events, gateways and sequence flows*

Using an Interface Template

Interface templates as well as *Service operation templates* and *signal templates* simplify the Interfaces creation by duplicating the components of the model used.

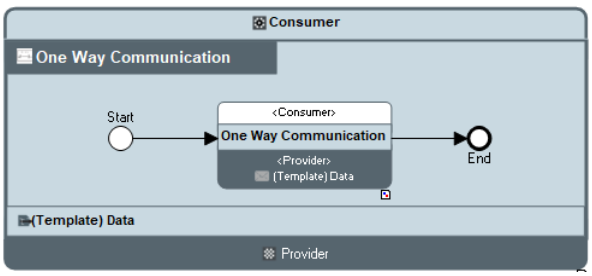
Then, the Interface be updated or modified.

Presentation of standard Interface Templates

Interface templates are provided to simplify the creation of your Interfaces. These Interfaces are supported by service operation templates.

Some *Interface templates* are provided with the solution.

The Interface template "One way communication"

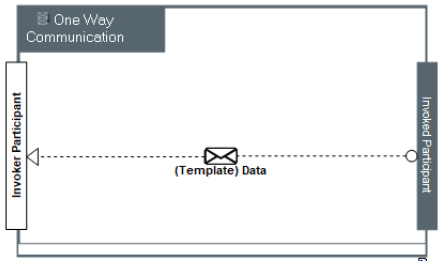


"One way communication" service interface template diagram (BPMN)

This *Interface* is based on an *service operation used* noted "One way communication" between the consumer and the provider.

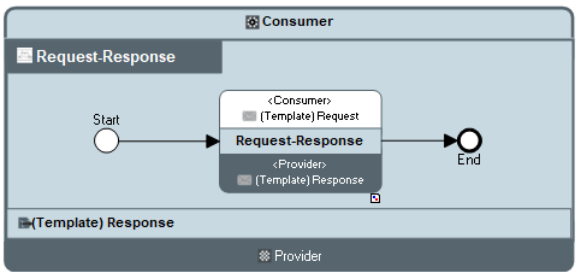
> *A service operation use represents the usage of a service operation in an service interface.*

The *service operation used* represents the signal "(Template) Data" exchanged between the consumer and the provider.



"One way communication" service operation diagram (BPMN)

The Interface template "Request-Response"

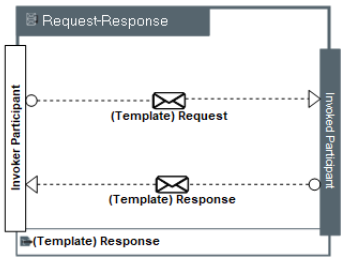


"Request-Response" service interface template diagram (BPMN)

This *Interface* is based on an *service operation used* noted "Request-Response" between the consumer and the provider.

> *A service operation use represents the usage of a service operation in an service interface.*

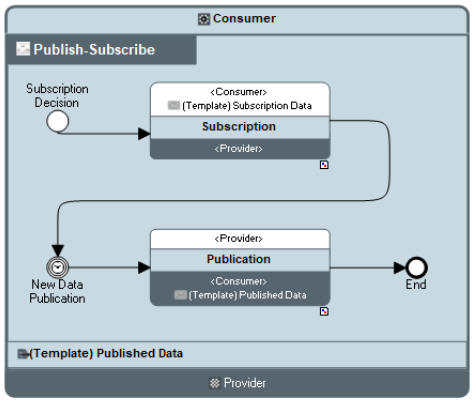
The *service operation used* represents the service operations of the signals "(Template) Request" and "(Template) Response" exchanged between the consumer and the provider.



"Request-Response" service operation diagram (BPMN)

This service operation represents the sending of a request signal and the sending of the response signal.

The Interface template "Publish-Subscribe"



"Publish-Subscribe" Interface Diagram (BPMN)

This *Interface* is based on a *service operation used* noted "Publish-Subscribe" between the consumer and the provider. The request for subscription is sent. An event represents the waiting time before the acceptance for publication.

Accessing the list of Interface templates

To access the list *Interface templates* of a repository:

- From the **Environnement > Hierachy > Libraries > MEGA > IT Architecture** navigation menu, select **Templates > Interfaces**.
  The list of Interface templates appears.

In the same way, to access to the list of *service operation templates*:

- From the **Environnement > Hierachy > Libraries > MEGA > IT Architecture** navigation menu, select **Templates > Service Operations**.
  The list of service operation templates appears.

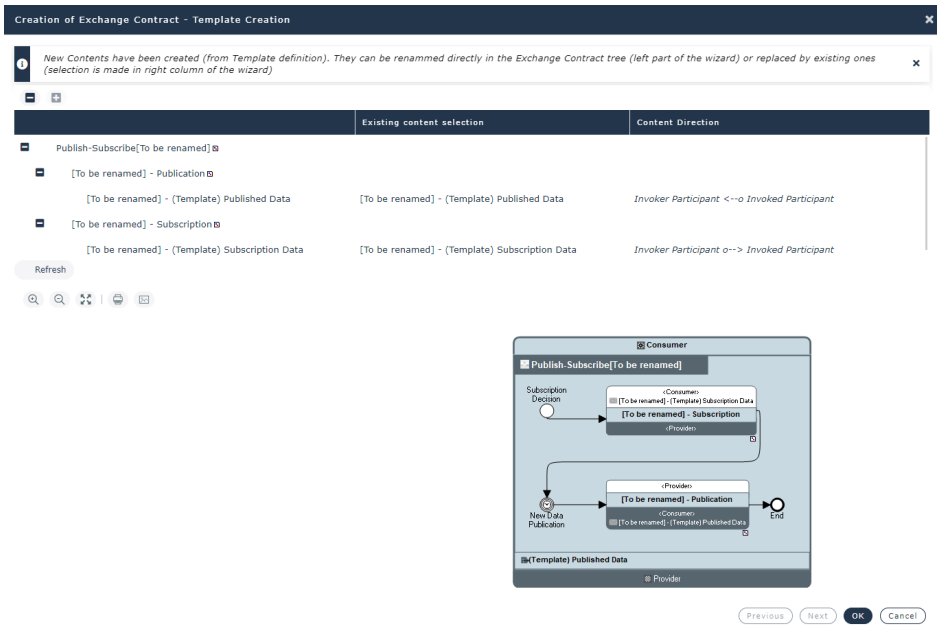
Furthermore, to access the list of *signal templates*:

- From the **Environnement > Hierachy > Libraries > MEGA > IT Architecture** navigation menu, select **Templates > Signal**.
  The list of signal templates appears.

Creating an Interface from an Interface template

To create an Interface from a list using an Interface template:

1. From the **Environment** navigation menu, open the exploration area **Container > Hierachy**.
2. From the library that interests you, create an Interface.
3. In the following dialog box, select the **Creation Mode**: **Template Based Creation**
4. Select the template that interests you and click **Next**.
   A dialog box displays the list of components of the Interface.



   The name of duplicated components is prefixed with "[To be renamed]".
   The signal templates used are duplicated.
5. Double-click the name you wish to modify.
6. (Option) In the **Existing signal selection** column, select the signal you want to reuse.
   As a consequence, the created signal ""[To be renamed]" is destroyed.

7. Click **OK**.
   The Interface is created.

   ☛ *Then you can change the Interface components, for example from its diagram, see Creating an Interface Diagram (BPMN).*

Creating an Interface Template

You can use an existing Interface to create Interface template.

To specify that an Interface is a template:

1. Select the Interface that interests you.
2. Open the **Characteristics** properties page.
3. Check the **Interaction Behavior Template** box.
   The Interface is added to the list of existing Interface templates.

The Interface template components declared as templates are duplicated when the Interface template is used.

To access the list of an Interface template components declared as template:

1. From the **Environnement > Hierachy > Libraries > MEGA > IT Architecture** navigation menu, select **Templates > Interfaces**.
2. Open the **Template Definition** property page of the Interface that interests you.
3. Check the **Template** box of the components to be duplicated.

Creating a Service Operation Template

To specify that a service operation is a template service operation:

1. Select the service operation that interests you.
2. Open the **Characteristics** properties page.
3. Check the **Interaction Behavior Template** box.
   The service operation is added to the list of existing service operation templates.

   ☛ *To access to the list of service operation templates: from the **Administration** navigation menu, select **Templates > Interfaces**.*

OPERATIONAL - PROCESSES

The **Process** aspect of the **Operational** view describes the activities that are normally conducted in the course of achieving business goals that support a capability. It describes operational activities, their Inputs/Outputs, operational activity actions and flows between them

📖 *An Operational Activity is an end-to-end collection of Activity Actions that creates an outcome for a customer, who may be the ultimate customer or an internal end-user of the Operational Activity.*

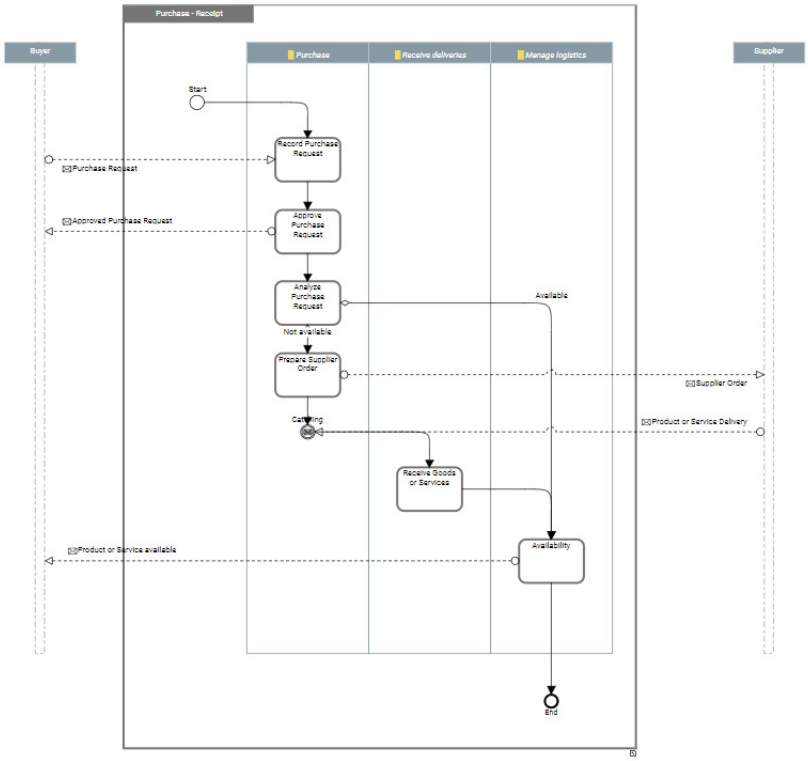
📖 *An Operational Activity Action is a distinct, identifiable phase or step within an Operational Activity that has a unique entrance criteria, exit criteria, and identifiable participating Operational Performer or Operational Domain.*

☛ *To display the **Operational Activities**, open the **Options** window and check that **Business Process and Architecture Modeling > Operational Activity Modeling** is activated.*

Operational Activity Example

In this Operational Activity diagram, the *Operational Performer* or the *Operational Domain* that create the Operational Activities are linked to the participants represented in columns.

The following diagram presents an example of a Operational Activity:



"Purchase reception" Operational Activity

The purchase request is recorded and must then be approved. The requester is informed of the approval or rejection of the request. If the request is validated, an analysis of the required order is carried out.

If stock is lower than a given threshold, an order is prepared and sent to the supplier for resupply.

If the product is available, or as soon as it is received from the supplier, it is made available to the requester.

In this example, the *Operational Performers* concerned are represented in columns.

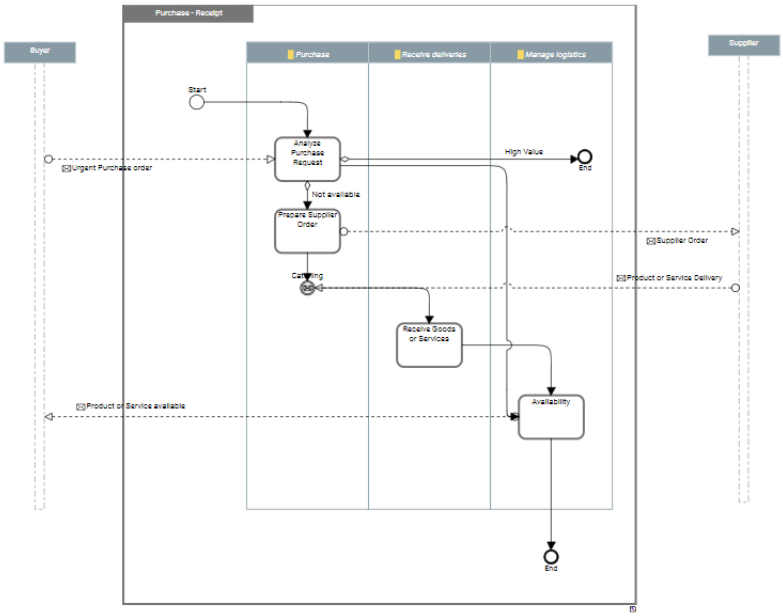
> *An Operational Performer is a skill or grouping of skills of interest for the enterprise.*

In the organization previously presented, three org-units: purchasing assistant, purchasing manager and buyer, systematically participate to execute the first four steps: record and approve the request, analyze and send the order.

Optimization of the Personnel Function "Process Purchase Requests" has saved one step: when amount of the order is not significant, the purchasing assistant can himself approve or refuse the purchase request.

In the case of urgent orders, you can again save steps by authorizing the purchasing assistant to send the order when the amount is not significant.

We obtain the following Operational Activity for processing of urgent orders:



"Purchase reception" Operational Activity

The first step consists of analyzing the purchase request. If the total amount is large, normal processing is carried out.

Otherwise, the availability request and a restock request are sent, if necessary. Continuation of this Operational Activity is identical to the previous one: when the order has been received, it is made available to the requester.

Operational Activity representation principles

Highlighting organizational choices

Each enterprise has activities related to its business that must be performed whatever the organization in place. These activities can be purchasing, sales, sales administration, manufacturing, etc.

Defining their organization consists of assigning these activities to the org-units that will perform them.

We can distinguish between:

- Processes relating to the business of the enterprise: these are difficult to change unless the enterprise decides to totally review its business.
- Processing depending on organizational choices.

Number of steps

Certain steps in a Personnel Function are exclusively linked to the chosen organization. In such cases, it is useful to check whether these steps provide any real added value to clients or only concern the way things are done.

Delivery times can also be reduced by restructuring the order of these steps.

To highlight possible improvements, you can represent a Operational Activities by flows exchanged between enterprise *Operational Activity Action*.

> 🕮 *An Operational Activity Action is a distinct, identifiable phase or step within an Operational Activity that has a unique entrance criteria, exit criteria, and identifiable participating Operational Performer or Operational Domain.*

Using Operational Activities

> 🕮 *An Operational Activity is an end-to-end collection of Activity Actions that creates an outcome for a customer, who may be the ultimate customer or an internal end-user of the Operational Activity.*

Accessing Operational Activities

To access the list of *Operational Activities*:

1. From the **Operational** navigation menu, select **Processes**.
2. Click the **Owned Value Streams** tab
   The list of Operational Activities appears.

Creating an Operational Activity

To create a *Operational Activity*:

1. From the **Operational** navigation menu, select **Processes**.
2. Click the **Owned Value Streams** tab
3. Click **New**.
   The Operational Activity is created and added to the list of Operational Activities.
4. Specify the name of the new *Operational Activity*,

Properties of an Operational Activity

The **Characteristics** properties page of the Operational Activity provides access to:

- its **Owner**, by default on creation of the Operational Activity, the current enterprise.
- its **Name**,
- the text of its **Description**.

You can describe an Operational Activity as follows:

- The **Value Stream Fulfillments** page provides access to:
  - the **Fulfilled Value Stream**
  - the **Fulfilling Process**, see Representing the Operational Activity Implementation
- the **Diagram** page is used to create the Operational Activity diagrams:
  - Functional Process Tree Diagram,
  - The Functional Process diagram,
  - The Value Stream Capability Diagram.

Functional Process Tree Diagram

The **Functional Process Tree diagram** describes the tree structure of the current Operational Activity. Such diagram describes the current Operational Activity and its sub-Operational Activities.

The Functional Process diagram

The Functional Process diagram shows the sequence of the value creation steps performed, the events that occur and the conditions under which they are sequenced.

It also makes it possible to assign the participants who carry out these value-creation steps to the business skills needed to implement them.

This representation of a Operational Activity helps to answer the following question: "What are the skills needed to implement the Operational Activity?".

The **Functional Process Diagram** describes the components of the Operational Performer and the connections between components. It thus describes:

- the *Operational Activity*,

  📖 *An Operational Activity is an end-to-end collection of Activity Actions that creates an outcome for a customer, who may be the ultimate customer or an internal end-user of the Operational Activity.*

- the *Operational Activity Action*,

  📖 *An Operational Activity Action is a distinct, identifiable phase or step within an Operational Activity that has a unique entrance criteria, exit criteria, and identifiable participating Operational Performer or Operational Domain.*

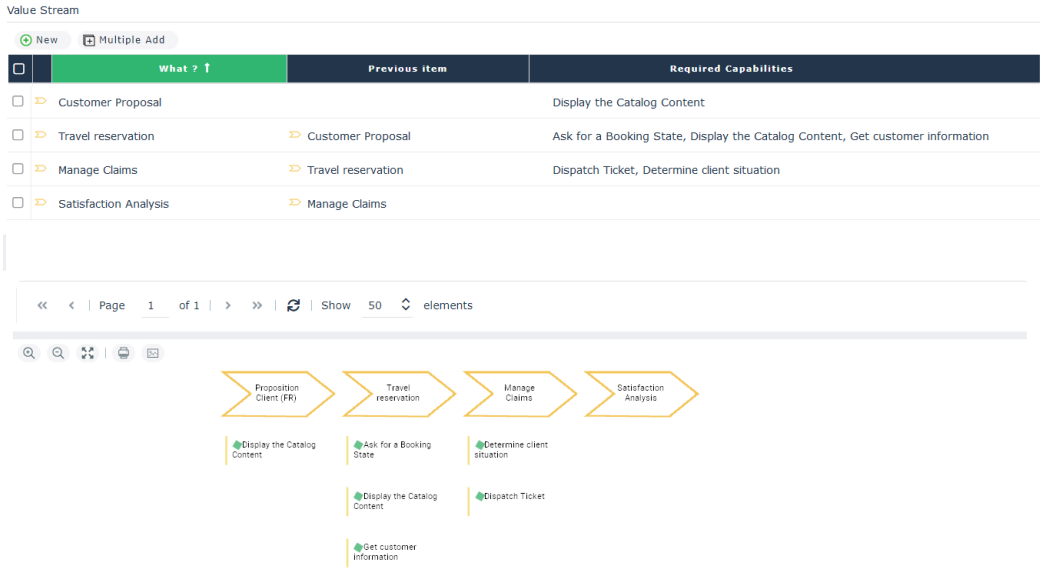
- Communications based on: *Sequence Flows* and *Message Flows with Content*.

  📖 *A Message Flow represents circulation of information within a service interface. A message flow transports its content.*

  ☛ *For more details on flows, see Describing Service and Request Points.*

The Value Stream Capability Diagram

This diagram makes it easy to initiate a Operational Activity diagram by creating the Operational Activity steps and the links they have with the different Capabilities.



Representing the Operational Activity Implementation

You can represent the fulfillment of a Operational Activity by a Personnel Function.

To access the list of *Personnel Functions* from the **Operational** navigation menu:

1. Click the **Processes** menu.
2. Open the **Characteristics** property page of the Personnel Function that interests you.
3. In the **Fulfillments** section, click the **New** button.
4. In the Add dialog box, select **Operational Activity Fulfillments** and select the concerned Operational Activity.

OPERATIONAL - STATES

The **States** aspect of the **Operational** view addresses the evolution of *Operational Performer* objects over the time using *State Machine* objects represented by a *State Machine diagram*.

> 🕮 *An Operational Performer is a skill or grouping of skills of interest for the enterprise.*
>
> 🕮 *A state machine is a set of states and transitions governing the state changes that can match any time-dependent object.*

To create a *State Machine* from the **Operational** navigation pane.

1. Select **States**.
   The list of State Machines connected to *Operational Performer* objects is displayed.
2. Click the **New** button.
   A creation wizard opens.
3. Enter the **Name** of the new State Machine.
4. Select (or create) a **Operational Performer** in the **Subject of State Machine** box.
5. Click **OK**.
   The state machine is created.

For more details on the use of *State Machines* with **HOPEX UAF**, see Strategic - States.

OPERATIONAL - SEQUENCES

The **Sequences** aspect of the **Operational** view helps UAF Architect to manage the scenario of operational flows avaible for *Operational Domains, Operational Performers* and *Operational Architectures*.

> 🕮 *A Operational Domain is a grouping of Operational Performers and their associated Operational Activities on the conjunction of two main criteria. 1. Their need in accomplishing one or more Capabilities. 2. The common skills and functionalities required to accomplish these Capabilities.*
>
> 🕮 *An Operational Performer is a skill or grouping of skills of interest for the enterprise.*
>
> 🕮 *A Operational Architecture represents the relationships of a Operational Domain with its partners.*

Scenario of Operational Flow Diagram

The **Scenario of Operational Flows Diagram** describes scenario of flow of:

- an *Operational Performers*, see Describing Operational Performers,
- or an *Operational Domain,* see Describing a Operational Domain.

This diagram contains the agents necessary for the scenario (Operational Performer or Operational Domain) and exchanged business flows. It thus describes:

- the *Operational Domains*,

  > 🕮 *A Operational Domain is a grouping of Operational Performers and their associated Operational Activities on the conjunction of two main criteria. 1. Their need in accomplishing one or more Capabilities. 2. The common skills and functionalities required to accomplish these Capabilities.*
  >
  > ☛ *For more information on the use of a Operational Domain, see Describing a Operational Domain.*

- the *Operational Performers*.

  > 🕮 *An Operational Performer is a skill or grouping of skills of interest for the enterprise.*

- communications based on: *Flows* and *Operational Flows Channels*.

  > 🕮 *A flow represents the circulation of information between components or within a component. A flow can carry a content.*
  >
  > 🕮 *An Operational Flow Channel is used to graphically group a number of flows into a single flow.*
  >
  > ☛ *For more details, see Creating an operational flow channel.*

Creating an operational flow channel

> 🕮 *An Operational Flow Channel is used to graphically group a number of flows into a single flow.*

To create an operational flow channel, you must first create the channel and then link the flows that it groups.

To create an *operational flow channel*:

1. In the objects toolbar of the scenario of operational flow diagram, click **Operational Flow Channel**.
2. Click the first object in communication and, holding the mouse button pressed, draw a link to the other object.
   The operational flow channel appears in the diagram.

To connect the flows to the *operational flow channel*:

1. Open the **Characteristics** properties page of the operational flow channel.
2. In the **Grouped Flow** section, click **Connect**.
   A selection dialog box opens and presents the list of the flows of the scenario.
3. Select the flows that you want to group and click **OK**.
   The content of the selected flows is displayed in the **Grouped Flow** list.
4. Click the **Refresh Channels** button.
   The operational flows grouped in the channel disappears and the corresponding content is displayed around the channel.

Scenario of Operational Environment Flows Diagram

The **Scenario of Operational Environment Flows Diagram** describes scenario of flow of an *Operational Architecture*, see Managing an Operational Architecture,

This diagram contains the agents necessary for the scenario (Operational Performer or Operational Domain) and exchanged business flows. It thus describes:

- the *Operational Domains*,

  📖 *A Operational Domain is a grouping of Operational Performers and their associated Operational Activities on the conjunction of two main criteria. 1. Their need in accomplishing one or more Capabilities. 2. The common skills and functionalities required to accomplish these Capabilities.*

  ☛ *For more information on the use of a Operational Domain, see Describing a Operational Domain.*

- the *Operational Performers*.

  📖 *An Operational Performer is a skill or grouping of skills of interest for the enterprise.*

- the *Operational Partners*.

  📖 *An Operational Partner is a conceptual stakeholder type that interacts with the enterprise in the context of the enterprise Operational Architecture. Examples : 'Private Sector Customer', 'Regulation Authority', 'Supplier'.*

- communications based on: *Flows* and *Operational Flows Channels*.

  📖 *A flow represents the circulation of information between components or within a component. A flow can carry a content.*

  📖 *An Operational Flow Channel is used to graphically group a number of flows into a single flow.*

  ☛ *For more details, see Creating an operational flow channel.*

OPERATIONAL - INFORMATION

The **Information** aspect of the **Operational** view identifies and defines strategic information elements and their relationships that are applicable to the architecture.

Business Dictionary

A business dictionary collects and structures a set of concepts that expresses the knowledge of a particular area.

The basic component of a business dictionary is the **Concept**.

> 🕮 *A concept is a statement expressing the essential nature of a being, an object, a word through its essential properties and characteristics or its specific qualities.*

The word that is associated with a **Concept** and which depends on language is a **Term**.

> 🕮 *A term is a word or group of words that is used for a specific meaning in a specific context.*

For information regarding Business Dictionaries in **HOPEX**, see "Business Dictionary" chapter in **HOPEX Data Governance** guide

To create a business dictionary in **HOPEX UAF**:

1. Click the **Operational** > **Hierarchy** navigation menu.
2. Click the icon of the **Business Dictionary** folder and click **New** > **Business Dictionary**.
3. Indicate:
   - the name of the Business Dictionary
   - the owner (optional)
   - a description (optional)
4. Click **OK**.

From the **Hierarchy** view of the business dictionaries you can create concepts and terms, as well as concept domains.

Concept

> 🕮 *A concept is a statement expressing the essential nature of a being, an object, a word through its essential properties and characteristics or its specific qualities.*

The concept is the basic element of a **Business Dictionary**.

A concept is associated with one or more terms that designate the concept in a given language.

For information regarding Concepts in **HOPEX**, see "Concept" chapter in **HOPEX Data Governance** guide

Accessing the list of Concepts

To access concepts in **HOPEX UAF**:

1. From the navigation menu, select **Operational > Information**.
2. Click the **Concept** tab.

Creating Concepts

To create a *concept*:

1. From the navigation menu, select **Operational > Information**.
2. Click the **Concept** tab.
3. Click the **New** button associated with the concept list or the concept folder.
4. Enter the **Name** of the concept and the **Owner Business Dictionary**.
5. The **Existing Terms** section lists terms with the same name as the new concept. You can choose to use an already existing term, or create a new term.

   A term is a word or group of words that is used for a specific meaning in a specific context.

   *If a term has already been created with the same name as the new concept, this term is automatically connected and appears automatically in the **Term** section.*
6. In the **Description** field, enter the text of the concept definition.
7. Click **Next** to associate an image with the concept or **OK** to finish.
   The name of the new concept appears in the tree. It also appears in the tree structure of the holding business dictionary. A new term with the same name as the concept is also created.

Creating a Concept Diagram

A concept diagram is a graphical representation of the concepts used in the context of a concept domain, as well as the links that exist between these concepts.

For information regarding Concept Diagram in **HOPEX**, see "Building Concept Diagrams" chapter in **HOPEX Data Governance** guide.

Concept Domain

A concept is a statement expressing the essential nature of a being, an object, a word through its essential properties and characteristics or its specific qualities.

A concept domain is a sub-set of elements of a subject area that reduces the scope of a fields.

For information regarding Concept Domains in **HOPEX**, see "Concept Domain" chapter in **HOPEX Data Governance** guide.

Accessing the list of Concept Domains

To access concept domains in **HOPEX UAF**:

1. From the navigation menu, select **Operational > Information**.
2. Click the **Concept Domain** tab.

Creating a Concept Domain

To create a concept domain:

1. From the navigation menu, select **Operational > Information**.
2. Click the **Concept Domains** tab.
3. Click **New**.
   The concept domain is created.

Creating a Concept Domain Diagram

With **HOPEX UAF**, a Concept Domain can be described by the following diagrams:

- the **Diagram** page is used to create the Operational Activity diagrams:
  - Concept Domain Diagram,
  - Concept Domain Structure Diagram.

Concept Domain Diagram

A concept domain diagram is a graphical representation of the concepts used in the context of a concept domain, as well as the links that exist between these concepts.

A concept domain can be described by a number of concept diagrams.

For information regarding Concept Diagram in HOPEX, see "Concept structure Diagram" chapter in HOPEX Data Governance guide

Concept Domain Structure Diagram

A structure diagram defines the sub-domains of the concept domains and their relationships.

OPERATIONAL - CONSTRAINTS

The **Constraints** aspect of the **Operational** view is dedicated to the constraints imposed to *Operational domains* and *Operational performers*.

> 🕮 *A Operational Domain is a grouping of Operational Performers and their associated Operational Activities on the conjunction of two main criteria. 1. Their need in accomplishing one or more Capabilities. 2. The common skills and functionalities required to accomplish these Capabilities.*
>
> 🕮 *An Operational Performer is a skill or grouping of skills of interest for the enterprise.*

Two types of **Constraints** may be imposed to *Operational domains* and *Operational performers*:

- Mesurable properties, specified in the capability **Measurable Property** and **Qualifying Values** of the capability.

  > ☛ *For more details on Measurable Properties, see Describing a Measurable Property.*

- Directing Regulations, specified in the capability **Regulations** properties.
  - *For more details on Directing Regulation structure, see Security - Taxonomy.*

Two types of contraints reports are available:

- The Measure Reports,
- The Politicy Reports.

OPERATIONAL - TRACEABILITY

The **Traceability** aspect of the **Operational** view provides access to the **Business Capabilities to Operational Performers** matrixes and reports displaying the fulfillments of Capabilities by Operational Performers.

To associate an Operational Performer to a Capability, you must create a capability fulfillment. For more details, see Capability Fulfillment.

- Capabilities are presented in rows

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

- Operational Performers are presented in columns

  📖 *An Operational Performer is a skill or grouping of skills of interest for the enterprise.*

  📖 *A Fulfillment describes the relationship between a logical entity and a physical entity that implements it. The physical entity gives the list of logical entities that fulfill it.*

For more details on Traceability matrixes and reports, see Using Traceability Aspect.

For more details on Exhibited Capabilities, see Managing Exhibited Capabilities.

SERVICES VIEWPOINT

The **Services** viewpoint is dedicated to Services management process. It shows the specifications of services required/provided depending on the service levels. Those **Services** support Operational Activities or are expected to exhibit a Capability.

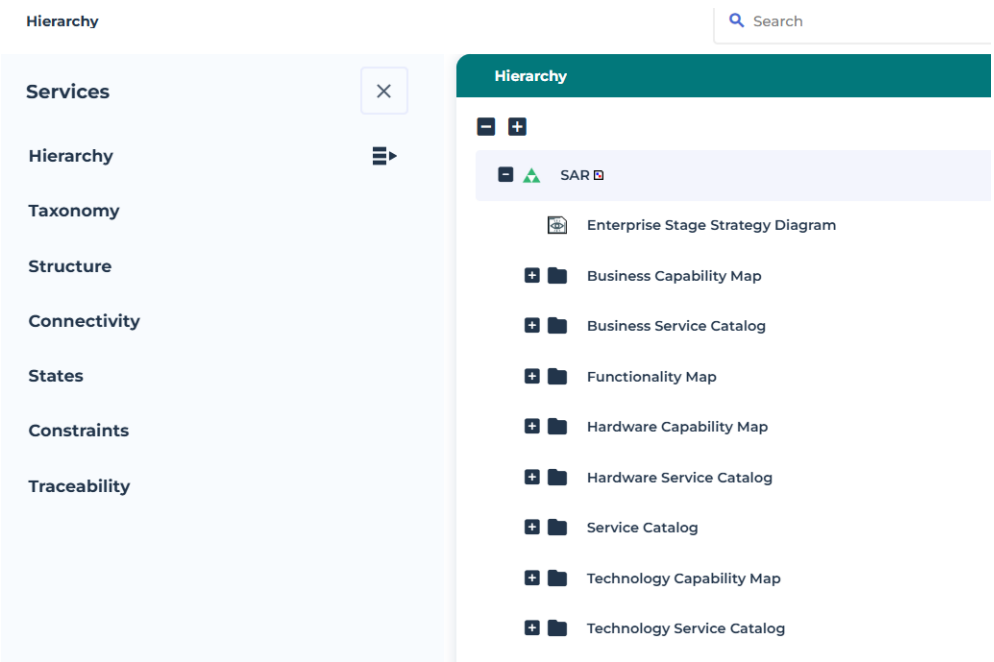
> 🕮 *A service is a mechanism to enable access to one or more capabilities, where the access is provided using a prescribed service interface and is exercised consistent with service constraints and policies.*

This chapter comprises the following sections:

- ✓ Services - Hierarchy;
- ✓ Services - Taxonomy;
- ✓ Services - Structure;
- ✓ Services - Connectivity;
- ✓ Services - States;
- ✓ Services - Constraints;
- ✓ Services - Traceability.

SERVICES - HIERARCHY

The **Hierachy** aspect of the **Services** view helps UAF Architect to manage the links between the services expected by the operational activities and the existing ones.



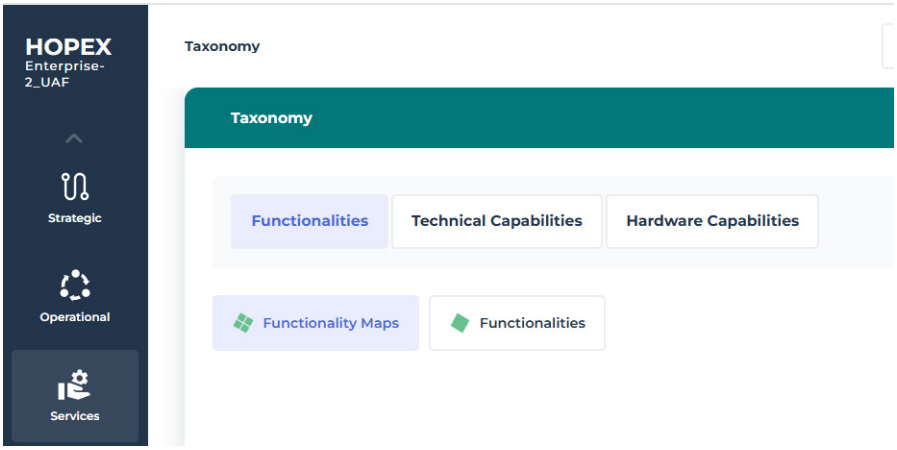
To get more information about the use of folders in **HOPEX** trees, see **HOPEX**, "Handling Trees" chapter of the **HOPEX Common Features** guide.

The hierarchy navigation enables to navigate on the main concepts of the view:

- Business Capability Map, see Describing a Capability Map,
- Business Service Catalog, see Using service catalogs,
- Functionality Map, see Describing Functionalities,
- Hardware Capability Map, see Describing Hardware Capabilities,
- Hardware Service Catalog, see Using service catalogs,
- Service Catalog, see Using service catalogs
- Technology Capability Map, see Describing Technical Capabilities,
- Technology Service Catalog, see Using service catalogs.

SERVICES - TAXONOMY

The **Taxonomy** aspect of the **Services** view helps **UAF Architect** to describe and classify all the expected functionalities.



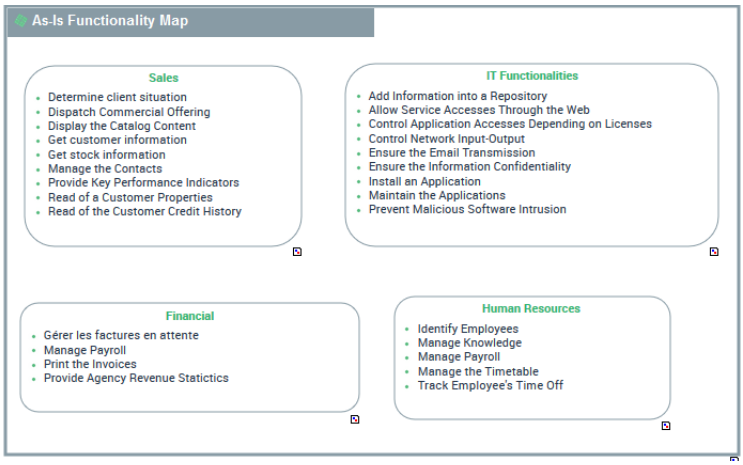
Describing Functionalities

📖 *A Functionality is a service required by an org-unit in order to perform its work. This functionality is generally necessary within an activity in order to execute a specific operation. If it is a software functionality, it can be provided by an application.*

Describing Functionality Maps

A *functionality map* describes all the functionalities the enterprise is able to cover for its internal needs or for meeting the needs of its clients.

> 🕮 *A Functionality Map is an assembly of functionalities and their dependencies that, together, defines the scope of a hardware or software architecture.*



Example of a functionality map

> ☛ *For more details on Functionality Maps management, see "Describing the Functionality Map" of* ***HOPEX IT Business Management*** *guide.*

Accessing Functionality Maps

To access the list of Capability maps:

1. From the **Services** navigation menu, select **Taxonomy**.
2. From the **Functionalities** tab, select **Functionality Map** sub tab.
   The list of functionality maps is displayed.

Creating a Functionality Map

To create a *Functionality Map*:

1. From the **Services** navigation menu, select **Taxonomy**.
2. From the **Functionalities** tab, select **Functionality Map** sub tab.
   The list of functionality maps appears.
3. Click **New**.
   A creation dialog box opens.
4. Specify the **Name** and click **OK**.
   The new functionality map appears in the list.

Creating a functionality map diagram with HOPEX UAF

A functionality map can be described by two diagram types:

- a functionality decomposition tree is a diagram that describes the tree structure of a functionality. Focusing on a particular functionality, this

type of diagram enables summary representation of functionality breakdown into sub-functionalities.

- A structure diagram used to represent the set of functionalities with their dependencies that, jointly, define the scope of a hardware or software architecture corresponding to the Functionality Map.

To create a functional map diagram:

1. Right-click the functionality map that interests you and select **Create Diagram**.
2. Select the diagram type.
   The diagram opens in the edit area. The frame of the functionality map described appears in the diagram.

Creating a functionality component in a functionality map diagram

The components represented in a functionality map are **Functionality components**.

> 🕮 *A Functionality is a service required by an org-unit in order to perform its work. This functionality is generally necessary within an activity in order to execute a specific operation. If it is a software functionality, it can be provided by an application.*

To add a functionality component in the functionality map diagram:

1. In the diagram insert toolbar, click **Functionality Component**.
2. Click the functionality map frame.
   The functionality component creation window opens.
3. Click, for example, **Reusing an Existing Functionality**.
4. Click **Display Scope** to access the list of functionalities linked to the enterprise.
5. Select the functionality that interests you.
6. Click **OK**.
   The functionality component appears in the diagram.

Defining Functionality dependencies

A dependency link between one functionality and another is used to specify the elements on which this dependency is based.

To create dependency links between two functionalities in a functionality map diagram:

1. In the insert toolbar, click **Functionality Dependency**.
2. Click the functionality component, and keeping the left mouse button pressed, move the cursor to the functionality component used.
3. Release the mouse button.
   The dependency link appears in the diagram.

Describing a Functionality

> 🕮 *A Functionality is a service required by an org-unit in order to perform its work. This functionality is generally necessary within an activity in order to execute a specific operation. If it is a software functionality, it can be provided by an application.*

The **Characteristics** properties page of a functionality provides access to:

- its **Owner**, by default during creation of the functionality, the current enterprise.
- its **Name**,
- the text of its **Description**.
- its **Desired capability effect**.

> ☛ *For more information on the desired capbility effects, see Creating a Functionality Diagram with HOPEX UAF.*

Creating a Functionality Diagram with HOPEX UAF

The **Functionality Decomposition Tree** is a diagram that describes the tree structure of a functionality. Focusing on a particular functionality, this type of diagram enables summary representation of functionality breakdown into sub-functionalities.

To create a functionality diagram:

1. Right-click the functionality that interests you and select **Create Diagram**.
2. Select **Functionality Diagram**.
   The diagram opens in the edit area. The frame of the functionality described appears in the diagram.

Describing Technical Capabilities

> 🕮 *A Technology Capability is the ability to deliver a technology service which is required by a technology artifact in order to function. This technology capability is generally required within a technical process handle by the technology artifact.*

With **HOPEX UAF**, the use of technology capabilities and technology capability maps is identical to the use of the functionalities and functionality maps.

> ☛ *For more details on the functionality maps and functionalities, see Describing Functionalities.*

Describing a Technology Capability Map

> 🕮 *A Technology Capability Map is an assembly of functionalities and their dependencies that, together, defines the scope of a hardware or software architecture.*

Accessing the list of Technology Capability Maps

To access the list of *Technology Capability Maps*:

1. From the **Services** navigation menu, select **Taxonomy**.
2. From the **Technical Capabilities** tab, select **Technical Capability Map** sub tab.
   The list of technology capability maps is displayed.

Creating a Functionality Map

To create a *Technology Capability Map*:

1. From the **Services** navigation menu, select **Taxonomy**.
2. From the **Technical Capabilities** tab, select **Technical Capability Map**
   The list of technology capability maps appears.
3. Click **New**.
   A creation dialog box opens.
4. Specify the **Name** and click **OK**.
   The new technology capability map appears in the list.

Describing a Technology Capability

A Technology Capability is the ability to deliver a technology service which is required by a technology artifact in order to function. This technology capability is generally required within a technical process handle by the technology artifact.

Accessing the list of Technology Capabilities

To access the list of *Technology Capability*:

1. From the **Services** navigation menu, select **Taxonomy**.
2. From the **Technical Capabilities** tab, select **Technical Capabilities** sub tab.
   The list of technology capability is displayed.

Creating a Technology Capability

To create a *Technology Capability* :

1. From the **Services** navigation menu, select **Taxonomy**.
2. From the **Technical Capabilities** tab, select **Technical Capability** The list of technology capabilities appears.
3. Click **New**.
   A creation dialog box opens.
4. Specify the **Name** and click **OK**.
   The new technology capability appears in the list.

Describing Hardware Capabilities

A Hardware Capability is the ability to deliver a physical outcome which is required by an organizational resource in order to perform its work. This hardware capability is generally necessary within a process in order to execute a specific operation.

With **HOPEX UAF**, the use of hardware capabilities and hardware capability maps is identical to the use of the functionalities and functionality maps.

For more details on the functionality maps and functionalities, see Describing Functionalities.

Describing a Hardware Capability Map

A Hardware Capability Map is an assembly of hardware capabilities and their dependencies that, together, defines the scope of a hardware or software architecture.

Accessing the list of Hardware Capability Maps

To access the list of *Hardware Capability Maps*:

1. From the **Services** navigation menu, select **Taxonomy**.
2. From the **Hardware Capabilities** tab, select **Hardware Capability Map** sub tab.
   The list of hardware capability maps is displayed.

Creating a Hardware Capability Map

To create a *Hardware Capability Map*:

1. From the **Services** navigation menu, select **Taxonomy**.
2. From the **Technical Capabilities** tab, select **Hardware Capability Map** The list of hardware capability maps appears.
3. Click **New**.
   A creation dialog box opens.
4. Specify the **Name** and click **OK**.
   The new hardware capability map appears in the list.

Describing a hardware capability

A Hardware Capability is the ability to deliver a physical outcome which is required by an organizational resource in order to perform its work. This hardware capability is generally necessary within a process in order to execute a specific operation.

Accessing the list of Hardware Capabilities

To access the list of *Hardware Capability*:

1. From the **Services** navigation menu, select **Taxonomy**.
2. From the **Technical Capabilities** tab, select **Hardware Capabilities** sub tab.
   The list of hardware capability is displayed.

Creating a Hardware Capability

To create a *Hardware Capability*:

1. From the **Services** navigation menu, select **Taxonomy**.
2. From the **Technical Capabilities** tab, select **Hardware Capability** The list of hardware capabilities appears.
3. Click **New**.
   A creation dialog box opens.
4. Specify the **Name** and click **OK**.
   The new hardware capability appears in the list.

SERVICES - STRUCTURE

The **Structure** aspect of the **Services** view helps **UAF Architect** to describe and classify all the expected functionalities into *Service Catalogs*.

A service catalog describes the list of functionalities covered by a solution as well as the technical or functional elements that implement these functionalities.

> 🕮 *A service catalog contains a list of key service offers for which solutions are recommended.*

List of service catalogs available in HOPEX UAF

HOPEX UAF offers the following service catalogs:

- *business service catalogs*,

  > 🕮 *A business service catalog provides a centralized information source for the business services offered by the service provider organization. It contains a customer-oriented view of the services associated to business capabilities, how they are supposed to be used, the processes that they support as well as the expected service quality level. The business service catalog presents the list of functionalities mentioned as well as implementation recommendations.*

- *services catalogs*,

  > 🕮 *A service catalog contains a list of key service offers for which solutions are recommended.*

- *Technology service catalogs*,

  > 🕮 *A technology service catalog provides a centralized information source for the technology services offered by the service provider organization. It contains a customer-facing view of the technology services in use, how they are intended to be used, the process they enable, and the levels and quality of service the customer can expect from each service. The technology service catalog provides the list of reference technology capabilities and their recommended implementation.*

- *hardware service catalogs*.

  > 🕮 *A hardware service catalog provides a centralized information source for the hardware services offered by the service provider organization. It contains a customer-oriented view of the hardware used, how they are supposed to be used, the processes that they support as well as the expected service quality level. The hardware service catalog presents the list of hardware functionalities mentioned as well as implementation recommendations.*
  >
  > ☛ *For more details on the use of service catalogs, see Using service catalogs.*

Using service catalogs

Implementation of a service catalog

In **HOPEX UAF**, a service catalog is made up of service catalog item. For example, a *hardware services catalog* is made up of several *hardware service catalog items*.

> 🕮 *A hardware service catalog provides a centralized information source for the hardware services offered by the service provider organization. It contains a customer-oriented view of the hardware used, how they are supposed to be used, the processes that they support as well as the expected service quality level. The hardware service catalog presents the list of hardware functionalities mentioned as well as implementation recommendations.*
>
> 🕮 *A hardware service catalog item defines which hardware functionality is in the catalog and which hardware artifacts provide the hardware functionality.*

A *service catalog item* is connected to one or more *functionalities*.

> 🕮 *A Functionality is a service required by an org-unit in order to perform its work. This functionality is generally necessary within an activity in order to execute a specific operation. If it is a software functionality, it can be provided by an application.*

With **HOPEX UAF**, the implementation of a functionality is represented by an *Implementation*.

> 🕮 *An implementation describes the relationship between a logical entity and a physical entity that implements it. The physical entity gives the list of logical entities that it implements.*

The table below draws up the summary of objects that implement the service catalogs according to their category.

Type of service catalog	Type of functionality	Types of object that deliver the service
Business	Capabilities	Business capability
Service	Functionalities	Org-Unit
Technical	Technology Capabilities	Technologies
hardware	Hardware capabilities	Hardware and IoT Device.

Populating a service catalog

The management principle of a service catalog is identical for all types of service catalogs. The types of service catalogs offered are:

This chapter is based on the example of a *business service catalog*.

Creating a business service catalog

To create a *Business service catalog*:

1. From the **Services** navigation menu, select **Structure**
2. Select the tab corresponding to the type of service catalog that interests you: **Business Service Catalog**, for example.
3. Click the arrow at the right of the **All Business Service Catalogs** field and select **New**.
   A Business Service Catalog is created, the corresponding tree appears below.
4. Modify the **Name** of the new catalog.

> ☛ *Likewise, you can create a technical service catalog or a hardware service catalog.*

Adding a business service catalog item

The **Characteristics** property page of a business service catalog provides access to:

- its **Owner**, by default, during creation of the logical application system, the current library.
- its **Name**,
- the list of business service catalog items owned.

To add a *Business service catalog item*:

1. From the **Services** navigation menu, select **Structure**
2. Select the tab corresponding to the type of service catalog that interests you: **Business Service Catalog**, for example.
3. Using the arrow at the right of the **All Business Service Catalogs** field, select the Business Service Catalog that interests you.
   The tree of *Business service catalog items* appears below.
4. In the **Action** column of the Business Service Catalog line, click **Add** button.
   An **Add** window opens.
5. Create a Capability or reuse existing ones and click **Add**.
   The business catalog item pops up in the list with the associated capability.

> ☛ *The business catalog item appears also in the **Characteristics** property page of a business service catalog.*

Specifying the implementation of a service catalog item

To describe how the functionality (or capability) associated to a service catalog item is fulfilled, you must define an *Implementation*.

> 🕮 *An implementation describes the relationship between a logical entity and a physical entity that implements it. The physical entity gives the list of logical entities that it implements.*

To describe the fulfillment of a service catalog item:

1. Click the **Plus** button of your Business Service Catalog to display the tree of *Business service catalog items*.

2. In the **Action** column of the Business Service Catalog item line, click **Add** button.
   An **Add** window opens.
3. Select the type of the objects that implement the Business Service Catalog Item: **Application** for example.
4. Create an object or reuse existing ones and click **Add**.
   The selected elements pops up in the list with the associated type.

> ☛ *The list of Business Service Catalog Item Fulfilments appears also in the **Characteristics** property page of a business service catalog item in the **Business Service Catalog Item** section.*
>
> ☛ *For more details, see Using fulfillment mechanisms.*

Service Catalog Description

To access this report:

❱ Open the **Reporting** property of the Service catalog that interests you.

This report consists of a table displaying:

- service catalogs and service catalog items together with realizers and service points

  > 🕮 *A service point is a point from which a system receives a request from another system and provides the requested service.*

- interactions with their context.

Service Catalog	Service Catalog Item	Agent Realizer	Service Point	Interact with	Context of Interaction
SAR Business Service Catalog					
SAR Technical Service Catalog					
SAR Hardware Service Catalog	Lifejacket Functionalities				
Video Information Service Catalog	Analysis Technology Standards Changes				
Video Technical Service Catalog	Video Upload Tool				
DODAF Business Transformation 1	Video User Authentication	Medical Assistance			
Video Hardware Service Catalog	Video Luminosity Measurement	Luminosity Measurement Tool			
Video Information Service Catalog	Video Standard Compliance Check	Breakdown Analysis Recorder			
		Video Camera			
		SAR Drone			
		Video Drone			
		Video Smartphone			
Video Hardware Service Catalog	Hardware Service Catalog Item	Tracking System		ESM System	Lifeboat
Video Information Service Catalog	Video Standard Compliance Check	Video Classification Application		Video Check Application	Video Management Application Environment

SERVICES - CONNECTIVITY

The **Connectivity** aspect of the **Services** view helps **UAF Architects** in the specification of service interfaces, e.g., provided and required service methods, signal receptions, and/or flow properties, to ensure compatibility and reusability of services.

Using Interfaces from Services view

> 🕮 *An Interface is a template of a contract between entities (organizational, IT …). The contract is described by available operations which can be triggered trough messages exchanged by roles (vendor, buyer..).*

To access the list of *Interfaces*:

1. From the **Services** navigation menu, select **Connectivity**.
2. Click the **Interfaces** tab.
   The list of the Interfaces is displayed.

For more details on the mangement of *Interfaces* with **HOPEX UAF**, see Describing an Interface.

Using Signals from Services view

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

To access the list of *Signals*:

1. From the **Services** navigation menu, select **Connectivity**.
2. Click the **Signals** tab.
   The list of the Signals is displayed.

For more details on the mangement of *Signals* with **HOPEX UAF**, see Using Signals.

Using Service Interactions

To access the list of **Service Interactions**:

1. From the **Services** navigation menu, select **Connectivity**.
2. Click the **Service Interactions** tab.
   The list of the Service Interactions is displayed.

For more details on the creation of Service Interactions with **HOPEX UAF**, see Using Service Interactions.

Using Service Dependencies

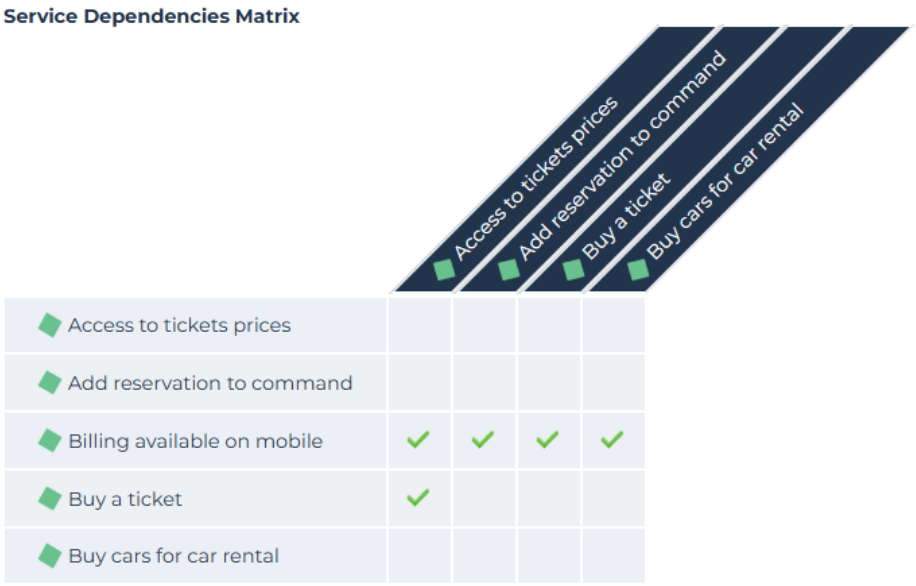
Services are represented by Functionalities. For more details on the dependencies between functionalities, see Defining Functionality dependencies.

To access the list of **Service Dependencies**:

1. From the **Services** navigation menu, select **Connectivity**.
2. Click the **Service Dependencies** tab.
   The list of **Service Dependencies Matrix** appears.

To create a **Service Dependencies Matrix**:

1. From the **Services** navigation menu, select **Connectivity**.
2. Click the **Service Dependencies** tab.
   The list of **Service Dependencies Matrix** appears.
3. Click **New**.
   A creation wizard opens.
4. Enter the name of your matrix.
5. Click **Connect** to select the type of Service Catalog that interests you and select the objects.
6. Click **Connect** and **OK**.
   The corresponding **Service Dependencies Matrix** appears.



SERVICES - STATES

The **States** aspect of the **Services** view addresses the evolution of objects dedicated to services over the time using *State Machine* objects represented by a *State Machine diagram*.

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

In the **Services** view, the state machines can be connected to the follwing object types:

- Functionnality,

  > *A Functionality is a service required by an org-unit in order to perform its work. This functionality is generally necessary within an activity in order to execute a specific operation. If it is a software functionality, it can be provided by an application.*

- Hardware Capability,

  > *A Hardware Capability is the ability to deliver a physical outcome which is required by an organizational resource in order to perform its work. This hardware capability is generally necessary within a process in order to execute a specific operation.*

- Technology Capability.

  > *A Technology Capability is the ability to deliver a technology service which is required by a technology artifact in order to function. This technology capability is generally required within a technical process handle by the technology artifact.*

To create a *State Machine* from the **Operational** navigation pane.

1. Select **States**.
   The list of State Machines connected to ***Service*** objects is displayed.
2. Click the **New** button.
   A creation wizard opens.
3. Enter the **Name** of the new State Machine.
4. Select an **Object Type**.
5. Select (or create) the object in the **Subject of State Machine** box.
6. Click **OK**.
   The state machine is created.

For more details on the use of *State Machines* with **HOPEX UAF**, see Strategic - States.

SERVICES - CONSTRAINTS

The **Constraints** aspect of the **Services** view is dedicated to the constraints imposed to *Functionalities*. *Software Capabilities* and *Hardware Capabilities*.

> 🕮 *A Functionality is a service required by an org-unit in order to perform its work. This functionality is generally necessary within an activity in order to execute a specific operation. If it is a software functionality, it can be provided by an application.*
>
> 🕮 *A Hardware Capability is the ability to deliver a physical outcome which is required by an organizational resource in order to perform its work. This hardware capability is generally necessary within a process in order to execute a specific operation.*
>
> 🕮 *A Technology Capability is the ability to deliver a technology service which is required by a technology artifact in order to function. This technology capability is generally required within a technical process handle by the technology artifact.*

Two types of **Constraints** may be imposed to *Functionalities*. *Software Capabilities* and *Hardware Capabilities*:

- Mesurable properties, specified in the capability **Measurable Property** and **Qualifying Values** of the capability.

  > ☛ *For more details on Measurable Properties, see Describing a Measurable Property.*

- Directing Regulations, specified in the capability **Regulations** properties.
  - *For more details on Directing Regulation structure, see Security - Taxonomy.*

Twotypes of contraints reports are available:

- The Measure Reports,
- The Politicy Reports.

SERVICES - TRACEABILITY

The **Traceability** aspect of the **Services** view provides an **Operational Activity to Services Traceability Matrix Report**: a matrix between service catalog items and the performed operational activities.

To create an **Operational Activity to Services Traceability Matrix Report**:

1. From the **Services** navigation menu, select **Traceability**.
2. Click **New**.
   A creation wizard opens.
3. Enter the name of your report and click **Next**.
   A Parameters window opens.
4. Click **Connect** to select the type of Service Catalog that interests you and select the objects.
5. Click **Next** to select the report chapters.
6. Click **OK**.
   The report opens. The selected chapters are represented by tabs.

PERSONNEL VIEWPOINT

The **Personnel** viewpoint defines and explores organizational resource types. Shows the taxonomy of types of organizational resources as well as connections, interaction and growth over time. It aims to clarify the role of Human Factors (HF) when creating architectures in order to facilitate both Human Factors Integration (HFI) and systems engineering (SE).

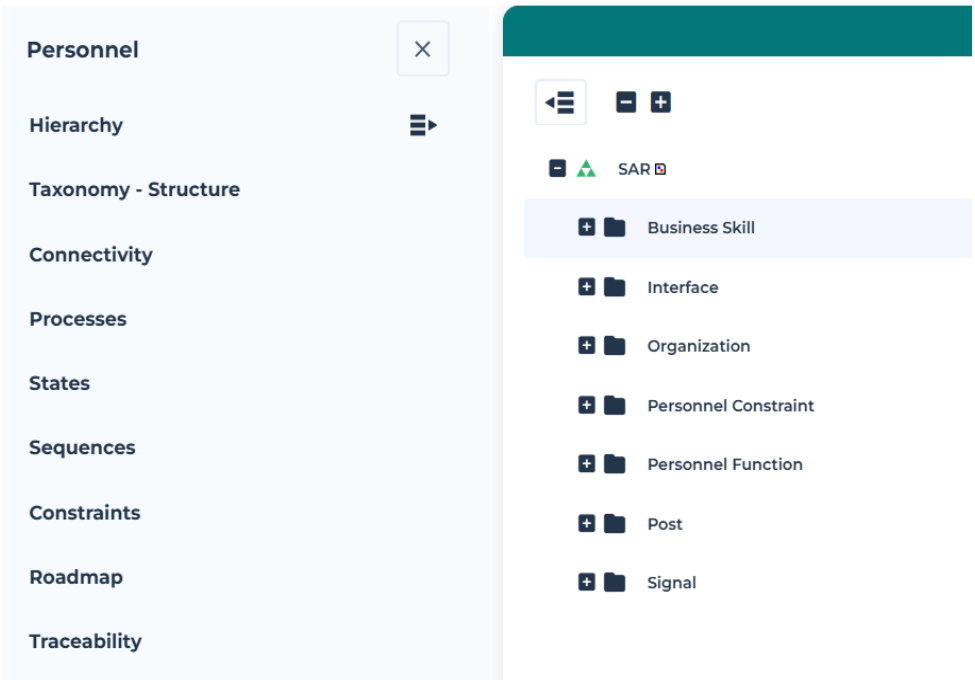
It defines and explores organizational resource types. Shows the taxonomy of types of organizational resources as well as connections, interaction and growth over time.

This chapter comprises the following sections:

- ✓ Personnel - Hierarchy;
- ✓ Personnel - Taxonomy - Structure;
- ✓ Personnel - Connectivity;
- ✓ Personnel - Process;
- ✓ Personnel - States;
- ✓ Personnel - Sequences;
- ✓ Personnel - Constraints;
- ✓ Personnel - Roadmap;
- ✓ Personnel - Traceability.

PERSONNEL - HIERARCHY

The **Hierachy** aspect of the **Personnel** view helps UAF Architect to manage the description of personnal resources: their composition, dependencies and skills.



To get more information about the use of folders in **HOPEX** trees, see **HOPEX**, "Handling Trees" chapter of the **HOPEX Common Features** guide.

The hierarchy navigation enables to navigate on the main concepts of the view:

- The current Enterprise and its Enterprise Stages, see Creating a Transformation Stage,
- Business Skill,see Business Skills,
- Interface,see Using Interfaces form Personnel view,
- Organization,see Organizations,
- Personnel Constraint,see Personnel Constraints,
- Personnel Function,see Personnel - Process,
- Post,see Posts,
- Signal,see Using Signals form Personnel view.

PERSONNEL - TAXONOMY - STRUCTURE

The **Taxonomy - Structure** aspect of the **Personnel** view helps UAF Architect to describe the human organization required to run the Enterprise Personnel Functions.

Organizations

Organizations enable to display the possible relationships between organizational resources.

> 🕮 *An organization represents a person or a group of persons that intervenes in the enterprise Personnel Functions or information system. An organization can be internal or external to the enterprise. An internal organization is an organizational element of the enterprise structure such as a department, a service, or a workstation. An internal organization 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 organization is an organization that exchanges flows with the enterprise. Example: Customer, Supplier, Government Office.*

Accessing Organizations

To access Organizations:

1. From the **Personnel** navigation menu, select **Taxonomy - Structure**.
2. Click the **Organizations** tab.
   The list of Organizations is displayed. This list displays the organizations that are owned by the enterprise or imported from other containers.

Creating an Organization

To create an *Organization*:

1. From the **Personnel** navigation menu, select **Taxonomy - Structure**.
   The list of Organizations appears.
2. Click the **Organizations** tab.
3. Click **New**.
   The new Organization appears in the list.

Properties of an Organization

The **Characteristics** properties page of an organization provides access to:

- its **Owner**, the current enterprise by default,
- its **Name**,
- the text of its **Description**,
- **Internal / External**

> 🕮 *An internal org-unit is an organizational element of enterprise structure such as a management, department, or job function. It is defined at a level depending on the degree of detail to be provided on*

the organization (see org-unit type). Example: financial management, sales management, marketing department, account manager.

📖 *An external org-unit is an external entity that exchanges flows with the enterprise. Example: customer, supplier, government office.*

- **E-mail**

The **Diagram** properties page of an organization provides access to several diagram types:

- External Data Flows Diagram

  ☛ *See Scenario of Organizational Environment Flow Diagram*

- External Interactions Diagram
  This is diagram displays the services provided and used by the organization.
- Hierachy Tree diagram
  This is the equivalent of an organization chart. It displays the decomposition of an organization into several sub-organizations.
- Internal Architecture Diagram

  ☛ *See Creating an Organizational Structure Diagram*

- Internal Data Flows Diagram

  ☛ *See Creating an Organizational Internal Flow Diagram.*

Creating an Organizational Structure Diagram

You can insert the following objects in this diagram:

- Org-Unit component
- Position-Type component
- Service Interactions and endpoints

📖 *A Service Interaction represents an interaction for service purpose between entities in a specific context inside or outside a company. These entities can be enterprise org-units, applications, activities or Personnel Functions, as well as external org-units. The content of this interaction is described in an Interface.*

Posts

📖 *A Post is a type of status assigned to an individual or a group of individuals which defines a job in an organization or hierarchy.*

Accessing Posts

To access Posts:

1. From the **Personnel** navigation menu, select **Taxonomy - Structure**.
2. Click the **Posts** tab.
   The list of Posts is displayed.

Creating a Post

To create a *Post*:

1. From the **Personnel** navigation menu, select **Taxonomy - Structure**.
2. Click the **Posts** tab.
   The list of Posts appears.
3. Click **New**.
   The new Post appears in the list.

Properties of a Post

The **Characteristics** properties page of an organization provides access to:

- its **Owner**, the current enterprise by default,
- its **Name**,
- the text of its **Description**,
- **Internal / External**

PERSONNEL - CONNECTIVITY

The **Connectivity** aspect of the **Personnel** view helps UAF Architect in the specification of service interfaces and signals.

Using Signals form Personnel view

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

To access the list of *Signal*:

1. From the **Personnel** navigation menu, select **Connectivity**.
2. Click the **Signals** tab.
   The list of the Signals is displayed.

For more details on the mangement of *Signals* with **HOPEX UAF**, see Using Signals.

Using Interfaces form Personnel view

> 🕮 *An Interface is a template of a contract between entities (organizational, IT …). The contract is described by available operations which can be triggered trough messages exchanged by roles (vendor, buyer..).*

To access the list of *Interfaces*:

1. From the **Personnel** navigation menu, select **Connectivity**.
2. Click the **Interfaces** tab.
   The list of the Interfaces is displayed.

For more details on the mangement of *Interfaces* with **HOPEX UAF**, see Describing an Interface.

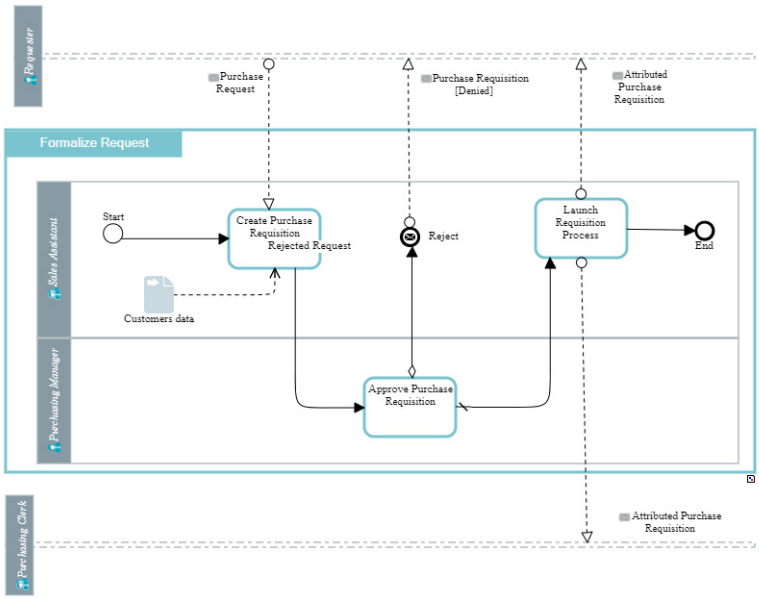
PERSONNEL - PROCESS

The **Process** aspect of the **Personnel** view is used to describe the Personnal Functions performed in the enterprise.

> *A Personnel Function is a set of operations performed by org-units within a company or organization, to produce a result. It is depicted as a sequence of operations, controlled by events and conditions. In the BPMN notation, the Personnel Function represents a sub-Personnel Function from the organizational point of view.*

Personnel Functions

In the example of the purchase request Personnel Function, the organization is represented by the following diagram.



The purchase request is received by a purchasing assistant, who enters the request and submits this for the approval of the purchasing manager.

If the request is rejected, the purchasing manager informs the requester.

If the request is approved, the assistant sends a completed request to buyers responsible for issuing the order, and sends a confirmation message to the requester.

This chapter explains how to use the main objects presented in this diagram.

- The framework containing the different components represents the *Personnel Function* described in the diagram. The name of the Personnel Function "Process Purchase Request" appears at the top left side of the framework.

  > 🕮 *A Personnel Function is a set of operations performed by org-units within a company or organization, to produce a result. It is depicted as a sequence of operations, controlled by events and conditions. In the BPMN notation, the Personnel Function represents a sub-Personnel Function from the organizational point of view.*

- The *participants* in execution of this process are *organizations*. They are represented in pools for greater clarity.

  > 🕮 *An organization represents a person or a group of persons that intervenes in the enterprise Personnel Functions or information system. An organization can be internal or external to the enterprise. An internal organization is an organizational element of the enterprise structure such as a department, a service, or a workstation. An internal organization 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 organization is an organization that exchanges flows with the enterprise. Example: Customer, Supplier, Government Office.*

- The different steps in this process are *Personnel Function Actions*. Organization of these steps is described by sequence flows.

  > 🕮 *A Personnel Function Action is an elementary step in process executed by an org-unit. It cannot be broken down. An operation can be industrial (manufacturing a component), logistical (receiving a delivery), or can involve information processing (entering an order).*

- *Message flows* enable representation of data or information circulating between a process and the exterior.

  > 🕮 *A Message Flow represents circulation of information within a service interface. A message flow transports its content.*

For information regarding Personnel Functions (Processes) in HOPEX, see "Processes" chapter in HOPEX Business Process Analysis guide

Accessing Personnel Functions

To access Personnel Functions:

1. From the **Personnel** navigation menu, select **Process**.
   The list of the Personnel Functions is displayed.

Creating a Personnel Function

To create a *Personnel Function*:

1. From the **Personnel** navigation menu, select **Process**.
   The list of the Personnel Functions appears.
2. Click **New**.
   The new Personnel Function appears in the list.

Personnel Function diagrams

You can describe a Personnel Function using the following diagrams:

- Personnel Function Tree Diagram,
- Personnel Function diagram.

Personnel Function Tree Diagram

The **Personnel Function Tree diagram** describes the tree structure of the current Personnel Function. Such diagram describes the current Personnel Function and its sub-Personnel Functions.

Personnel Function diagram

For information regarding Personnel Functions (Processes) in HOPEX, see "Processes" chapter in HOPEX Business Process Analysis guide

The components of a Process Function diagram are:

- *Personnel Functions*,

  A Personnel Function is a set of operations performed by org-units within a company or organization, to produce a result. It is depicted as a sequence of operations, controlled by events and conditions. In the BPMN notation, the Personnel Function represents a sub-Personnel Function from the organizational point of view.

- The *participants* represented in pools for greater clarity.
- *Personnel Function Actions*. Organization of these steps is described by sequence flows.

  A Personnel Function Action is an elementary step in process executed by an org-unit. It cannot be broken down. An operation can be industrial (manufacturing a component), logistical (receiving a delivery), or can involve information processing (entering an order).

- *Message flows* enable representation of data or information circulating between a process and the exterior.

  A Message Flow represents circulation of information within a service interface. A message flow transports its content.

PERSONNEL - STATES

The **States** aspect of the **Personnel** view addresses the evolution of objects dedicated to personnal over the time using *State Machine* objects represented by a *State Machine diagram*.

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

In the **Personnel** view, the state machines can be connected to the follwing object types:

- Organization,

  > 🕮 *An organization represents a person or a group of persons that intervenes in the enterprise Personnel Functions or information system. An organization can be internal or external to the enterprise. An internal organization is an organizational element of the enterprise structure such as a department, a service, or a workstation. An internal organization 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 organization is an organization that exchanges flows with the enterprise. Example: Customer, Supplier, Government Office.*

- Post,

  > 🕮 *A Post is a type of status assigned to an individual or a group of individuals which defines a job in an organization or hierarchy.*

To create a *State Machine* from the **Personnel** navigation pane.

1. Select **States**.
   The list of State Machines connected to **Personnel** objects is displayed.
2. Click the **New** button.
   A creation wizard opens.
3. Enter the **Name** of the new State Machine.
4. Select an **Object Type**.
5. Select (or create) the object in the **Subject of State Machine** box.
6. Click **OK**.
   The state machine is created.

For more details on the use of *State Machines* with **HOPEX UAF**, see Strategic - States.

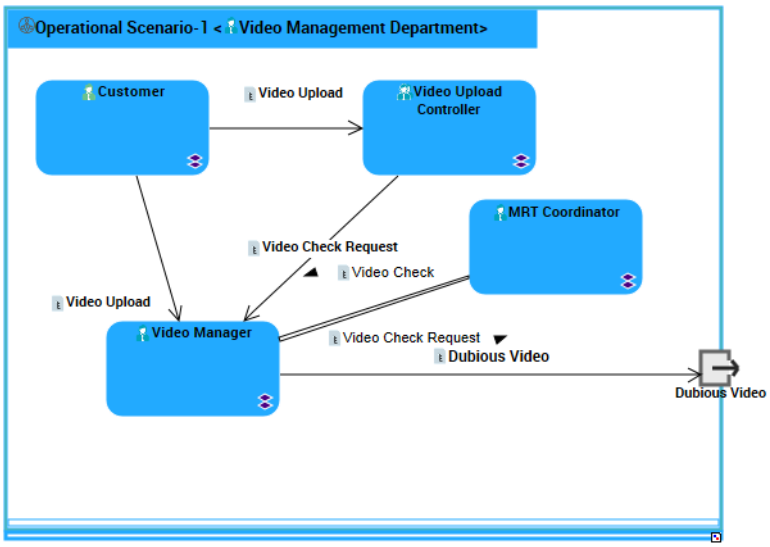
PERSONNEL - SEQUENCES

The **Sequences** aspect of the **Personnel** view helps UAF Architect to manage the scenario of organizational flows avaible for *Organizations.*

> *An organization represents a person or a group of persons that intervenes in the enterprise Personnel Functions or information system. An organization can be internal or external to the enterprise. An internal organization is an organizational element of the enterprise structure such as a department, a service, or a workstation. An internal organization 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 organization is an organization that exchanges flows with the enterprise. Example: Customer, Supplier, Government Office.*

Creating an Organizational Internal Flow Diagram

You can create an Organizational Internal Flow diagram for an organization to describe how participants of this organization (Organizations and Posts) exchange flows for this specific scenario.



Example of Operational Scenario Diagram

You can insert the following components in an Organizational Internal Flow Diagram:

- *Organizations*,

> *An organization represents a person or a group of persons that intervenes in the enterprise Personnel Functions or information system. An organization can be internal or external to the enterprise. An internal organization is an organizational element of the enterprise structure such as a department, a service, or a workstation. An internal*

organization 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 organization is an organization that exchanges flows with the enterprise. Example: Customer, Supplier, Government Office.

- *Posts*, see Posts

  📖 *A Post is a type of status assigned to an individual or a group of individuals which defines a job in an organization or hierarchy.*

Exchanges between components are represented by *Flows*, and *Personal Flow Channels.*

📖 *A flow represents the circulation of information between components or within a component. A flow can carry a content.*

📖 *An Personal Flow Channel is used to graphically group a number of flows into a single flow.*

Scenario of Organizational Environment Flow Diagram

The **Scenario of Organization Environment Flow Diagram** describes scenario of flow of an *Organization*, see Organizations.

This diagram contains the agents necessary for the scenario (Organizations or Posts) and exchanged flows. It thus describes:

- *Organizations*,

  📖 *An organization represents a person or a group of persons that intervenes in the enterprise Personnel Functions or information system. An organization can be internal or external to the enterprise. An internal organization is an organizational element of the enterprise structure such as a department, a service, or a workstation. An internal organization 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 organization is an organization that exchanges flows with the enterprise. Example: Customer, Supplier, Government Office.*

- *Posts*, see Posts

  📖 *A Post is a type of status assigned to an individual or a group of individuals which defines a job in an organization or hierarchy.*

- the *Organizational Partners*.

  📖 *An Organizational Partner is a conceptual stakeholder type that interacts with the enterprise in the context of the enterprise Organization.*

- communications based on: *Flows* and *Personal Flow Channels*.

  📖 *A flow represents the circulation of information between components or within a component. A flow can carry a content.*

  📖 *An Personal Flow Channel is used to graphically group a number of flows into a single flow.*

PERSONNEL - CONSTRAINTS

The **Constraints** aspect of the **Personnel** view is dedicated to the constraints imposed to *Organizations* and *Posts*.

> 🕮 *An organization represents a person or a group of persons that intervenes in the enterprise Personnel Functions or information system. An organization can be internal or external to the enterprise. An internal organization is an organizational element of the enterprise structure such as a department, a service, or a workstation. An internal organization 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 organization is an organization that exchanges flows with the enterprise. Example: Customer, Supplier, Government Office.*
>
> 🕮 *A Post is a type of status assigned to an individual or a group of individuals which defines a job in an organization or hierarchy.*

List of Personnel constraints and reports

Three types of **Constraints** may be imposed to capabilities:

- Personnel constraints, specified in the object **Constraints** page,

  ☛ *For more details on Personnel Constraints, see Personnel Constraints*

- Mesurable properties, specified in the **Measurable Property** and **Qualifying Values** property pages.

  ☛ *For more details on Measurable Properties, see Describing a Measurable Property.*

- Directing Regulations, specified in the objet **Regulations** properties.
  - *For more details on Directing Regulation structure, see Security - Taxonomy.*

Three types of contraints reports are available:

- The Constraint Reports,
- The Measure Reports,
- The Politicy Reports.

Personnel Constraints

> 🕮 *An Operational Constraint is a rule that is under business jurisdiction.*

To access Personnel Constraints from Personnel view:

1. From the **Personnel** navigation menu, select **Constraints**.
2. Click **Personnel Contraints** tab.
   The list of the Personnel Constraints is displayed.

For more details on the mangement of *Operational Rules* with **HOPEX UAF**, see Operational - Constraints.

Business Skills

A Business Skill is a Capability of an Organization (human resource) that has been acquired by training.

To be able to subsequently check that each Capability is implemented by a suitable Business Skill, you must define the required business skills and functionalities, for each business function.

Accessing Business Skills

To access Business Skills:

1. From the **Personnel** navigation menu, select **Constraints**.
2. Click the **Business Skills** tab.
   The list of Business Skills is displayed.

Creating a Business Skill

To create an *Business Skill*:

1. From the **Personnel** navigation menu, select **Constraints**.
   The list of Business Skills appears.
2. Click the **Business Skills** tab.
3. Click **New**.
   The new Business Skill appears in the list.

Properties of a Business Skill

The **Characteristics** property page of the business skill provides access to:

- its **Owner**, by default during creation of the business skill, the current enterprise.
- its **Name**,
- the text of its **Description**.

With **HOPEX UAF** a business skill is described by the following pages:

- the **Structure** page specifies a list of business skill held and the dependencies between them.

  ☛ *For more details on business skill diagrams, see Creating a Business Skill Diagram.*

- the **Fulfillments** page provides access to the list of architecture elements that implement the business skill.

  ☛ *For more details on business skill fulfillment, see Business Skill Fulfillment.*

- The **Capability Usage** page provides access to:
  - the **Owner** section provides access to the Business Skill that use the skill described. For more details on the skills used, see Creating a business skill component in a diagram.
  - the **Business Capability** section provides access to the capabilities that use the skill described. For more details on the skills used, see Describing a Capability.
  - the **Business Function** section provides access to the Operational Performers that use the skill described. For more details on the skills used, see Describing Operational Performers.

Business Skill Fulfillment

To associate an organization with a business skill, you must create a business skill fulfillment.

📖 *A Fulfillment describes the relationship between a logical entity and a physical entity that implements it. The physical entity gives the list of logical entities that fulfill it.*

To specify that a business skill is fulfilled by an organization:

1. Open the **Fulfillments** property page of the business skill that interests you.
2. Click **New**.
   The creation window for a business skill realization opens.
3. Select **Reusing an org-unit**.
4. Select the organization that interests you and click **OK**.
   The business skill fulfillment appears in the list with the name of the selected organization.

Creating a Business Skill Diagram

To create a business skill diagram:

1. In the list of Business Skill, click the **Create a diagram** icon related to the business skill of your choice.
   The diagram creation window opens.

To create a business skill component in a diagram, see Creating a business skill component in a diagram.

To define the dependencies of business skills, see Defining the business skill dependencies

Creating a business skill component in a diagram

The components represented in a business skill diagram are **Business skills**.

> 🕮 *A Business Skill is a Capability of an Organization (human resource) that has been acquired by training.*

To add a sub-skill to the business skill diagram:

- In the object inserting bar, click the **Business Skill Component** .

Defining the business skill dependencies

You can create a dependency link between two business skills to specify that one business skill is required for the other in the context of a business skill.

To create dependency links between two business skills:

- From the object inserting bar, click the **Business skill dependency** button and connect the source business component to the target business component.

> ☛ *To display Capabilities, Capabilities maps and business skills, check option **HOPEX Solutions > Business Process Analysis > Capability visibility** is activated.*

PERSONNEL - ROADMAP

The **Roadmap** aspect of the **Personnel** defines the requirements and functions to ensure that actual persons with the right competencies, and in the right numbers, are available to fulfill.

Organization Resource Catalogs Access

To access an Organization Resource Catalog:

1. From the navigation menu, select **Personnel > RoadMap**.
2. Select **Personnel Availability** tab.
3. Using the arrow at the right of the **Organization Resource Catalog** field, select the Catalog that interests you.
   The tree of catalog components appears below.

Organization Resource Catalogs Creation

To create an Organization Resource Catalog:

1. From the navigation menu, select **Personnel > RoadMap**.
2. Select **Personnel Availability** tab.
3. Click the arrow at the right of the **Organization Resource Catalog** field, select **New**.
   An Organization Resource Catalog is created, the corresponding tree appears below.

Creating an Organization Catalog Item

To create an Organization Catalog Item:

1. Select the Organization Resource Catalog that interests you.
2. In the tree of catalog components, click **Plus** button and select **New > Organization Catalog Item**.
   An **Add** window opens with a list of existing *Actual Organizations*.
3. Create an *Actual Organization* or reuse existing ones and click **Add**.
   The Organization Resource Catalog tree is updated.

Creating an Actual Organizational Position

To create an Actual Organizational Position:

1. Select the Organization Catalog Item that interests you.
2. In the tree of catalog components, click **Plus** button and select **New > Actual Organizational Position**.
   An **Add** window opens with a list of existing *Actual Positions*.
3. Create an *Actual Position* or reuse existing ones and click **Add**.
   The Organization Resource Catalog tree is updated

Organization Resource Catalog Report

To access an Organization Resource Catalog Report:

1. From the navigation menu, select **Personnel > RoadMap**.
2. Select **Personnel Availability Reports** tab.
3. Click the arrow at the right of the **Organization Resource Catalog** field and select the catalog that interests you.
   A Organization Resource Catalog Report opens.

PERSONNEL - TRACEABILITY

The **Traceability** aspect of the **Personnel** view provides access to several reports.

For more details on **Traceability** matrixes and reports, see Using Traceability Aspect.

Two types of matrix and report are provided:

- The **Operational Activities to Functions** matrix and report presenting the *Operational Activities* fulfilled by the specified Action processes.
  - *Operational Activities* in rows

    📖 *An Operational Activity is an end-to-end collection of Activity Actions that creates an outcome for a customer, who may be the ultimate customer or an internal end-user of the Operational Activity.*
  - Action processes in columns may be:

    📖 *An Operational Activity is an end-to-end collection of Activity Actions that creates an outcome for a customer, who may be the ultimate customer or an internal end-user of the Operational Activity.*

    📖 *A Personnel Function Action is an elementary step in process executed by an org-unit. It cannot be broken down. An operation can be industrial (manufacturing a component), logistical (receiving a delivery), or can involve information processing (entering an order).*

    📖 *A Resource Function is the executable representation of a process. the events of the workflow, the tasks to be carried out during the processing, the algorithmic elements used to specify the way in which the tasks follow each other, the information flows exchanged with the participants.*
- The **Operational Resources to Business Skills** matrix and report presenting the skill required by an organization or a post.
  - *Organizations* or *Posts* in rows

    📖 *An organization represents a person or a group of persons that intervenes in the enterprise Personnel Functions or information system. An organization can be internal or external to the enterprise. An internal organization is an organizational element of the enterprise structure such as a department, a service, or a workstation. An internal organization 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 organization is an organization that exchanges flows with the enterprise. Example: Customer, Supplier, Government Office.*

    📖 *A Post is a type of status assigned to an individual or a group of individuals which defines a job in an organization or hierarchy.*
  - *Business Skills* in columns

    📖 *A Business Skill is a Capability of an Organization (human resource) that has been acquired by training.*

RESOURCES VIEWPOINT

The **Resources** viewpoint provides facilities for the definition of solution architectures to implement operational requirements.

This view captures a solution architecture consisting of resources, e.g., organizational, software, artifacts, capability configurations, and natural resources that implement the operational requirements.

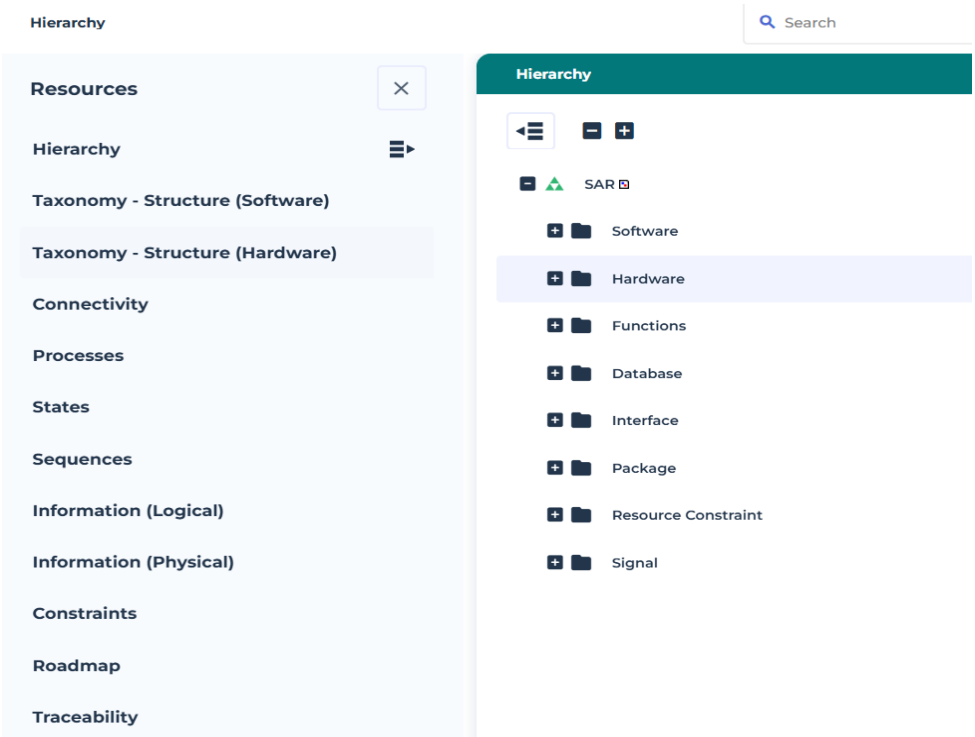
Further design of a resource is typically detailed in SysML or UML.

This chapter comprises the following sections:

- ✓ Resources - Hierarchy;
- ✓ Resources - Taxonomy - Structure (Software);
- ✓ Resources - Taxonomy - Structure (Hardware);
- ✓ Resources - Connectivity;
- ✓ Resources - Processes;
- ✓ Resources - States;
- ✓ Resources - Sequences;
- ✓ Resources - Information (Logical);
- ✓ Resources - Information (Physical);
- ✓ Resources - Constraints;
- ✓ Resources - Roadmap;
- ✓ Resources - Traceability.

RESOURCES - HIERARCHY

The Hierachy aspect of the Resource view helps UAF Architect to manage a solution architecture consisting of resources: organizational, software, artifacts, capability configuration, natural resources that implement the operational requirements.



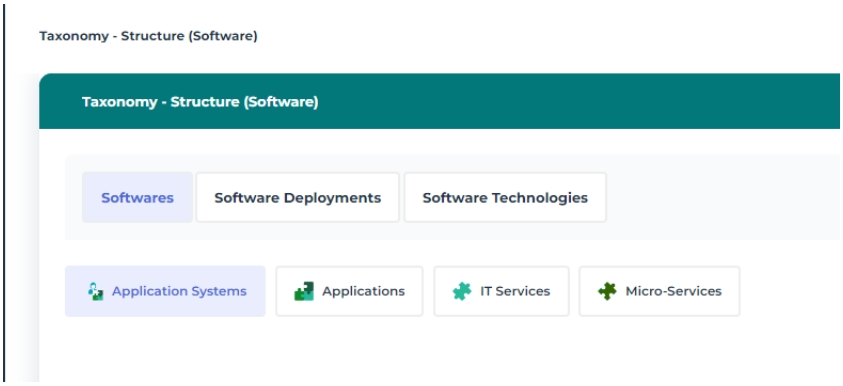
To get more information about the use of folders in **HOPEX** trees, see **HOPEX**, "Handling Trees" chapter of the **HOPEX Common Features** guide.

The hierarchy navigation enables to navigate on the main concepts of the view:

- The current Enterprise and its Enterprise Stages, see Creating a Transformation Stage,
- Software, see Resources - Taxonomy - Structure (Software),
- Hardware, see Resources - Taxonomy - Structure (Hardware),
- Functions, see Resources - Processes,
- Database, see Databases,
- Interface, see Using Interfaces from Resources view,
- Package, see Packages,
- Resource Constraint, see Constraints in the Resource view,
- Signal, see Using Signals from Resources view.

RESOURCES - TAXONOMY - STRUCTURE (SOFTWARE)

The **Taxonomy - Structure (Software)** aspect of the **Resources** view helps UAF Architect to describe the software components of the enterprise.



Describing Software components - Application Systems

Describing an Application System with HOPEX UAF

An Application System is an assembly of other application systems, applications and end users interacting with application components to implement one or several functions.

Accessing Application Systems with HOPEX UAF

To access Application Systems:

1. From the navigation menu, select **Resources > Taxonomy - Structure (Software)**.
2. Click the **Software** tab.
3. Click the **Application System** sub-tab.
   The list of Application Systems of the Enterprise is displayed.

Application Systems diagrams with HOPEX UAF

An *application system* is described by several types of diagrams:

- A *Sketching diagram* is used to enables you to draw diagrams freehand and connect the shapes to objects of the repository using **High Level Operational Concept**.

  📖 *A High Level Operational Concept is a rough draft of a model; unformal representation.*

  ☛ *For further details, see Using High Level Operational Concepts with HOPEX UAF.*

- An *Application System Deployment Architecture* is used to represent the technical architecture chosen for the deployment of each component that support the application system as well as the techniques used for their communications .

  ☛ *For further details, see Describing an Application System Deployement Architecture with HOPEX UAF.*

- An *External Data Flows diagram* describing external data flows of an application system in a given environment. It contains the subject application system and application flows exchanged with partners (other application systems, applications, data stores, org-units or position type.

  ☛ *For further details, see Using a Scenario of Flows Diagram in the Resources view.*

- An *External Interaction diagram* describing external service interactions of an application system in a given environment. It contains the subject application system and its interactions with partners (other application systems, org-units or position types).

  ☛ *For further details, see Describing an Application Environment with HOPEX UAF.*

- An *Internal Architecture diagram* displaying the first level components of an application system, the access points (service/request points) and the connections between the components.

  ☛ *For further details, see Creating an application system structure diagram.*

- An *Internal Data Flows diagram* describing internal data flows of an application system. It contains the agents necessary for the scenario (application systems, applications, microservices, data stores, org-units or position types) and exchanged application flows.

  ☛ *For further details, see Using a Scenario of Flows Diagram in the Resources view.*

Application Systems Properties

The **Characteristics** property page for an application system provides access to several sections.

- The **Identification** section provides access to the following information:
  - the **Name**,
  - its **Owner**, by default during creation of the application system, the current library.
  - the text of its **Description**.
  - the internal **Code**,
  - the **Version number**,
  - **Description**.
- About the **Functional Scope** section of the application system, see Defining Application Functional Scope.
- the **Use Cases** section, see Creating an application Use Case Diagram.
- The **Responsibility** section relates to the person(s) responsible for the application system.
  - Software Designer
  - Local Application Owner

    ☛ *For more details on these roles, see Business Roles of HOPEX IT Architecture..*
- The **Attachments** section is limited to associated attachments.

    ☛ *For more details on other property pages proposed by **HOPEX IT Architecture**, see HOPEX IT Architecture properties pages content.*

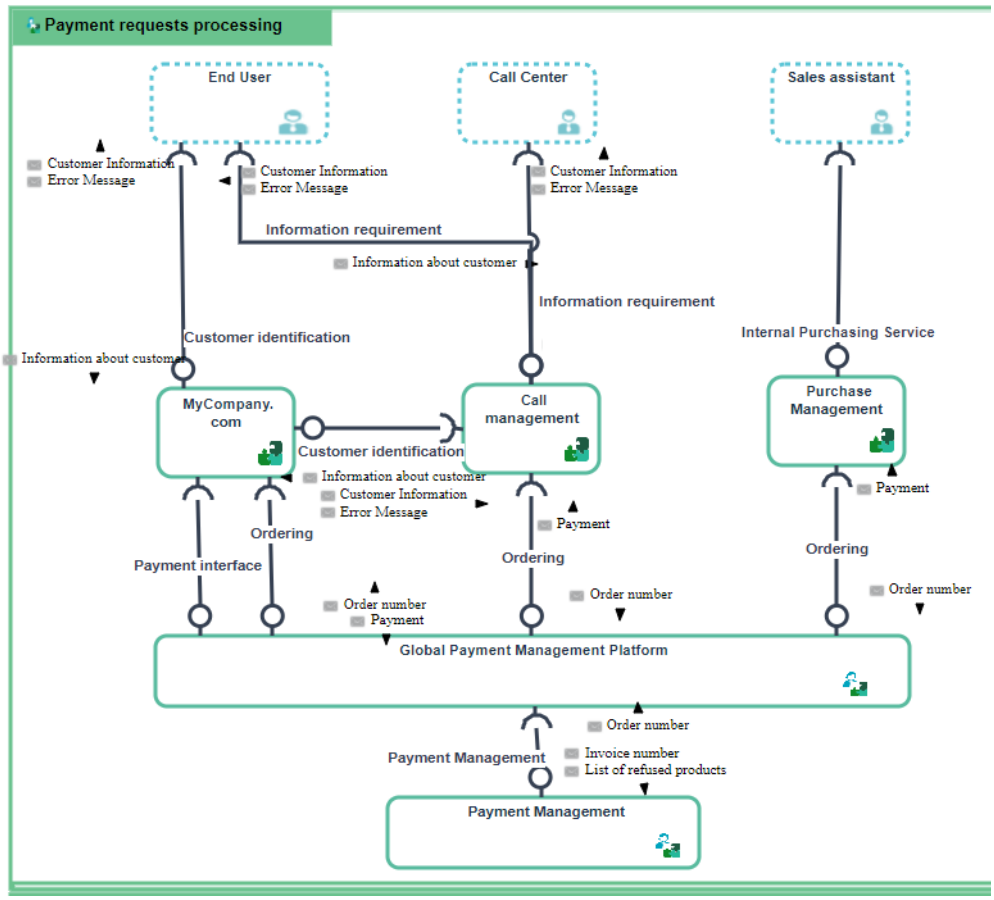
With **HOPEX IT Architecture** an application system is described by other property pages. See HOPEX IT Architecture properties pages content.

Creating an application system structure diagram

This diagram describes the internal structure of an application system:

- services offered or required,
- the application components and their interactions; these are application systems service, applications and microservices,
- the end users interacting with the application components.

```
The following diagram describes the application system
corresponding to purchasing requests processing.
```



The following diagram describes the application system corresponding to purchasing requests processing.

To create an application system structure diagram:

1. Open the **Diagrams** page of the application system of your choice and click **Create a diagram**.
2. Select **Structured diagram > Internal Architecture**.
   The **application system structure diagram** appears.

Adding an application system to an application system structure diagram

To describe an application system that implements another application system, you can add an *application system* of the application system structure diagram.

```
For example, the purchasing requests processing system uses
the "Purchasing Management Platform" and "Payment
Management" application system services.
```

To add an **Application System**:

1. In the objects toolbar of the application system structure diagram, click  **Application System**.
2. Click in the frame of the described application system.
   An addition window box prompts you to choose the **application system** implemented (for example "Payment management").
3. Select an application system.
4. Click **OK**.
   The application system appears in the diagram.

Adding an end user to an application system structure diagram

To specify that an application system, such as purchasing request processing, is activated by internal or external org-units, you will add an associated *end user*.

> *The end user represents an organizational unit interacting at the boundaries of an application system or a logical application system.*

To add an *end user*:

1. In the application system structure diagram objects toolbar, click  **End User** and click in the frame of the diagram.
   An addition window prompts you to choose the **Object Type** that you wish to use: *Org-Unit* or *Position type*.
2. For example, select the **Org-unit** object type.

   > *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 enterprise structure such as a management, department, or job function. It is defined at a level depending on the degree of detail to be provided on the organization (see org-unit type). Example: financial management, sales management, marketing department, account manager. An external org-unit is an external entity that exchanges flows with the enterprise. Example: customer, supplier, government office.*

   > *A position type represents a status assigned to an individual or a group of individuals with the aim of defining an organization or a hierarchy.*
3. Select the org-unit that interests you and click **OK**.
   The actor appears in the diagram.

Describing an application system environment with HOPEX UAF

> *An application system environment allows presenting the other application systems, applications or microservices with which this application system can interact.*

Creating an application system environment

To create an *Application System Environment* from the **Application Systems** navigation menu:

1. Open the **Environments** page of the application system of your choice.
   The list of application system environments appears in the edit area.
2. Click **New**.
   The new application system environment appears in the list, it has the name of the application system followed by "Environment".

Application system environment properties

The **Characteristics** properties page for an application system environment provides access to:

- its **Owner**, by default during creation of an application system environment, the current library.
- its **Name**,
- the text of its **Description**.

With **HOPEX UAF** an application system environment is described by other property pages. See HOPEX IT Architecture properties pages content.

Application system environment diagrams

An *application system environment* is described by several types of diagram:

- an *Application System Environment* diagram describes the exchanges between the subject application system and its partners in a specific context.

  ☛ *For more details, see Describing an application system environment diagram.*

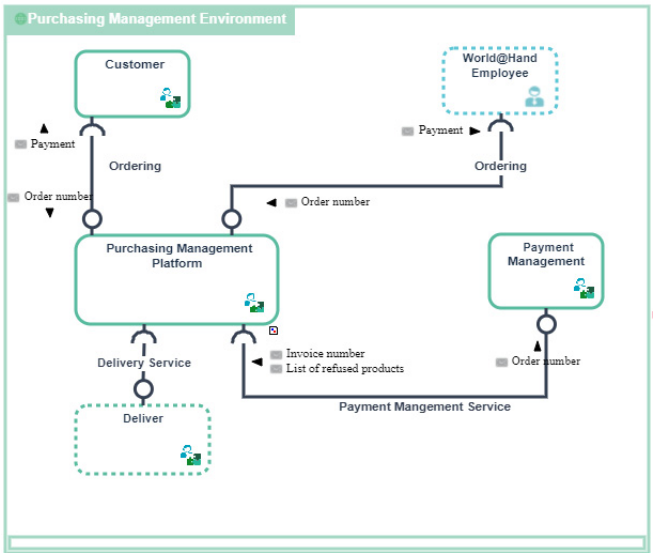
- a *scenario of application system environment flows* presents the flows exchanged between the application services or the microservices used by the described application system in a specific context.

  ☛ *For more details, see Using a Scenario of Flows Diagram in the Resources view.*

Describing an application system environment diagram

An application system environment is described by an **application system environment diagram** that describes the service interactions between the internal application systems, its users and the partner application systems.

> ☛ *For more details on use of a structure diagram, see Application structure diagram*



Application system environment diagram for the Purchasing Requests

```
Purchase requests are formulated by clients or employed
using the "Purchasing Management Platform".

The "Purchasing Management Platform" application system
uses an internal application system for the "Payment
management" and a partner application system for the
"Delivery".
```

The elements of an application system environment diagram are:

- the main *application system* principal described by the environment.

  > 🕮 *An application system is an assembly of other application systems, applications and end users interacting with application components to implement one or several functions.*

- *partner application systems* that represent the other application system with which the main application system described by the environment interacts.

  ```
  In this example, this concerns two loan services offered to
  individuals and companies.
  ```

  > 🕮 *A partner application system is an application system external to the environment of the described application service. The partner*

application system can be a service supplier or a service consumer with respect to application system users.

- The categories of users of services provided by the environment are represented either by an *Org-Unit* or by a *Position Type*.

  📖 *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 enterprise structure such as a management, department, or job function. It is defined at a level depending on the degree of detail to be provided on the organization (see org-unit type). Example: financial management, sales management, marketing department, account manager. An external org-unit is an external entity that exchanges flows with the enterprise. Example: customer, supplier, government office.*

  📖 *A position type represents a status assigned to an individual or a group of individuals with the aim of defining an organization or a hierarchy.*

  `This concerns two user categories: individuals and companies.`

- *Service interactions* between components

  📖 *A Service Interaction represents an interaction for service purpose between entities in a specific context inside or outside a company. These entities can be enterprise org-units, applications, activities or processes, as well as external org-units. The content of this interaction is described in a service interface.*

Describing Software components - Applications

Describing an Application with HOPEX UAF

📖 *An Application is a software component that can be deployed and provides users with a set of functionalities.*

Accessing Applications with HOPEX UAF

To access Applications:

1. From the navigation menu, select **Resources > Taxonomy - Structure (Software)**.
2. Click the **Software** tab.
3. Click the **Applications** sub-tab.
   The list of Application of the Enterprise is displayed.

Application diagrams with HOPEX UAF

An *application* is described by several types of diagrams:

- A *Sketching diagram* is used to enables you to draw diagrams freehand and connect the shapes to objects of the repository using **High Level Operational Concept**.

  A High Level Operational Concept is a rough draft of a model; unformal representation.

  ☛ *For further details, see Using High Level Operational Concepts with HOPEX UAF.*

- An *Application Deployment Architecture* is used to represent the technical architecture chosen for the deployment of each component that support the application as well as the techniques used for their communications .

  ☛ *For further details, see Describing an Application Deployement Architecture with HOPEX UAF.*

- An *External Data Flows diagram* describing external data flows of an application system in a given environment. It contains the subject application system and application flows exchanged with partners (other application systems, applications, data stores, org-units or position type.

  ☛ *For further details, see Using a Scenario of Flows Diagram in the Resources view.*

- An *External Interaction diagram* describing external service interactions of an application system in a given environment. It contains the subject application system and its interactions with partners (other application systems, org-units or position types).

  ☛ *For further details, see Application Environment Diagram presentation.*

- An *Internal Architecture diagram* displaying the first level components of an application, the access points (service/request points) and the connections between the components.

  ☛ *For further details, see Application structure diagram.*

- An *Internal Data Flows diagram* describing internal data flows of an application. It contains the agents necessary for the scenario (application systems, applications, microservices, data stores, org-units or position types) and exchanged application flows.

  ☛ *For further details, see Using a Scenario of Flows Diagram in the Resources view.*

Describing an Application Environment with HOPEX UAF

An application environment is used to represent a use context of an application. An application environment allows presenting the other application systems, applications, microservices or actors with which this application can interact.

An *application environment* is described by several types of diagrams:

- a *scenario of application environment flows* describes the flows exchanged between the described application and its partners:

applications, application systems, IT services or microservices used by the described application in a specific context.

☛ *For more details, see Using a Scenario of Flows Diagram in the Resources view.*

- a *scenario of sequences of flows* presents the agents necessary for the scenario (application, IT services, microservices, data stores) and sequence application flows exchanged.

☛ *For more details, see Using a flow scenario sequence diagram.*

To create an *Application environment:*

1. Open the **Environments** page from the application of you interest.
   The list of application environments appears in the edit area.
2. Click **New**.
   The new application environment appears in the list under the name "Environment" followed by the name of the application.

To access the list of application environments from the **Applications** navigation menu:

❱ Open the **Environments** page from the application of you interest.
The list of application environments appears in the edit area.

Application Environment Diagram presentation

With **HOPEX UAF**, an *application environment* is entirely described by a an application environment diagram that is used to describe the service interactions between the environment applications described, its users and the external applications.

An application environment diagram includes:

- *applications* that represent the environment described.

```
In the example, this concerns the applications used for
buying spare parts.
```

🕮 *An Application is a software component that can be deployed and provides users with a set of functionalities.*

- *applications*, *application services* or *partner microservices* that represent the external elements used in the described environment.

```
This example concerns automated Web services.
```

🕮 *An IT Service is a software component of an application, that can't be deployed alone and that realizes a sub-set of the functionalities of this application either for end users of this application or inside the application (or another application). This includes batch programs.*

- *Service interactions* between components.

🕮 *A Service Interaction represents an interaction for service purpose between entities in a specific context inside or outside a company. These entities can be enterprise org-units, applications, activities or Personnel Functions, as well as external org-units. The content of this interaction is described in an Interface.*

- request and service points.

The **Components** property page of the application environment provides access to partners elements: Applications, Microservices, IT Services, System users.

Describing Software components - IT Services and Microservices

Describing an IT Service with HOPEX UAF

An IT Service is a software component of an application, that can't be deployed alone and that realizes a sub-set of the functionalities of this application either for end users of this application or inside the application (or another application). This includes batch programs.

Accessing IT Services with HOPEX UAF

To access IT Services:

1. From the navigation menu, select **Resources > Taxonomy - Structure (Software)**.
2. Click the **Software** tab.
3. Click the **IT Services** sub-tab.
   The list of IT Services of the Enterprise is displayed.

IT Service diagrams with HOPEX UAF

An *IT Service* is described by several types of diagrams:

- An *Internal Architecture diagram* displaying the first level components of an IT Service, the access points (service/request points) and the connections between the components.

  ☛ *For further details, see Using IT Service Structure Diagram.*

- An *Internal Data Flows diagram* presents the flows exchanged between the described IT services or microservices. A scenario can represent a particular use case of the IT service or more globally all the flows exchanged within this IT service.

  ☛ *For further details, see Using a Scenario of Flows Diagram in the Resources view.*

Describing a Microservice with HOPEX UAF

A Microservice is a software component that can be deployed autonomously, but which does not directly provide an end user service. It can interact with other application services, applications or application systems. This is a deployable software component that uses software technologies. For example: an authentication service, a PDF file printing service.

Accessing Microservices with HOPEX UAF

To access Microservices:

1. From the navigation menu, select **Resources > Taxonomy - Structure (Software)**.
2. Click the **Software** tab.
3. Click the **Microservices** sub-tab.
   The list of Microservices of the Enterprise is displayed.

Microservice diagrams with HOPEX UAF

A *Microservice* is described by several types of diagrams:

- A *Sketching diagram* is used to enables you to draw diagrams freehand and connect the shapes to objects of the repository using **High Level Operational Concept**.

  > 🕮 *A High Level Operational Concept is a rough draft of a model; unformal representation.*
  >
  > ☛ *For further details, see Using High Level Operational Concepts with HOPEX UAF.*

- An *Microservice Deployment Architecture* is used to represent the technical architecture chosen for the deployment of each component that support the microservice as well as the techniques used for their communications .

  > ☛ *For further details, see Describing an Application Deployment Architecture.*

- An *Internal Architecture diagram* displaying the first level components of a microservice, the access points (service/request points) and the connections between the components.

  > ☛ *For further details, see Using a Microservice Structure Diagram.*

- An *Internal Data Flows diagram* presents the flows exchanged between the microservice elements in a given context.

  > ☛ *For further details, see Using a Scenario of Flows Diagram in the Resources view.*

Describing Software Deployments

Describing an Application System Deployement Architecture with HOPEX UAF

> 🕮 *An Application System Deployment Architecture describes one of the configurations possible for deploying an application system. It contains the deployment architectures of application components and specifies the communication protocols (and port numbers) they use to communicate with each other.*

Accessing Application System Deployement Architectures with HOPEX UAF

To access Application System Deployement Architectures:

1. From the navigation menu, select **Resources > Taxonomy - Structure (Software)**.
2. Click the **Software** tab.
3. Click the **Application System Deployement Architectures** sub-tab. The list of Application System Deployement Architectures of the Enterprise is displayed.

Creating an Application System Deployement Architectures with HOPEX UAF

To create an Application System Deployement Architectures:

1. From the navigation menu, select **Resources > Taxonomy - Structure (Software)**.
2. Click the **Software** tab.
3. Click the **Application System Deployement Architectures** sub-tab.
4. Click **New** button.
   The Application System Deployment Architecture is created and its diagram opens.

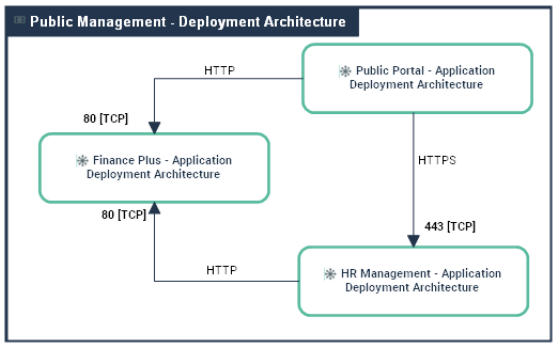
To connect an Application to an Application System Deployement Architectures:

1. From the navigation menu, select **Resources > Taxonomy - Structure (Software)**.
2. Click the **Software** tab.
3. Click the **Application System Deployement Architectures** sub-tab.
   The list of Application System Deployment Architecture appears
4. Select the Application System Deployement Architecture that interests you and use the **Application System** column to connect an applicaiton system.

> ☛ *You can create a Application System Deployment Architecture by creating an ApplicationSystem Deployment Architecture diagram from the application System that interests you. See Application Systems diagrams with HOPEX UAF.*

Using an Application System Deployement Architectures with HOPEX UAF

An application system deployment architecture is described by an application system deployment architecture diagram composed of the following elements:



An application system deployment architecture diagram includes the following elements:

- *Application Deployment Architecture*,

> 🕮 *An Application Deployment Architecture describes one possible deployment configuration of an application. It contains the deployment architectures to be hosted, recommends hosting architectures and identifies required communication techniques (communication protocols and port numbers) they use to communicate with each other. . An*

application may have several deployment architectures (E.g.: autonomous installation, horizontal or vertical deployment, etc.)

☛ *For more details on application deployment architectures, see Describing an Application Deployement Architecture with HOPEX UAF.*

- *Application System Deployment Architectures*,

  📖 *An Application System Deployment Architecture describes one of the configurations possible for deploying an application system. It contains the deployment architectures of application components and specifies the communication protocols (and port numbers) they use to communicate with each other.*

- *microservices*,

  📖 *A Microservice is a software component that can be deployed autonomously, but which does not directly provide an end user service. It can interact with other application services, applications or application systems. This is a deployable software component that uses software technologies. For example: an authentication service, a PDF file printing service.*

- *Deployable Data Packages*,

  📖 *A Deployable Data Package represents a data part of an application deployment that must be hosted and accessed by application services (code) to run. Each deployable data package is associated to required technologies (for data hosting and access) and can host several data structures. Architect can also prescribes a kind of hosting artefact (IaaS/PaaS cloud service or IT server model). Architect can also prescribes a kind of hosting artefact (IaaS/PaaS cloud service or IT server model).*

- *Technical Server Port* and *Technical Client Port*,

  📖 *A Client Port is a point used to request the opening of communications from a technical architecture or an application technical area in compliance with a particular communication protocol (SMTP, HTTP, etc.).*

  📖 *A Server Port is a point used to open communications with a technical architecture or an application technical area in compliance with a particular communication protocol (SMTP, HTTP, etc.).*

- *Package Connections*.

  📖 *A Package Connection represents a connection requirement between deployable packages. A client (package) requires a connection to a server (package). The connection is defined by a Connection Type and can be characterized with technical flow measures.*

Describing an Application Deployement Architecture with HOPEX UAF

📖 *An Application Deployment Architecture describes one possible deployment configuration of an application. It contains the deployment architectures to be hosted, recommends hosting architectures and identifies required communication techniques (communication protocols and port numbers) they use to communicate with each other. . An application may have several deployment architectures (E.g.: autonomous installation, horizontal or vertical deployment, etc.)*

Accessing Application Deployement Architectures with HOPEX UAF

To access Application Deployement Architectures:

1. From the navigation menu, select **Resources > Taxonomy - Structure (Software)**.
2. Click the **Software** tab.

3. Click the **Application Deployement Architectures** sub-tab.
   The list of Application Deployement Architectures of the Enterprise is displayed.

Creating an Application Deployement Architectures with HOPEX UAF

To create an Application Deployement Architectures:

1. From the navigation menu, select **Resources > Taxonomy - Structure (Software)**.
2. Click the **Software** tab.
3. Click the **Application Deployement Architectures** sub-tab.
4. Click **New** button.
   The Application Deployment Architecture is created and its diagram opens.

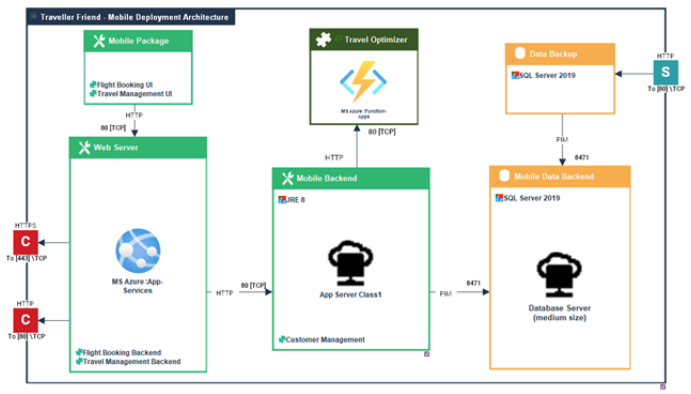
To connect an Application to an Application Deployement Architectures:

1. From the navigation menu, select **Resources > Taxonomy - Structure (Software)**.
2. Click the **Software** tab.
3. Click the **Application Deployement Architectures** sub-tab.
   The list of Application Deployment Architecture appears
4. Select the Application Deployement Architecture that interests you and use the **Application** column to connect an applicaiton.

> ☛ *You can create a Application Deployment Architecture by creating an Application Deployment Architecture diagram from the application that interests you. See Application diagrams with HOPEX UAF.*

Using an Application Deployement Architectures with HOPEX UAF

An application deployment architecture allows you to describe the overview elements that must be deployed to implement an application architecture: *Deployable Application modules*, *Deployable Data Modules* as well as *Package connections* used for data exchange.



An deployment architecture diagram includes the following elements:

- *Deployable Application Packages*,

  A Deployable Application Package is a split of application code according to technical criteria for hosting purpose. For example, it may be N tiers, Front End/Back End/… or GUI/Business Logic/Database etc... Each deployable application package is associated to required technologies (for running) and can host code for several IT services. Architect can also prescribes a kind of hosting artefact (IaaS/PaaS cloud service or IT server model).

- *Deployable Data Packages*,

  A Deployable Data Package represents a data part of an application deployment that must be hosted and accessed by application services (code) to run. Each deployable data package is associated to required technologies (for data hosting and access) and can host several data structures. Architect can also prescribes a kind of hosting artefact (IaaS/PaaS cloud service or IT server model). Architect can also prescribes a kind of hosting artefact (IaaS/PaaS cloud service or IT server model).

- *microservices*,

  A Microservice is a software component that can be deployed autonomously, but which does not directly provide an end user service. It can interact with other application services, applications or application systems. This is a deployable software component that uses software technologies. For example: an authentication service, a PDF file printing service.

- *Technical Server Port* and *Technical Client Port*,

  A Client Port is a point used to request the opening of communications from a technical architecture or an application technical area in compliance with a particular communication protocol (SMTP, HTTP, etc.).

  A Server Port is a point used to open communications with a technical architecture or an application technical area in compliance with a particular communication protocol (SMTP, HTTP, etc.).

- *Package Connections*.

  A Package Connection represents a connection requirement between deployable packages. A client (package) requires a connection to a server (package). The connection is defined by a Connection Type and can be characterized with technical flow measures.

Describing Software Technologies

A Software Technology is a basic component necessary for operation of business applications. Software technologies include all basic software such as: application server, electronic mail server, software components for presentation, data entry, storage, business information sharing, operating systems, middleware, navigators, etc.

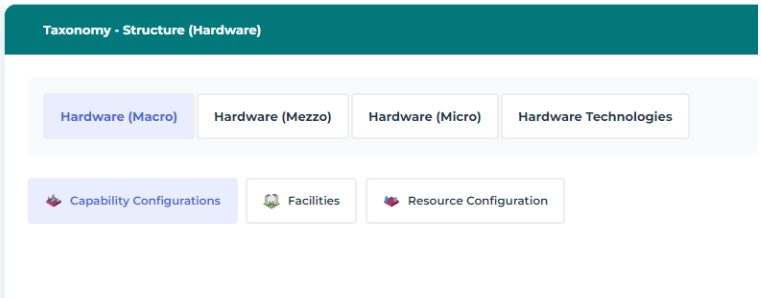
To access Software Technologies with **HOPEX UAF**:

1. From the navigation menu, select **Resources > Taxonomy - Structure (Software)**.
2. Click the **Software Technologies** tab.
   The list of Software Technologies of the Enterprise is displayed.

For information regarding Concept Diagram in HOPEX, see "Describing a Software Technology" chapter in HOPEX IT Architecture guide

RESOURCES - TAXONOMY - STRUCTURE (HARDWARE)

The **Taxonomy - Structure (Hadware)** aspect of the **Resources** view helps UAF Architect to describe the hardware components of the enterprise.



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

Communications between these components are represented by service interactions and the equipment means supporting these service interactions are the communication channels.

> 🕮 *A Capacity Configuration is a combination of physical assets and organization configured to provide a capability.*

Services offered by the system to its users are represented by service points. Service points are physically supported by communication ports that enable access to communication means of the system.

Describing Hardware (Macro)

The **Performed Process** property of any Harware (Macro components provides access to the Resource Functions executed. See Using Resource Functions.

Describing Capability Configurations

> 🕮 *A Capacity Configuration is a combination of physical assets and organization configured to provide a capability.*

Accessing Capability Configurations with HOPEX UAF

To access Capacity Configurations:

1. From the navigation menu, select **Resources > Taxonomy - Structure (Hardware)**.

2. Click the **Hardware (Macro)** tab.
3. Click the **Capability Configurations** sub-tab.
   The list of Capacity Configurations of the Enterprise is displayed.

Using a Capacity Configuration Internal Architecture Diagram:

The components that can be used to describe a Capacity Configuration in a Capacity Configuration Internal Architecture Diagram.

- *Capacity Configuration*
- *Facility*

  A Facility is a model of site of interest for the enterprise. Examples: Data Center, Factory or Outlet

  ☛ *For more details on Facilities, see Describing a Facility.*
- *Resource Configuration*

  A Resource Configuration is a set of physical and human resources configured to provide a business capability.

  ☛ *For more details on Resource Configurations, see Describing a Resource Configuration.*
- *IT Infrastructure*

  An IT Infrastructure is composed of several connected computer devices (IT network nodes or computing systems) and computer networks.

  ☛ *For more details on IT infrastructures, see Describing an IT Infrastructure.*
- *Computer Networks*

  An IT network is set of IT equipment components (e.g.: routers, switches, firewalls) that allow remote communications between computing devices (e.g.: IT server). An IT network can be broken down into sub-networks.

  ☛ *For more details on IT infrastructures, see Describing an IT network.*
- *Hardware*

  An IT Infrastructure is composed of several connected computer devices (IT network nodes or computing systems) and computer networks.

  ☛ *For more details on Hardwares, see Describing an Hardware.*

To describe that a Capacity Configuration uses operators, you will create:

An organization represents a person or a group of persons that intervenes in the enterprise Personnel Functions or information system. An organization can be internal or external to the enterprise. An internal organization is an organizational element of the enterprise structure such as a department, a service, or a workstation. An internal organization 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 organization is an organization that exchanges flows with the enterprise. Example: Customer, Supplier, Government Office.

A Post is a type of status assigned to an individual or a group of individuals which defines a job in an organization or hierarchy.

A Capacity Configuration is created to assure one or several services. Expected and realized services are represented by:

- *service points*

  📖 *A service point is a point from which a system receives a request from another system and provides the requested service.*

- *request points*

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

  ☛ *For more details, see Describing Service and Request Points.*

In a Capacity Configuration internal architecture diagram, *Service Interactions* enable representation of exchanges between organizational entities.

📖 *A Service Interaction represents an interaction for service purpose between entities in a specific context inside or outside a company. These entities can be enterprise org-units, applications, activities or Personnel Functions, as well as external org-units. The content of this interaction is described in an Interface.*

Exchange terms are defined by a *service interface* assigned to the service interaction.

📖 *An Interface is a template of a contract between entities (organizational, IT …). The contract is described by available operations which can be triggered trough messages exchanged by roles (vendor, buyer..).*

You can define service interactions between:

- Two components of *Capacity Configuration* type to represent exchanges between these entities,
- A component of *Capacity Configuration* type and an *IT infrastructure* to represent the terms of use of the equipment resource by the organizational resource.
- two components of *IT infrastructure* type to represent the terms of use of one IT resource by another in the context of the modeled Capacity Configuration.
- a *service point* and one or more *Capacity Configuration* type components to represent implementation of the service within the Capacity Configuration,
- a component of *Capacity Configuration* type and a *request point* to represent that the entity calls a resource of an external organization.

  ☛ *For further details, see Creating a Service Interaction from a diagram.*

Network channels support the transfer of information from one hardware asset to another.

📖 *A Network Channel is a physical connector between resource elements. It supports service interactions defining communication protocols between physical resources It connects external resource elements through their Communication Ports.*

Communication ports enable connection of Capacity Configuration physical assets with external equipment elements.

📖 *A Communication Port is a physical point of communication with a resource. It adheres to the specific communication protocol. A communication port implements service and requests points.*

For more details ont Ports and Network Channels, see Describing technical communications.

Describing a Facility

A Facility is a model of site of interest for the enterprise. Examples: Data Center, Factory or Outlet

Accessing Facilities with HOPEX UAF

To access Facilities:

1. From the navigation menu, select **Resources > Taxonomy - Structure (Hardware)**.
2. Click the **Hardware (Macro)** tab.
3. Click the **Facilities** sub-tab.
   The list of Facilities of the Enterprise is displayed.

Using a Facility Internal Architecture Diagram:

In a Facility diagram, you can insert:

- *IT Infrastructures*; see Describing Hardware (Mezzo)

  An IT Infrastructure is composed of several connected computer devices (IT network nodes or computing systems) and computer networks.

- *Computer Networks*

  An IT network is set of IT equipment components (e.g.: routers, switches, firewalls) that allow remote communications between computing devices (e.g.: IT server). An IT network can be broken down into sub-networks.

  ☛ *For more details on IT infrastructures, see Describing an IT network.*

- *Hardware*

  An IT Infrastructure is composed of several connected computer devices (IT network nodes or computing systems) and computer networks.

  ☛ *For more details on Hardwares, see Describing an Hardware.*

- *Organization* and *Post*

  An organization represents a person or a group of persons that intervenes in the enterprise Personnel Functions or information system. An organization can be internal or external to the enterprise. An internal organization is an organizational element of the enterprise structure such as a department, a service, or a workstation. An internal organization 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 organization is an organization that exchanges flows with the enterprise. Example: Customer, Supplier, Government Office.

📖 *A Post is a type of status assigned to an individual or a group of individuals which defines a job in an organization or hierarchy.*

- *service points* and *request points*

  📖 *A service point is a point from which a system receives a request from another system and provides the requested service.*

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

  ☛ *For more details, see Describing Service and Request Points.*

Network channels support the transfer of information from one hardware asset to another.

📖 *A Network Channel is a physical connector between resource elements. It supports service interactions defining communication protocols between physical resources It connects external resource elements through their Communication Ports.*

Communication ports enable connection of Capacity Configuration physical assets with external equipment elements.

📖 *A Communication Port is a physical point of communication with a resource. It adheres to the specific communication protocol. A communication port implements service and requests points.*

For more details ont Ports and Network Channels, see Describing technical communications.

Describing a Resource Configuration

📖 *A Resource Configuration is a set of physical and human resources configured to provide a business capability.*

Accessing Resource Configurations with HOPEX UAF

To access Resource Configurations:

1. From the navigation menu, select **Resources > Taxonomy - Structure (Hardware)**.
2. Click the **Hardware (Macro)** tab.
3. Click the **Resource Configurations** sub-tab.
   The list of Resource Configurationss of the Enterprise is displayed.

Using a Resource Configuration Diagram

In a resource configuration diagram, you can insert:

- *IT Infrastructures*; see Describing Hardware (Mezzo)

  📖 *An IT Infrastructure is composed of several connected computer devices (IT network nodes or computing systems) and computer networks.*

- *Computer Networks*

  📖 *An IT network is set of IT equipment components (e.g.: routers, switches, firewalls) that allow remote communications between*

computing devices (e.g.: IT server). An IT network can be broken down into sub-networks.

☛ *For more details on IT infrastructures, see Describing an IT network.*

- *Hardware*

  An IT Infrastructure is composed of several connected computer devices (IT network nodes or computing systems) and computer networks.

  ☛ *For more details on Hardwares, see Describing an Hardware.*

- *Position types* or *Org-Units*.

  An organization represents a person or a group of persons that intervenes in the enterprise Personnel Functions or information system. An organization can be internal or external to the enterprise. An internal organization is an organizational element of the enterprise structure such as a department, a service, or a workstation. An internal organization 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 organization is an organization that exchanges flows with the enterprise. Example: Customer, Supplier, Government Office.

  ☛ *A Post is a type of status assigned to an individual or a group of individuals which defines a job in an organization or hierarchy.*

- Ports and network channels, see Describing technical communications.
- service and request points, see Describing Service and Request Points.
- Service Interaction, see Connecting a Service Interaction to a Network Channel.

  A Service Interaction represents an interaction for service purpose between entities in a specific context inside or outside a company. These entities can be enterprise org-units, applications, activities or processes, as well as external org-units. The content of this interaction is described in a service interface.

Describing Hardware (Mezzo)

Describing an IT Infrastructure

An IT Infrastructure is composed of several connected computer devices (IT network nodes or computing systems) and computer networks.

You can describe the components of an *Infrastructure* in an infrastructure assembly diagram.

Accessing IT Infrastructure with HOPEX UAF

To access IT Infrastructures:

1. From the navigation menu, select **Resources > Taxonomy - Structure (Hardware)**.
2. Click the **Hardware (Macro)** tab.
3. Click the **IT Infrastructure** sub-tab.
   The list of IT Infrastructures of the Enterprise is displayed.

Using an IT Infrastructure diagram

In an infrastructure diagram, you can insert:

You can insert in this diagram:

- *IT servers*, *Computer devices* and *IoT Devices*, see Describing Hardware (Micro),
- *Computer Networks*

  📖 *An IT network is set of IT equipment components (e.g.: routers, switches, firewalls) that allow remote communications between computing devices (e.g.: IT server). An IT network can be broken down into sub-networks.*

  ☛ *For more details on IT infrastructures, see Describing an IT network.*
- Ports and network channels, see Describing technical communications.
- service and request points, see Describing Service and Request Points.
- service interactions, see Resources - Sequences.

Describing an IT network

📖 *An IT network is set of IT equipment components (e.g.: routers, switches, firewalls) that allow remote communications between computing devices (e.g.: IT server). An IT network can be broken down into sub-networks.*

To access IT Networks:

1. From the navigation menu, select **Resources > Taxonomy - Structure (Hardware)**.
2. Click the **Hardware (Macro)** tab.
3. Click the **IT Networks** sub-tab.
   The list of IT Networks of the Enterprise is displayed.

The IT Network diagram is an IT Infrastructure diagram. For more details on this type of diagram, see Using an IT Infrastructure diagram.

Describing an Hardware

📖 *Non-IT Hardware can embed computers. Together with their embedded computers, they provide information and IS services. Examples: Connected Truck with Delivery Calendar Application and connected Drone with Online Payment Application. Hardware device can also provide hardware functionalities. Example: Connected fridge providing ordering functionalities and of course a freezing hardware functionality and connected drones fly and provide Online Payment.*

To access Hardwares:

1. From the navigation menu, select **Resources > Taxonomy - Structure (Hardware)**.
2. Click the **Hardware (Macro)** tab.
3. Click the **Hardware** sub-tab.
   The list of Hardwares of the Enterprise is displayed.

The Hardware diagram is an IT Infrastructure diagram. For more details on this type of diagram, see Using an IT Infrastructure diagram.

Describing Hardware (Micro)

The types of devices commonly available in **HOPEX UAF** are:

- Bridge,
- Firewall,
- Hub,
- IT Server,

  An IT Server is an IT component providing a service to users connected via an IT network. This IT component can house databases and run applications.

- Printer,
- Network Device,

  A Network Device can host and run Software Technology. Conjointly with its hosted software, it provides services. This consists of, for example: Wifi Access Point, Firewall, router, switch, printer, Hard Drive.

- Router,
- Satellite,
- Switch.
- Modem,
- Wifi Hotspot,
- Cloud Service.

  The Cloud Service (considered as an IoT device) can be used in a deployment architecture. The Cloud service picture appears in the frame of the corresponding deployable package.

  *The Cloud Services are provided by your administrator using the module import features. For more details, see **Modules > Importing a Module** documentation.*

To access the list of a specific type of device:

1. From the navigation menu, select **Resources > Taxonomy - Structure (Hardware)**.
2. Click the **Hardware (Micro)** tab.
3. From the list, select the type on component that interests you.

Describing Hardware Technologies

To access Hardware Technologies with **HOPEX UAF**:

1. From the navigation menu, select **Resources > Taxonomy - Structure (Hardware)**.
2. Click the **Hardware Technologies** tab.
   The list of Hardware Technologies of the Enterprise is displayed.

Describing technical communications

Communication ports

Communication Ports are physical points of communication that can be defined in technical infrastructures and Capacity Configurations.

> 🕮 *A Communication Port is a physical point of communication with a resource. It adheres to the specific communication protocol. A communication port implements service and requests points.*

Communication Ports assure physical transfer of information exchanged on service points and request points.

Communication ports comply with specific "Communication Protocols". See Network communication protocols.

Network channels

The *network channels* connect hardware resources between each other, to organizational resources or to communication ports.

> 🕮 *A Network Channel is a physical connector between resource elements. It supports service interactions defining communication protocols between physical resources It connects external resource elements through their Communication Ports.*

Creating a network channel

To create network channel:

1. In the object inserting bar of the resource assembly diagram, click **Network channel** .
2. Draw a link between the two communication entities.
   The channel appears directly in the diagram.

To define the communication protocol associated with the channel:

1. Open the **Supported Protocols** property page and click **Connect**.
2. In the query window that appears, select the communication protocol that interests you and click **Connect**.
   The protocol name appears alongside the channel.

Network communication protocols

A *Communication Protocol* is supported by network channel.

For example, an HTTPS protocol is based on an HTTP protocol for transport, those protocols are based on TCP, which is itself based on Ethernet.

A user may wish to build a customized layer of communication protocols and assign these to communication ports and communication channels.

> ☛ *Communication protocols supported by a communication port must be compatible with the communication ports to which they are connected.*

Connecting a Service Interaction to a Network Channel

To indicate that a service interaction is supported by network channel:

1. In the IT Network diagram objects toolbar, click the **Link** button.
2. Draw a link between the service interaction and network channel supporting it.
   A dotted line appears in the diagram.

To access the list of service interactions supported by a network channel:

- Open the **Managed service interactions** property page of the network channel you are interested in.
  The name of the service interaction appears in the list.

  ☛ *You can use the **Connect** button to connect other interactions to network channel.*

RESOURCES - CONNECTIVITY

The **Connectivity** aspect of the **Resources** view helps UAF Architect in the specification of service interfaces and signals.

Using Signals from Resources view

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

To access the list of *Signal*:

1. From the **Resources** navigation menu, select **Connectivity**.
2. Click the **Signals** tab.
   The list of the Signals is displayed.

For more details on the mangement of *Signals* with **HOPEX UAF**, see Using Signals.

Using Interfaces from Resources view

An Interface is a template of a contract between entities (organizational, IT …). The contract is described by available operations which can be triggered trough messages exchanged by roles (vendor, buyer..).

To access the list of *Interfaces*:

1. From the **Resources** navigation menu, select **Connectivity**.
2. Click the **Interfaces** tab.
   The list of the Interfaces is displayed.

For more details on the mangement of *Interfaces* with **HOPEX UAF**, see Describing an Interface.

RESOURCES - PROCESSES

The **Processes** aspect of the **Resources** view describes Personnel and resource functions, their Inputs/Outputs, resource function actions and flows between them.

Using Personnel Functions from Resource view

> 🕮 *A Personnel Function is a set of operations performed by org-units within a company or organization, to produce a result. It is depicted as a sequence of operations, controlled by events and conditions. In the BPMN notation, the Personnel Function represents a sub-Personnel Function from the organizational point of view.*

To access the list of *Personnel Functions*:

1. From the **Resources** navigation menu, select **Processes**.
2. Click the **Personnel Functions** tab.
   The list of the Personnel Functions is displayed.

For more details on the use of *Personnel Functions* with **HOPEX UAF**, see Personnel Functions.

Using Resource Functions

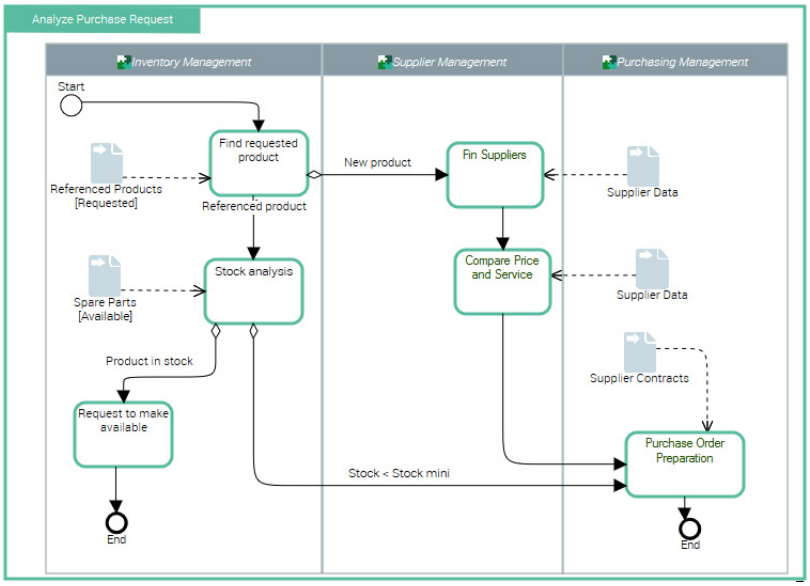
> 🕮 *A Resource Function is the executable representation of a process. the events of the workflow, the tasks to be carried out during the processing, the algorithmic elements used to specify the way in which the tasks follow each other, the information flows exchanged with the participants.*

In detailed specification phase, the progress of tasks implemented in an IT service can also be modeled by a *Resource Function*. More generally, operation of an architecture element can be described by a Resource Function modeling, for example, sequence flow of screens presented to the user.

Resource Function diagram example

The diagram below represents purchase request processing.

- A product search is carried out from the referenced products repository.
- If the product is new, search for a supplier and comparative study of prices is carried out. An order is then sent and the process ends.
- If the product is referenced, stock is analyzed.
- If stock is sufficient, a "Make available" request is activated and the process ends.
- If stock is less than minimum stock, an order is sent to the supplier and the process ends.



For information regarding Resource Functions (System Processes) in HOPEX, see "Describing System Processes" chapter in HOPEX IT Architecture guide

Accessing Resource Functions

To access the list of Resource Functions:

1. From the **Resources** navigation menu, select **Processes**.
2. Click the **Resource Functions** tab.
   The list of the Resource Functions is displayed.

Using a Resource Function Tree diagram

The **Resource Function Tree diagram** describes the tree structure of the current Resource Function. Such diagram describes the current Resource Function and its sub-Resource Functions.

Using a Resource Function diagram

The **Resource Function diagram** uses notation proposed by BPMN standard. The Resource Function algorithm can be expressed by sequencing of tasks and decisions.

Creating a Resource Function Action

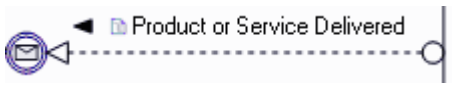
Resource Function Action correspond to Resource Function steps.

> 🕮 *A Resource Function Action is an elementary step that is included within a Resource Function. A Resource Function Action is used when the work in the Resource Function is not broken down to a finer level of the process. Generally, an end-user and/or an IT service are used to perform the Resource Function Action when it is executed.*

Message flows

Message flows represent exchanges between the Resource Function and the exterior.

> 🕮 *A Message Flow represents circulation of information within a service interface. A message flow transports its content.*

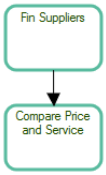


> ☛ *A message flow can be linked to an event of message type.*

Sequence flows

Organization of tasks in the Resource Function is represented by *sequence flows* between tasks.

> 🕮 *A Sequence Flow is used to show the order in which steps of an service contract will be performed. A sequence flow has only one source and only one target.*



> ☛ *For more information on managing sequence flows, see "Describing Operations Sequence" chapter in the HOPEX Business Process Analysis guide.*

Events

Events represent facts occurring during process execution.

> 🕮 *An Event represents a fact or an action occurring in the system, such as updating client information. It is managed by a broker. An application indicates that it can produce the event by declaring that it publishes it. If an application is interested in an event, it declares that it subscribes to the event.*

An example is the start or end of the Resource Function.

Start

Final

The event can also be sending or receiving a message flow.

Gateways

📖 *A Gateway represents elements that are used to control how sequence flows interact as they converge and diverge within a process.*

☛ *For more information on managing gateways, see "Using gateways" chapter in guide HOPEX Business Process Analysis.*

Creating a participant in a Resource Function Diagram

In a Resource Function diagram, a participant enables grouping of tasks assigned to an application or service.

RESOURCES - STATES

The **States** aspect of the **Resources** view addresses the evolution of objects dedicated to software and hardware resources over the time using *State Machine* objects represented by a *State Machine diagram*.

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

In the **Resources** view, the state machines can be connected to the follwing object types:

- For Software States:
  - Application,

    > 🕮 *An Application is a software component that can be deployed and provides users with a set of functionalities.*

  - Application system,

    > 🕮 *An Application System is an assembly of other application systems, applications and end users interacting with application components to implement one or several functions.*

  - IT Service,

    > 🕮 *An IT Service is a software component of an application, that can't be deployed alone and that realizes a sub-set of the functionalities of this application either for end users of this application or inside the application (or another application). This includes batch programs.*

  - Microservice,

    > 🕮 *A Microservice is a software component that can be deployed autonomously, but which does not directly provide an end user service. It can interact with other application services, applications or application systems. This is a deployable software component that uses software technologies. For example: an authentication service, a PDF file printing service.*

- For Hardware States:
  - Bridge,
  - *Capacity Configuration*
  - Cloud Service.

    > 🕮 *The Cloud Service (considered as an IoT device) can be used in a deployment architecture. The Cloud service picture appears in the frame of the corresponding deployable package.*

  - *Computer Networks*

    > 🕮 *An IT network is set of IT equipment components (e.g.: routers, switches, firewalls) that allow remote communications between*

computing devices (e.g.: IT server). An IT network can be broken down into sub-networks.

- *Facility*

  A Capacity Configuration is a combination of physical assets and organization configured to provide a capability.

- Firewall,
- Hardware,
- Hub,
- *IT Infrastructure*

  An IT Infrastructure is composed of several connected computer devices (IT network nodes or computing systems) and computer networks.

- IT Server,

  An IT Server is an IT component providing a service to users connected via an IT network. This IT component can house databases and run applications.

- Printer,
- Network Device,

  A Network Device can host and run Software Technology. Conjointly with its hosted software, it provides services. This consists of, for example: Wifi Access Point, Firewall, router, switch, printer, Hard Drive.

- Router,
- Satellite,
- Switch.
- Modem,
- Wifi Hotspot,

To create a *State Machine* from the **Resources** navigation pane.

1. Select **States**.
2. Select Software States tab or Hardware States tab, depending on the type of the object you want to connect.
   The list of State Machines connected to the selected object types is displayed.
3. Click the **New** button.
   A creation wizard opens.
4. Enter the **Name** of the new State Machine.
5. Select an **Object Type**.
6. Select (or create) the object in the **Subject of State Machine** box.
7. Click **OK**.
   The state machine is created.

For more details on the use of *State Machines* with **HOPEX UAF**, see Strategic - States.

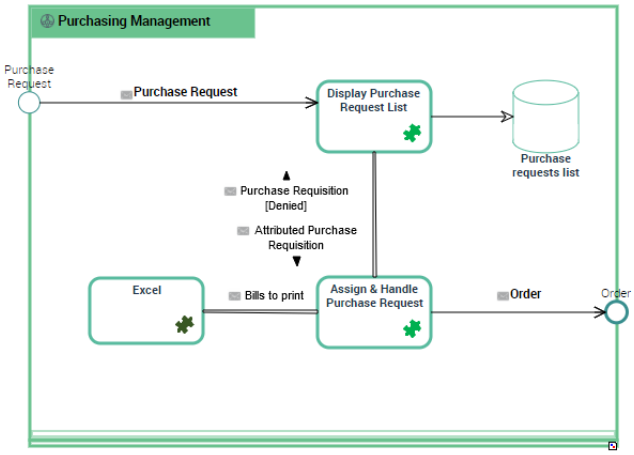
RESOURCES - SEQUENCES

The **Sequences** aspect of the **Resources** view helps UAF Architect to manage the several scenario of flows in the software architectures.

Using a Scenario of Flows Diagram in the Resources view

In the **Resources** view, a Scenario of Flow Diagram can be built for an Application Environment, an Application, an Application System or an Application System Environment. This diagram is used to describe the exchanges inside the described object in a specific context.

The scenario of application flow diagram below describes the "Purchase request management" application.



Example of a Scenario of Application Flows for "Managing Purchase Orders".

In a scenario of application flows diagram, the elements represented are:

- IT Services,services, see Describing an IT Service with HOPEX UAF,
- Microservices, see Describing a Microservice with HOPEX UAF,
- internal or external local application data stores, see Using Data Stores,
- *System Triggering Events* and *System Triggered Events*, see Creating a System Triggering Event.

The interactions offered between these elements:

- application flows that carry a content,
- application flow channels that group a number of application flows on a single link,
- application data channels that represent the interactions between the application data stores.

Accessing Scenarios of flows in the Resources view

Accessing Scenarios of Application System Flows

A **scenario of application system environment flows** describes the flows exchanged between the described application system components: applications, application systems, IT services or microservices used by the described application system in a specific context.

☛ *For more details, see Using a Scenario of Application Flows Diagram.*

To access the a *scenario of application system environment flows* from the **Resources > Sequences** navigation pane.

1. Select **Scenario** tab.
2. Click **Scenarios of Application System Environment Flows** sub-tab.
   The list of *scenarios of application system environment flows* is displayed.

Accessing Scenarios of Application Flows

A **scenario of application flows** describes the flows exchanged between the described application and its components: applications, application systems, IT services or microservices used by the described application in a specific context.

☛ *For more details, see Using a Scenario of Application Flows Diagram.*

To access the a *scenario of application flows* from the **Resources > Sequences** navigation pane.

3. Select **Environment Scenario** tab.
4. Click **Scenarios of Application Flows** sub-tab.
   The list of *scenarios of application flows* is displayed.

Accessing Scenarios of Application System Environment Flows

A **scenario of application system environment flows** describes the flows exchanged between the described application system and its partners: applications, application systems, IT services or microservices used by the described application system in a specific context.

☛ *For more details, see Using a Scenario of Application Flows Diagram.*

To access the a *scenario of application system environment flows* from the **Resources > Sequences** navigation pane.

5. Select **Environment Scenario** tab.
6. Click **Scenarios of Application System Environment Flows** sub-tab.
   The list of *scenarios of application system environment flows* is displayed.

Accessing Scenarios of Application Environment Flows

A **scenario of application environment flows** describes the flows exchanged between the described application and its partners: applications, application

systems, IT services or microservices used by the described application system in a specific context.

> ☛ *For more details, see Using a Scenario of Application Flows Diagram.*

To access the a *scenario of application environment flows* from the **Resources > Sequences** navigation pane.

7. Select **Environment Scenario** tab.
8. Click **Scenarios of Application Environment Flows** sub-tab.
   The list of *scenarios of application environment flows* is displayed.

Using diagrams of Scenario of Flows

Creating a Scenario of Application Flows diagram

To create a scenario of application flows from the **Applications** navigation menu:

1. Right-click the application that interests you and click **Create Diagram**.
2. In the Create a diagram window, select **Structured diagram > Internal Architecture**.
   The Scenario of Application Flows Diagram appears.

Adding an IT service to the scenario of application flows

> 🕮 *An IT service is a software component of an application, that can't be deployed alone and that realizes a sub-set of the functionalities of this application either for end users of this application or inside the application (or another application). This includes batch programs.*

To add an **IT service**:

1. In the objects toolbar of the scenario of application flows, click **IT Service**.
2. Click in the described application frame.
   An addition window box prompts you to choose the **IT Service** implemented (for example "Customer management").
3. Select the application service required and click **OK**.
   The application service appears in the diagram.

You can add a micoservice in the same way.

> 🕮 *A microservice is a software component that can be deployed autonomously, but which does not directly provide an end user service. It can interact with other application services, applications or application systems. This is a deployable software component that uses software technologies. For example: an authentication service, a PDF file printing service.*

Managing application flows in a scenario of application flows

Creating an application flow with content

> 🕮 *An application flow represents the use of a flow between two agents (e.g., applications) in a usage context (represented by a*

scenario of flows). An application flow is based on a flow, which represents the reference flow in context.

The application flows exchanged between the IT services, the microservices or the Application ports of a scenario of application flows are associated with a *content*.

> 🕮 *The content designates the content of a message or an event, independent of its structure. This structure is represented by an XML schema linked to the content. A content may be used by several messages, since it is not associated with a sender and a destination. There can be only one content per message or event, but the same content can be used by several messages or events.*

You must directly specify the *content* of an *application flow* directly on flow creation.

To create the *application flow*:

1. In the objects toolbar of the scenario of application flows, click **Application flow** and select the **Type of application flow** that interests you.
   - **Result/Provision** associated to a service result,
   - **Call/Request** associated to a service request,
   - **Signal** associated to an information exchange.

   > ☛ *For more details, see Associating a service interface used to an application flow.*
2. Click the first object representing the sender of the flow and, holding the mouse button pressed, draw a link to the object receiving the flow.
   The **Application Flow Creation** dialog box opens.
3. In the **Content** drop-down list, select the content you wish to associate with the flow.
   A tree of the *Flow Measures* used by other application flows with the same content is displayed.

   > ☛ *For more details on Flow Measures, see Application Flow qualification.*
4. Select the flow measures that interests you and click **Add**.
   The application flow is displayed with its content in the diagram.

Application Flow qualification

Application flow categories provide a way to define parameters of the application flows that are used described in the scenario of flows.

To qualify an application flow from *flow measures*:

1. Open the **Qualification** property page of the application flow that interests you.
2. In the **Flow Qualification** section, click the **Connect** button.
   A selection dialog box opens displaying the tree of existing Flow Categories.

   > ☛ *To access the list of Application Flow Categories: using the **Administration** navigation menu, select **Categorization** and unfold the **Measure schemes Categorization**. The list of Application Flow Categories appears.*
   >
   > ☛ *For more details on the Application Flow category concept, see Defining Data Categories.*

Associating a service interface used to an application flow

> *A service interface use is associated to a service interface. It enables representation of complex exchanges.*
>
> ☛ *For more details on service interfaces, see Describing a service interface.*

Associating a service interface to an application flow enables to establish a correspondence between a Scenario of Flows diagram and a Structure diagram of an application or an application system.

It identifies the service interface/operation called (if the application flow type is **Call/Request**) or the supplying service interface/operation (if the application flow type is **Result/Provision**).

To associate a *service interface* to an *application flow*:

1. Open the **Service Interface Used** page of the application flow that interests you.
2. Click **New**.
   A dialog box opens proposing a list of *service interfaces used* whose content is a request point or a result.

Creating an application flow channel

> *An application flow channel is used to graphically group a number of application flows into a single flow.*

To create an application flow channel, you must first create the channel and then link the application flows that it groups.

To create an *application flow channel*:

1. In the objects toolbar of the scenario of application flows, click **Application Flow Channel**.
2. Click the first object in communication and, holding the mouse button pressed, draw a link to the other object.
   The application flow channel appears in the diagram.

To connect the application flows to the *application flow channel*:

1. Open the **Characteristics**properties page of the application flow channel.
2. In the **Grouped Flow** section, click **Connect**.
   A selection dialog box opens and presents the list of the application flows of the scenario of application system flows.
3. Select the flows that you want to group and click **OK**.
   The content of the selected flows is displayed in the **Grouped Flow** list.
4. Click the **Refresh Channels** button.
   The application flows grouped in the channel disappears and the corresponding content is displayed around the channel.

> ☛ *If you remove the channel, only the application flows created from the **Grouped Flows** are removed. The connected application flows are displayed if you click the **Refresh Channels** button.*

Creating a System Triggering Event

The creation process for a *Creating a System Triggering Event* and a *Creating a System Triggered Event* is the same.

To create a *System Triggering Event*:

1. In the diagram insert toolbar, click the **System Triggering Event** button.
2. Position the object at the edge of the frame of the described object.
   A creation dialog box opens.
3. Click the arrow at the right of the **Referenced Content** field and select the content that interests you.
4. Click **Add**.
   The *System Triggering Event* appears in the diagram.

Any application flow whose origin is the *System Triggering Event* is connected to the same content.

To create an application flow from a *System Triggering Event*:

1. In the diagram insert toolbar, click **Event Participation**.
2. Click the event and, holding the mouse button pressed, draw a link to the object receiving the flow.
   The application flow is displayed with its content associated to the event.

Reinitializing components in a scenario of flows

If you insert in a scenario of flows diagram a component that is already described by a scenario of flows, you can note that a new section is created in the **Characteristics** property page of the component you have added. This section allows you to specify which scenario of flows of the component corresponds to the context of the current application system scenario of flow.

In the component scenario of flows diagram, the **Reinitialize components** button allows you to insert components coming from the upper level scenario of flow.

Adding an application data store to the scenario of application flows

> 🕮 *An application data store materializes the usage of data in the context of a software component (for instance an application). An application data store provides a mechanism to retrieve or update information stored outside of the current software component.*

A data store can be local or external to the application.

To add, for example, a local application data store to an scenario of application flows

1. In the scenario objects toolbar, click **Local Application Data Store**.
2. Click in the described application frame.
   An addition window prompts you to choose the **Object Type** that represents the physical structure that will concretely support the application data store.

   > ☛ *For more information on managing data stores, see Managing Data.*
3. Depending on the **Object type**, select then the object that interests you.
4. Click **OK**.
   The local application data store appears in the diagram with the name of the physical data domain selected.

Creating an application data channel

The applications, the application systems and the microservices can have read or write access to a local or external application data store.

To create an application data channel that represents a reading access:

1. In the diagram objects toolbar, click **Application Data Channel**.
2. Draw a link between the application data store and the object that reads the data.
   An application data channel appears in the Diagram.

> ☛ *To create a link with write access, you must draw a link between the object that reads and the application data store.*

Describing a Scenario of Application System Environment Flows

A scenario of application system environment represents the flows exchanged between the components of the application system environment.

> ☛ *For more details on use of a scenario of flows, see Using a Scenario of Application Flows Diagram*

The elements of a scenario of application system environment are:

- the main *application system* principal described by the environment.

  > 🕮 *An application system is an assembly of other application systems, applications and end users interacting with application components to implement one or several functions.*

- *partner application systems* that represent the other application system with which the main application system described by the environment interacts.

  > 🕮 *A partner application system is an application system external to the environment of the described application service. The partner application system can be a service supplier or a service consumer with respect to application system users.*

- *End User Participants* that represent the categories of users of application system provided by the environment.
- The categories of users of services provided by the environment are represented either by an *Org-Unit* or by a *Position Type*.

  > 🕮 *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 enterprise structure such as a management, department, or job function. It is defined at a level depending on the degree of detail to be provided on the organization (see org-unit type). Example: financial management, sales management, marketing department, account manager. An external*

org-unit is an external entity that exchanges flows with the enterprise. Example: customer, supplier, government office.

📖 *A position type represents a status assigned to an individual or a group of individuals with the aim of defining an organization or a hierarchy.*

- *Service interactions* between components

  📖 *A Service Interaction represents an interaction for service purpose between entities in a specific context inside or outside a company. These entities can be enterprise org-units, applications, activities or processes, as well as external org-units. The content of this interaction is described in a service interface.*

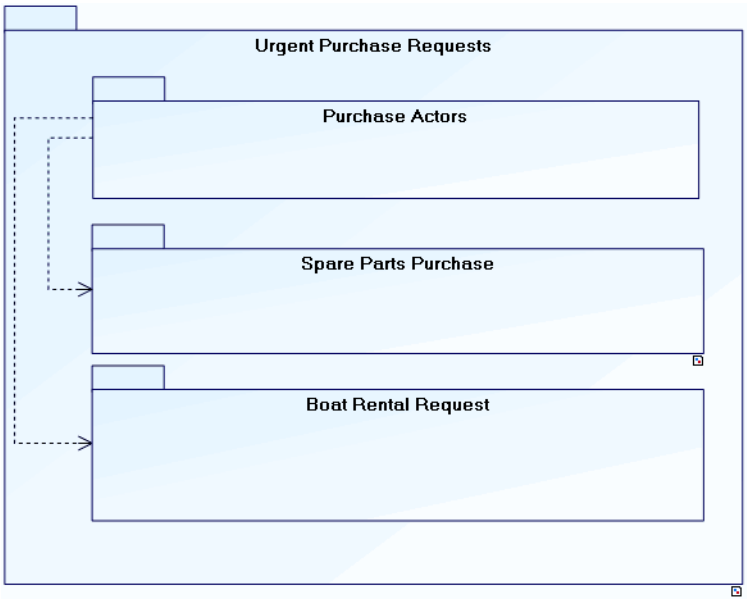
RESOURCES - INFORMATION (LOGICAL)

The **Information (Logical)** aspect of the **Resources** view helps UAF Architect to manage the logical models of the system data.

Packages

A *Package* is used to represent the static structure of a system, particularly the types of objects handled in the system, their internal structure, and the relationships between them.

The package allows you to classify elements referenced in a project. You can create sub-packages in a package to classify objects in finer detail, for example actors of a project.



Urgent purchase requests are provided to process purchase of spare parts and boat rental requests. In both of these cases, users are actors of the purchasing domain.

Accessing a Package with HOPEX UAF

To access Packages:

1. From the navigation menu, select **Resources > Hierarchy**.
2. Expand the **Package** folder.
   A default Package is automatically created when an Enterprise is created.

Description of a Package diagrams with HOPEX UAF

Each Package diagram is associated to a *Data Domain*. When a package diagram is created a *Data Domain* is alos automatically created.

> *A Data Domain is a sub-set of enterprise data designed to be used together in the context of a business operation. For example, the "Sales" data domain contains at least the following data: Clients, Orders, Products. Each entity of a data domain has CRUD characteristics. The data domains define the functional data frontiers used both for allocating data to applications (see Data stores) and for the governance or data in data management.p*
>
> ☛ *For further details about Data Domains, see Data Domains.*

A Package can be represented by the following diagrams:

- A *Data Domain Entity diagram* which is a graphical representation of the classes used in the context of the package.

  ☛ *For further details about Classes, see Classes.*

- A *Data Domain Structure diagram* describing the sub-domains of a data domain.

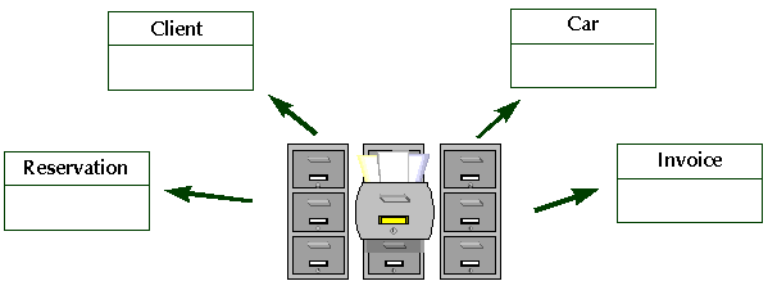
Classes

Definition: Class

A *class* is described by a list of attributes and operations.

A class is linked to other classes via *associations*. The set of classes and associations forms the core of a class diagram.

We can illustrate the class concept by comparing classes to index cards filed in drawers.



Classes can be technical objects used for programming.

```
Examples: dialog box, rectangle, string, table, etc.
```

Classes can represent technical objects used in industry.

```
Examples: Alarm, Sensor, Zone
```

Classes can also represent business objects:

Examples: customer, order, product, person, company, etc.

A class can also express a process, such as "Confirm client request", or implement a business rule, such as "Consistency in cost accounts".

Accessing Classes with HOPEX UAF

To access Classes:

1. From the navigation menu, select **Resources > Information (Logical)**.
2. Click the **Classes** tab.
   The list of Classes of the Enterprise is displayed.

Class Properties

For information regarding Classes in HOPEX, see "Classes" chapter in HOPEX IT Architecture guide

---

Data Domains

A Data Domain is a sub-set of enterprise data designed to be used together in the context of a business operation. For example, the "Sales" data domain contains at least the following data: Clients, Orders, Products. Each entity of a data domain has CRUD characteristics. The data domains define the functional data frontiers used both for allocating data to applications (see Data stores) and for the governance or data in data management.p

Data domains are used to define a logical data structure made up of classes and class views.

The application data domain is used to describe the data stores of software (Application system, Application, Application service or Micro Service).

☛ *For more details on how to use data domains in an application architecture, see the documentation of **HOPEX IT Architecture** > "Modeling technical and functional architectures".*

Accessing Data Domains with HOPEX UAF

To access Data Domains:

1. From the navigation menu, select **Resources > Information (Logical)**.
2. Click the **Data Domains** tab.
   The list of Data Domains of the Enterprise is displayed.

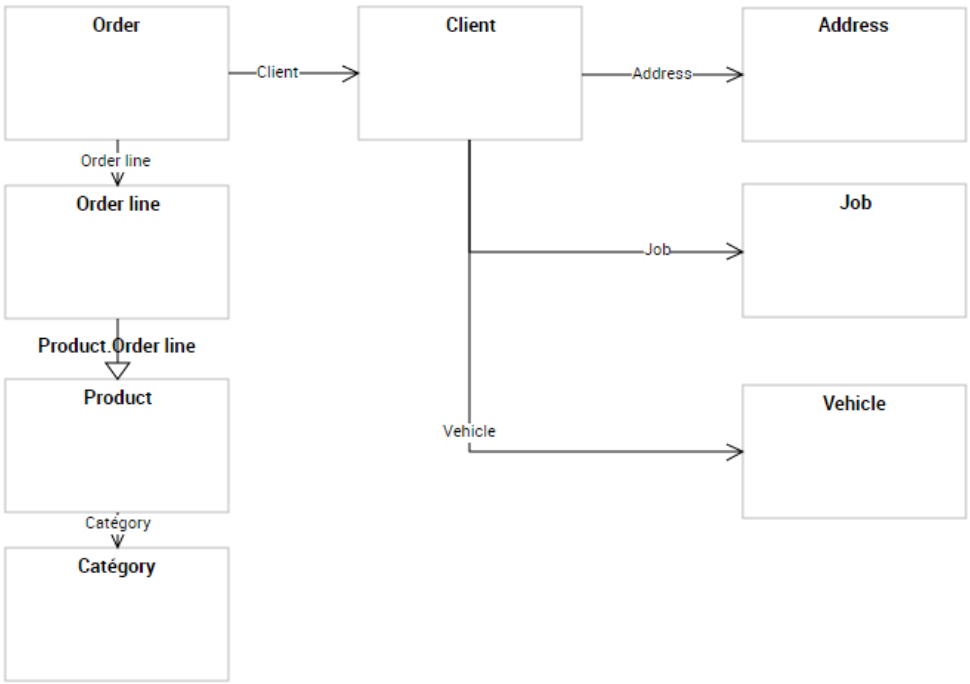
The Data Domain Structure Diagram

A data domain structure diagram defines classes and their relationships in a Whole/Part formalism in connection with the subject of the data domain described.

You can connect one or more data domain diagrams to a data domain, according to what you want to describe.

Example of diagram

The following data domain diagram represents a data structure relating to Orders; it describes classes and their relationships in a Whole/Part formalism.



Data Domains Properties

For information regarding Classes in HOPEX, see "The Data Domain Diagram" chapter in HOPEX Data Governance guide

RESOURCES - INFORMATION (PHYSICAL)

The **Information (Physical)** aspect of the **Resources** view helps UAF Architect to manage the physical models of the system data.

Databases

A database is the physical object that enables storage and organization of logical data for use by programs corresponding to distinct applications, to facilitate the independent evolution of the data and the application programs.

HOPEX Data Governance

Both are owned by a package and can reference objects held in other packages.

☛ *A corresponding physical structure can be defined via a physical data domain. It is made up of tables and table views. See Database.*

Accessing a Database with HOPEX UAF

To access Databases:

1. From the navigation menu, select **Resources > Hierarchy**.
2. Expand the **Database** folder.
   A default Database is automatically created when an Enterprise is created.

Description of a Database diagrams with HOPEX UAF

Each Database diagram is associated to a *Relational Schema*. When a Database diagram is created a *Data Domain* is alos automatically created.

🕮 *A Database is the physical object that enables storage and organization of logical data for use by programs corresponding to distinct applications, to facilitate the independent evolution of the data and the application programs.*

☛ *For further details about Relational Schemes, see Relational Schemas.*

A Database can be represented by the following diagrams:

- A *Relational Schema Structure diagram* describing the sub-schemes of a Relational Schema.
- A *Relational Schema Table diagram* which is a graphical representation of the tables used in the represented Relational Schema and their relationships.

☛ *For further details about Tables, see Tables.*

Relational Schemas

A Relational schema represents a set schema stored in a database management system and used in the technical architecture of an application.

Accessing Relational Schemas with HOPEX UAF

To access relational schemas:

1. From the navigation menu, select **Resources > Taxonomy - Structure (Physiscal)**.
2. Click the **Relational Schemas** tab.
   The list of relational schemas is displayed.

Relational Schema Diagram

A relational schema is made up of tables and/or physical views and can be described in two types of diagram:

- the table diagram which is used to display a set of tables and their relationships (FK).
- the structure diagram that is used to break down a relation a schema into sub-domains.

You can connect one or more diagrams to a relational schema according to what you want to describe.

For information regarding Relational Schemes in HOPEX, see "Relational Schemes" chapter in HOPEX Data Governance guide

Tables

Tables are components of Relational Schemas, see Relational Schemas.

Accessing Tables with HOPEX UAF

To access Classes:

1. From the navigation menu, select **Resources > Information (Physical)**.
2. Click the **Tables** tab.
   The list of Tables of the Enterprise is displayed.

Table Properties

For information regarding Classes in HOPEX, see "Database Components" chapter in HOPEX Data Governance guide

RESOURCES - CONSTRAINTS

The **Constraints** aspect of the **Resources** view is dedicated to the constraints imposed to Assets (Ex: *Applications*. *IT Service*. etc.).

List of Resources constraints and reports

Three types of **Constraints** may be imposed to assets:

- Constraints, specified in the asset **Constraints** page,

  ☛ *For more details on Constraints, see Constraints in the Resource view*

- Mesurable properties, specified in the **Measurable Property** and **Qualifying Values** properties of the asset.

  ☛ *For more details on Measurable Properties, see Describing a Measurable Property*.

- Directing Regulations, specified in the asset **Regulations** properties.
  - *For more details on Directing Regulation structure, see Security - Taxonomy*.

Three types of contraints reports are available:

- The Constraint Reports,
- The Measure Reports,
- The Politicy Reports.

Constraints in the Resource view

To access the list of **Resource Constraints** dedicated to assets:

1. From the navigation menu, select **Resources > Constraints**.
2. Click the **Constraints** tab to access the list of resource constraints.

To create a *Resource Constraint*:

1. From the **Resources** navigation menu, select **Constraints**.
   The list of the *Resource Constraints* appears.
2. Click **New**.
   The new *Resource Constraint* appears in the list.

RESOURCES - ROADMAP

The **Roadmap** aspect of the **Resources** view helps UAF Architect to create resource asset catalogs and to define the life cycle of the corresponding resources.

Accessing a Resource Asset Catalog

To access a Resource Asset Catalog:

1. From the navigation menu, select **Resources > RoadMap**.
2. Using the arrow at the right of the **Resource Asset Catalog** field, select the Catalog that interests you.
   The tree of catalog components appears below.

Resource Asset Catalogs Creation

To create a Resource Asset Catalog:

1. From the navigation menu, select **Resources > RoadMap**.
2. Click the arrow at the right of the **Resource Asset Catalog** field, select **New**.
   A Resource Asset Catalog is created, the corresponding tree appears below.

Creating a Resource Catalogued Asset

To create a Resource Catalogued Asset:

1. Select Resource Asset Catalog that interests you.
2. In the tree of catalog components, click **Plus** button.
   An **Creation of Object Life** window opens.
3. Using the arrow at the right of the **Catalogued Asset** field, click **Create** button.
4. Select the **Object Type** and click **Next**.
5. Create an object or reuse existing ones and click **Add**.
6. Enter the name of the object and click **OK**.
   The Catalogued Asset is created.
7. Specify the **Lifecycle to follow**.
8. Specify the **Begin Date** and the **End Date**.
9. Click **OK**.

Resource Asset Catalog Report

To access a Resource Asset Catalog Report:

- Open the **Gantt Chart** property of the Resource Asset Catalog that interests you.

RESOURCES - TRACEABILITY

The **Traceability** aspect of the **Resources** view provides access to several types of matrixes and reports about:

- *Operational Activities* in rows

  📖 *An Operational Activity is an end-to-end collection of Activity Actions that creates an outcome for a customer, who may be the ultimate customer or an internal end-user of the Operational Activity.*

- *Functions* in column

  📖 *A Personnel Function is a set of operations performed by org-units within a company or organization, to produce a result. It is depicted as a sequence of operations, controlled by events and conditions. In the BPMN notation, the Personnel Function represents a sub-Personnel Function from the organizational point of view.*

  📖 *A Resource Function is the executable representation of a process. the events of the workflow, the tasks to be carried out during the processing, the algorithmic elements used to specify the way in which the tasks follow each other, the information flows exchanged with the participants.*

For more details on Traceability matrixes and reports, see Using Traceability Aspect.

For more details on *Resource Functions*, see *Using Resource Functions*.

Systems of Resources to Resource Functions

The matrix shows the *Resource Functions* executed by *Systems of Resources*. You can create a link or delete an existing one using this matrix.

☛ *For more information, see Describing Hardware (Macro).*

- *Systems of Resources* in rows

  📖 *A Capacity Configuration is a combination of physical assets and organization configured to provide a capability.*

  📖 *A Facility is a model of site of interest for the enterprise. Examples: Data Center, Factory or Outlet*

  📖 *A Resource Configuration is a set of physical and human resources configured to provide a business capability.*

- *Resource Functions* in column

  📖 *A Resource Function is the executable representation of a process. the events of the workflow, the tasks to be carried out during the processing, the algorithmic elements used to specify the way in which the tasks follow each other, the information flows exchanged with the participants.*

Resource Functions to Operational Activities

The matrix shows the resource functions (in rows) that implement the Operational activities (in line).You can create a link or delete an existing one using this matrix.

☛ *For more information, see Using Resource Functions.*

- *Resource Functions* in rows

  🕮 *A Resource Function is the executable representation of a process. the events of the workflow, the tasks to be carried out during the processing, the algorithmic elements used to specify the way in which the tasks follow each other, the information flows exchanged with the participants.*

- *Operational Activities* in columns

  🕮 *An Operational Activity is an end-to-end collection of Activity Actions that creates an outcome for a customer, who may be the ultimate customer or an internal end-user of the Operational Activity.*

SECURITY VIEWPOINT

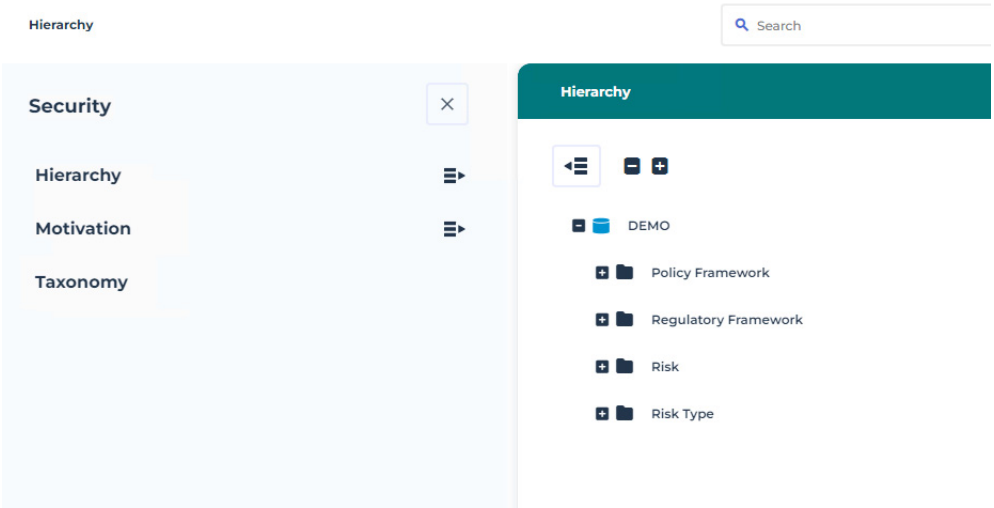
The **Security** Viewpoint illustrates the security assets, security constraints, security controls, families, and measures required to address specific security constraints.

This chapter comprises the following sections:

- ✓ Security - Hierarchy
- ✓ Security - Motivation
- ✓ Security - Taxonomy
- ✓ Security - Traceability

SECURITY - HIERARCHY

The **Hierachy** aspect of the **Security** view helps UAF Architect to access the main security elements.



To get more information about the use of folders in **HOPEX** trees, see **HOPEX**, "Handling Trees" chapter of the **HOPEX Common Features** guide.

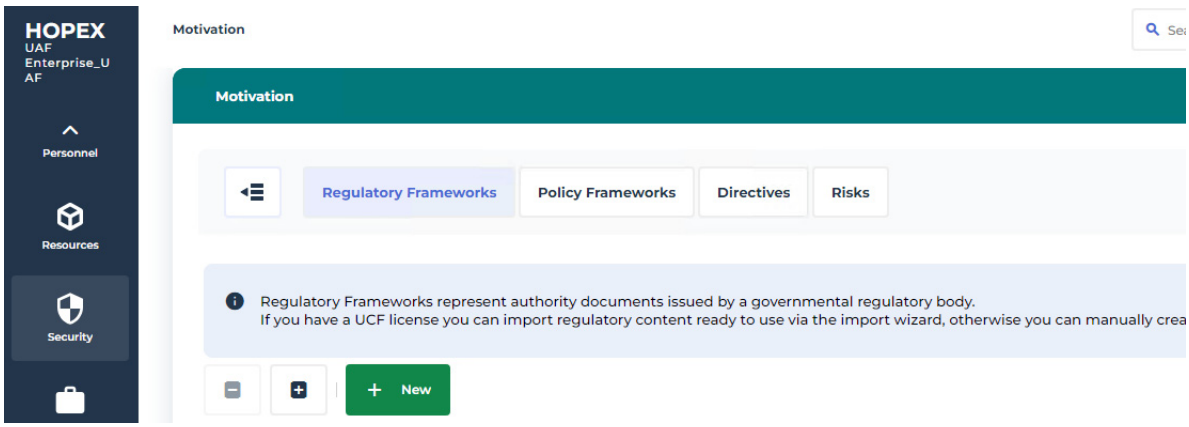
The hierarchy navigation enables to navigate on the main concepts of the view:

- Policy Frameworks, see Defining Policy Frameworks,
- Regulatory Frameworks ,see Defining Regulatory Frameworks,
- Risks and Risk Types, see Defining Risks.

SECURITY - MOTIVATION

The **Motivation** aspect of the **Security** view helps UAF Architect to identify security controls to mitigate against the security risks.

In **HOPEX UAF**, any risk should be associated to a Security Control. A Security Control is implemented as Regulation Article and Security Control Family is implemented as Regulation Section.



Defining Regulatory Frameworks

A regulatory framework is an authority document falling under any of following categories: regulations (rules of law that, if not followed, can result in penalties), guidelines, standards, best practices.

A regulatory framework defines the content of a regulation. It includes a set of sections, and under each section, sub-sections and articles.

You can define the objects in your organization that are constrained by the regulation as well as the business rules that ensure the implementation of the regulation within the company, see Classifying assets in Regulatory Frameworks.

Creating a Regulatory Framework

To create a regulatory framework in **HOPEX UAF**:

1. From the navigation menu, select **Security > Motivation**.
2. Click the **Regulation Frameworks** tab.
3. Click **New** button.

4. In the dialog box that appears, specify:
   - the name of the regulatory framework
   - its description
5. Click **OK**.
   The regulatory framework appears in the list.

Adding a Regulatory Section to the Regulatory Framework

To add a section to the regulatory framework:

1. Select the regulatory framework that interests you.
2. Click **New > Regulation Section**.
   The creation window for a section opens.
3. Specify:
   - (optional) the section code
   - the title
   - the description, which corresponds to the text of the section.
4. Click **OK**.
   The section appears in the navigation tree.

☛ *For more details, see Adding a Section to the Regulatory Framework.*

Adding a Regulatory Article to the Regulatory Section

To add a section to the regulatory section:

1. Select the regulatory section that interests you.
2. Click **New > Regulation Article**.
   The creation window for a section opens.
3. Specify:
   - the article code
   - the title
   - the description, which corresponds to the text of the section.
4. Click **OK**.
   The article appears in the navigation tree under the section.

☛ *For more details, see Adding an Article to a Section.*

Then, you can add a Definition to an Article.

☛ *For more details, see Adding a Definition to an Article.*

Importing the Module of Regulatory Frameworks

The libraries of regulations to be imported are delivered in the form of modules that can be downloaded into the HOPEX HAS console.

To download the module of a regulation:

- From your HOPEX version, open the HAS Administration Console and download the module. See Importing a Module into HOPEX.

Defining Policy Frameworks

A policy framework represents internal documents issued by the Organization, such as code of conducts, standard security measures and similar.

In **HOPEX UAF**, a *Policy framework* defines a set of business policies.

It is composed of sections and sub-sections that represent categories of business policies. Under these sections you can define the business policies, the assets constrained by the policies in question and their implementation.

Creating a Policy Framework

To create a policy framework in **HOPEX UAF**:

1. From the navigation menu, select **Security > Motivation**.
2. Click the **Policy Frameworks** tab.
3. Click **New** button.
4. In the dialog box that appears, specify:
   - the name of the regulatory framework
   - its description
5. Click **OK**.
   The policy framework appears in the list.

Adding a Regulatory Section to the Policy Framework

To add a section to the policy framework:

1. Select the policy framework that interests you.
2. Click **New > Policy Section**.
   The creation window for a section opens.
3. Specify:
   - (optional) the section code
   - the title
   - the description, which corresponds to the text of the section.
4. Click **OK**.
   The section appears in the navigation tree.

☛ *For more details, see Adding a Section to the Policy Framework.*

Adding a Business Policy to the Policy Framework

A business policy is a directive whose purpose is to govern or guide the company. A business policy serves as the basis for defining business rules and governing corporate processes. A business policy is always under the control of the company. It allows to control, guide and formalize the strategies and tactics of the company.

To add a business policy to a policy framework:

1. Select the policy section that interests you.
2. Click **New > Business Policy**.
   The creation window for a section opens.

3. Specify:
   - the name of the section
   - the title
   - the comment

   ☛ *For more details, see Adding a Business Policy.*

Defining Control Directives

📖 *A control directive is an interpretation of the law and contributes to the enforcement of any regulation article your organization has to comply with.*

☛ *For more details, see Adding a Definition to an Article.*

Implementation controls may be associated with the control directives. See: Defining Data Quality Controls.

Accessing the list of control directives

To access control directives from a tree:

1. From the navigation menu, select **Security > Motivation**.
2. Click the **Directives** tab.
   All control directives are displayed in a tree that enables to view their inter-relations.

   ☛ *For more details, see Accessing the list of control directives.*

Some columns indicate:

- whether the control directive constrains the organization
- the number of implementing controls associated
- the enforcement levels

  ☛ *For more details, see Enforcement level of control directives.*

Associating a control directive with an article of regulatory framework

From an article, you can create a control directive or connect an existing control directive.

To create a control directive from an article:

1. In the framework regulatory tree, select the article.
   Buttons appear to its right.
2. Click the **New** > **Control Directive** button.
   The window for creating a control directive appears.

   ☛ *You can also connect an existing control directive to an article:*

Viewing articles associated with a control directive

To view articles associated with a control directive:

1. Open the properties of the control directive.
2. Click the **Characteristics** page.
3. Expand the **Regulation Articles** section.

Associating a control directive with an article of regulatory framework

From an article, you can create a control directive or connect an existing control directive.

To create a control directive from an article:

1. In the framework regulatory tree, select the article.
   Buttons appear to its right.
2. Click the **New** > **Control Directive** button.
   The window for creating a control directive appears.

> ☛ *In the same way, you can also connect an existing control directive to an article.*

Associating a risk to a control directive

A risk must be associated to a control and controls are associated to control directives.

Defining Risks

> 🕮 *A risk is a hazard of greater or lesser probability to which an organization is exposed.*

To control risks, it is necessary to identify and qualify the risks encountered in the execution of a process.

When risks have been analyzed and assessed, management determines how each of these risks should be treated.

> 🕮 *A risk type defines a risk typology standardized within the context of an organization.*

Accessing the list of risks

To access the list of risks:

1. From the navigation menu, select **Security > Motivation**.
2. Click the **Risks** tab.
   All risks are displayed with the associated **Risk Type**.

> ☛ *For more details, see Risk types.*
>
> 🕮 *A risk type defines a risk typology standardized within the context of an organization.*

Risk characteristics

The **Characteristics** property page provides:

- the risk **Name** and identification **Code**
- the risk **Owner**
- the risk **Identification Mode**
  The risk could have been identified from:
  - an "incident database"
  - a "workshop"
  - a "survey"
  - an "audit"
- the risk **Description**

> ☛ *The risk **Status** cannot be modified since it is managed by the workflow associated with the risk.*

The **Responsibilities** section of the **Characteristics** page of a risk is used to define the different persons responsible for risk management. For more details, see Responsibilities (RACI).

The **Scope** section of the **Characteristics** page of a risk is dedicated to the contextualisation of the risk represented by:

- **Processes** for *Personnel Functions* exposed to the risk. See Personnel Functions.

  > 🕮 *A Personnel Function is a set of operations performed by org-units within a company or organization, to produce a result. It is depicted as a sequence of operations, controlled by events and conditions. In the BPMN notation, the Personnel Function represents a sub-Personnel Function from the organizational point of view.*
- **Operations** for *Personnel Function actions* exposed to the risk. See Personnel Functions.

  > 🕮 *A Personnel Function Action is an elementary step in process executed by an org-unit. It cannot be broken down. An operation can be industrial (manufacturing a component), logistical (receiving a delivery), or can involve information processing (entering an order).*
- **Entities** for *Organizations* concerned by the risk. See: Organizations.

  > 🕮 *An organization represents a person or a group of persons that intervenes in the enterprise Personnel Functions or information system. An organization can be internal or external to the enterprise. An internal organization is an organizational element of the enterprise structure such as a department, a service, or a workstation. An internal organization 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 organization is an organization that exchanges flows with the enterprise. Example: Customer, Supplier, Government Office.*
- **Applications**. See Describing an Application with HOPEX UAF.

  > 🕮 *An Application is a software component that can be deployed and provides users with a set of functionalities.*
- **Technologies**: See Describing Software Technologies.

  > 🕮 *A Software Technology is a basic component necessary for operation of business applications. Software technologies include all basic software such as: application server, electronic mail server, software components for presentation, data entry, storage, business information sharing, operating systems, middleware, navigators, etc.*

The **Analysis** section of the **Characteristics** page of a risk presents the results of the risk analysis that takes into account: the risk causes and the positive or negative risk consequences.

With **HOPEX UAF**, the risk analysis phase results are:

- **Risk Types**: for more details, see Risk types.

  A risk type defines a risk typology standardized within the context of an organization.

- **Related Risks.**

Risk types

A *Risk Type* enables risk characterization. For example, a risk type can be regulatory, legal, technical, etc.

A risk type defines a risk typology standardized within the context of an organization.

To access the list of risks:

1. From the navigation menu, select **Security > Hierarchy**.
2. Expend the **Risk Type** folder.
   The tree of Risk Types is dispalyed.

To create your own risk type:

1. From the navigation menu, select **Security > Hierarchy**.
2. Expend the **Risk Type** folder.
3. Click **New > Risk Type** button.
4. Enter the name of the risk type and click **OK**.
   The new risk type appears in the navigator menu tree.

   ☛ *Similarly, you can create a sub-risk type from a risk type.*

SECURITY - TAXONOMY

The **Taxonomy** aspect of the **Security** view helps UAF Architect to defines the hierarchy of security assets and asset owners that are available to implement security, security constraints (policy, guidance, laws, and regulations) and details where they are located (regulation framework and the policy framework).

General description of the Taxonomy aspect interface

This interface is divided into to parts:

- The rigth side is dedicated to frameworks (regulation framework and policy framework).
- The left side is dedicated to assets. The hierarchy of Enterprise elements is displayed. Using the **Search** box, you can easely access to a specific element.

Classifying assets in Regulatory Frameworks

A regulatory framework is an authority document falling under any of following categories: regulations (rules of law that, if not followed, can result in penalties), guidelines, standards, best practices.

For more details on regulatory Frameworks, see Defining Regulatory Frameworks.

To specify that a **Microservice** must respect a specific **Regulation Article**:

1. From the navigation menu, select **Security > Taxonomy**.
2. Select the **Regulation Framework** tab.

3. On the rigth side of the edition area, expand the framework folders.
4. On the left side of the edition area, select the microservice to classify.
5. Drag and drop the microservice to the selected regulation article.
   The microservice appears in the **Subjected Elements** folder.

Classifying assets in Policy Frameworks

📖 *A policy framework represents internal documents issued by the Organization, such as code of conducts, standard security measures and similar.*

☛ *For more details on Policy Frameworks, see Defining Policy Frameworks.*

To specify that an **Application** must respect a specific **Policy Frameworks** rules:

1. From the navigation menu, select **Security > Taxonomy**.
2. Select the **Policy Framework** tab.
3. On the rigth side of the edition area, expand the framework tree that interests you.
4. On the left side of the edition area, select the asset to classify.
5. Drag and drop the asset to the selected security element.
   The asset appears in the **Subjected Elements** folder.

SECURITY - TRACEABILITY

The **Traceability** aspect of the **Security** view provides access to several matrixes presenting the Assets impacted by risks and policies.

Risks to Assets matrix

The matrix shows the Assets defined in the scope of a list of risks.

> ☛ *For more information on the scope of a risk, see Risk characteristics.*

- Risks in rows

> 🕮 *A risk is a hazard of greater or lesser probability to which an organization is exposed.*

- Assets in columns

To create a Risks to Assets matrix:

1. From the navigation menu, select **Security > Traceability**.
2. Select the **Risks to Assets** tab.
   The list of existing matrix is displayed.
3. Click the **New** button.
   The new matrix is created.
4. Click the matrix to open it.
5. Use the **Add row** buton to add risks in the matrix.
6. Use the **Add column** butons to add assets in the matrix.
   The assets in a risk scope are represented by check marks.
7. Click the box between a risk and an asset to add or to remove the asset from the scope of the risk.

Policies to Assets matrix

The matrix shows the Assets defined as Subjected Elements of Policy Framework.

> ☛ *For more information on the Subjected Elements of Policy Framework, see Classifying assets in Policy Frameworks.*

- Policy Frameworks in rows
- Assets in columns

> 🕮 *A policy framework represents internal documents issued by the Organization, such as code of conducts, standard security measures and similar.*

To create a Policies to Assets matrix:

1. From the navigation menu, select **Security > Traceability**.
2. Select the **Policies to Assets** tab.
   The list of existing matrix is displayed.

3. Click the **New** button.
   The new matrix is created.
4. Click the matrix to open it.
5. Use the **Add row** buton to add Policy frameworks in the matrix.
6. Use the **Add column** butons to add assets in the matrix.
   The assets defined as **Subjected Elements** of a Policy framework are represented by check marks.
7. Click the box between a policy and an asset to add or to remove the asset from the Subjected Elements of a Policy framework.

PROJECTS VIEWPOINT

HOPEX UAF offers a set of features to:

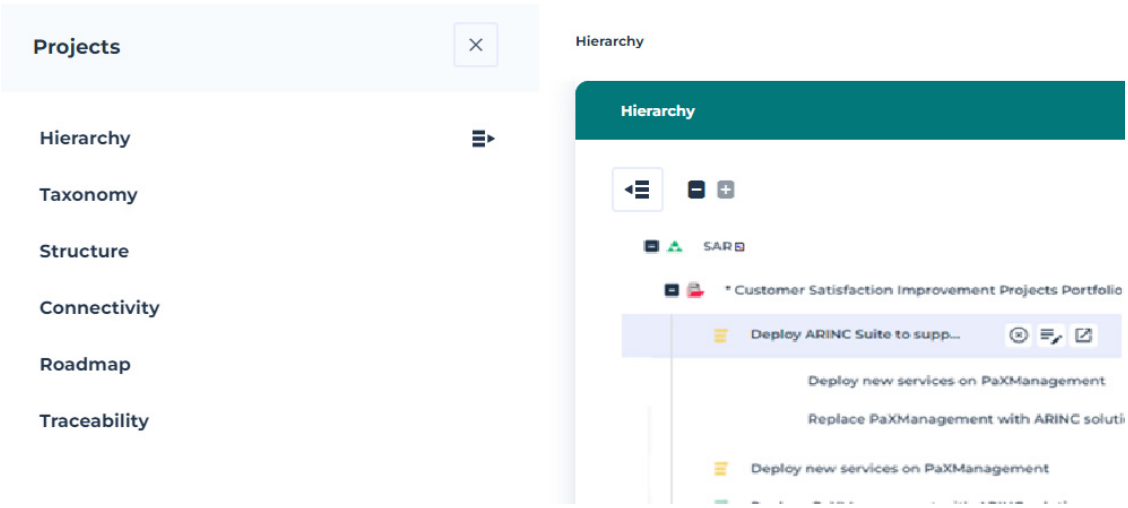
- Submit and assess the project demands and candidate projects.
- Validate the candidate projects: the project demand goes through a validation process that results in a project creation.
- Select and define the project priority: a limited list of projects is drawn up according to selection criteria (strategic, financial, etc.)
- Analyze and arbitrate the projects.
- Follow project progress.

This viewpoint is dedicated to Projects management. It describes projects and project milestones, how those projects deliver capabilities, the organizations contributing to the projects, and dependencies between projects.

This chapter comprises the following sections:

PROJECTS - HIERARCHY

The **Hierachy** aspect of the **Projects** view helps UAF Architect to access the main elements of the transformation projects.



To get more information about the use of folders in **HOPEX** trees, see **HOPEX**, "Handling Trees" chapter of the **HOPEX Common Features** guide.

The hierarchy navigation enables to navigate on the main concepts of the view:

- Project Portofolio, see Describing a Project,
- Project,see Using Projects Portolios,
- Project Milestone,Describing a Project.

PROJECTS - TAXONOMY

The **Taxonomy** aspect of the **Projects** view helps UAF Architect to describe a structured list of projects.

Describing a Project

A project is a temporary endeavor undertaken by a specific team, to create a unique product, service or result. Its serves a purpose which can be expressed in terms of capability that is acquired (new), extended (improvement) or decommissioned (rationalization).

Accessing projects

To access Projects:

1. From the **Project** navigation menu, select **Taxonomy**.
2. Click the **Projects** tab.
   The list of the Enterprise Projects is displayed.

Creating a Project Demand

The demand creators can document the **project charter** as well as the **business case**. They can in particular define the scope of the project in terms of deliverables and timelines.

They must also have created a project domain. See Project Domains.

Quick Access contains a shortcut **Create a Project** to facilitate the work in **HOPEX UAF**. For more details, see Quick Access Action.

To create a project demand from **Quick Access**:

1. C lick **Create a Project** tile.
   The project creation window appears.
2. Select the "Demand" project type and click **Next**.
3. Specify:
   - The project name
   - the owner project domain
   - the project code (optional)
   - the planned start date
   - the planned end date
4. Click **OK**.

Defining the Project Charter

The **Project Charter** properties page of the project provides access to:

- The **identification**:
  - project name
  - project owner domain
  - project code (optional)
  - project manager
  - state (life cycle status) Defined automatically.
  - status (workflow step). Defined automatically.
  - description (comment)
- The project **category or categories**.
- The **initiating ideas**: ideas that have inspired the project.

Defining the Business Case of a Project

The **Business Case** properties page of the project provides access to:

- Transformation Purpose,
- Project deliverables,
- Project dependencies,
- Project costs,
- Project benefits,
- Project risks.

Transformation Purpose

A project has an objective with respect to the capabilities of the enterprise (as defined in a capability map); it can:

- deliver the means to acquire a new capability (innovation)
- extend the coverage of a capability already held (improvement)
- restrict or abandon the coverage of an existing capability (rationalization).

To add a transformation objective to the project:

1. In the **Transformation Purpose** section, click **New**.
   The creation dialog box for a transformation objective opens.
2. Specify:
   - its name
   - the transformation type (Innovation, Improvement, Rationalization)
   - the capability transformed
3. Click **OK**.

> ☛ *You can also create a Transformation Purpose using the Projects to Capabilities matrix. For more details, see Projects - Traceability.*

To specify that a transformation purpose of a project is connected to an *exhibited capability*:

> 🕮 *An exhibited Capability is a Capability that is exhibited by an Enterprise Stage with quantified measure (KPI) and potential geopolitical scope (Site) for a defined market segment (Operational Partner).*

1. In the **Transformation Purpose** section of the project, select the transformation purpose that interests you.
2. In the **Exhibited Capability** column, select the capability.

Project dependencies

A project can depend on other projects:

- In a "positive" sense: a project can have another project as a prerequisite, of which one of the deliverables is necessary to build a deliverable of the dependent project (this is the equivalent of an AND logic: both projects must be conducted jointly to reach the final result).
- In a "negative" sense: two projects can be concurrent and mutually exclusive (this is the equivalent of the OR logic: only one of the projects must be managed, not both).

To associate a dependency with the project:

1. In the **Project Dependencies** section, click **New**.
2. Specify:
   - The name of the dependency
   - The project required
   - The type of dependency: "Exclusive" or "Prerequisite".
3. Click **OK**.

> ☛ *You can also create Project Dependencies using the Connectivity Matrix. For more details, see Connectivity Matrix.*

Project benefits

You can specify:

- the **Qualitative Benefits**: to be entered as a comment.
- the **Financial Value** of the project: in currency = project NPV (net present value), calculated outside the tool according to the standards of the enterprise.
- the **Return on Investment**: calculated attribute, as a % (Financial value - Budget) / budget
- the **Forecast Return on Investment**: calculated attribute, as a % (Financial value - Estimated total cost) / Estimated total cost
- the **Actual Return on Investment**: calculated attribute, as a % (Financial value - Real total cost) / Real total cost

Project risks

With **HOPEX UAF**, you can identify the risks linked to a project. Each risk is associated with a single project.

To create a project risk:

1. Expand the **Risk** section.
2. Click **New**.
   The risk creation dialog box appears.
3. Enter the name of the risk and the type of risk (cost, deadline, quality).
4. Click **OK**.

Project deliverables

A project deliverable defines the result of a project and its impact on or its contribution to the architectural solution landscape of the enterprise.

It is defined by a solution block (example: an organization, an application, an infrastructure element) delivered by the project in the target architectural landscape. Within the framework of a project deliverable, a block can be:

- New: the project delivers a new block to the target architectural landscape.
- Updated: the project modifies an existing block in the current landscape, for example by extending its lifecycle, and delivers the updated version to the target architectural landscape.
- Deleted: the project deletes an existing target architectural block, which will therefore not be part of the target landscape.

To add a deliverable to the project:

1. In the **Deliverables** section, click **New**.
   The window for creating a deliverable appears.
2. Specify if you want to:
   - create a new block
   - update an existing block
   - decommission an existing block
3. Click **Next**.
4. Specify:
   - the deliverable name
   - the deliverable type
   - the deliverable production dates
5. Click **OK**.

> ☛ *You can also manage Project Deliverables using the Project Structure facilities. For more details, see Project Structure.*

Deliverable production dates

To model component change scenarios for elements in your portfolio without impacting the life of components in place, you will associate an *object life* with the deliverables.

> 🕮 *An object life is a set of time periods representing the updated calendar of object life cycle states.*

When the project is terminated (via the corresponding workflow command), the life cycle of deliverables is automatically transferred to the objects concerned.

To define the life of a project deliverable:

1. In the **Deliverables** section, select the deliverable in question.
2. Click **Properties**.
   The properties window of the deliverable appears.
3. Click the drop-down list then **Object Life**.
4. Click **New**.
   The creation of object life dialog box appears.

5. Specify the following characteristics:
   - the **life cycle** that defines the list of possible object states.

     A life cycle is represented by all the states an abject can take over time, as well as the transitions enabling change of state.

     For more information on proposed life cycles, see Defining Life Cycles.
   - a **Begin Date** and an **End Date** which enable positioning of the object life in time.
6. Click **OK**.
   A Gantt char is used to view the steps of the life cycle of a deliverable.

On the project, the **Gantt char for the lifecycle of the project deliverables** details the lifecycle of the project deliverable.

Project costs

The specification of the costs of a project take place through the cost lines.

One or more cost lines can be associated with a project.

A cost line enables identification of cost kind and type.

A cost line is characterized by:

- a type: operating or capital;
- a nature: infrastructure (for a deployment), license (for an application), service, manpower;
- state of the cost line.

Associated with a cost line can be:

- a periodic expense
- one or several fixed expenses

Creating a cost line

To create a cost line for a project:

1. Expand the **Costs** section.
2. Under **Cost Line**, click **New**.
   The **Creation of a cost line** box opens.
3. To create a single cost line, select option **Create only one cost line**.
4. Click **Next**.
5. Specify the **Name** of the cost line.
6. Select the **Cost Type**.
7. Select the **Cost Nature**.
8. Select the **state** of the cost line.

   The states proposed in the drop-down list are the states of the life cycle associated with the object life.
9. Click **Next**.
10. Define the periodic expense.

    Fixed expenses, which can be multiple, are defined separately. For more details on fixed expense creation, see Adding a fixed expense.
11. Click **OK**.
    The new cost line appears in the **Cost Line**.

Adding a fixed expense

To associate a fixed expense with a cost line:

1. In the **Cost Line** section, select the cost line that interests you.
2. In the **Cost Line Expenses** section, click **New**.
   The **Creation of Expense** dialog box opens.
3. Specify:
   - the **Name** of the expense
   - the **Date** of the expense,
   - the **Amount** of the expense.
4. Click **OK**.
   The new expense appears in the **Fixed Expenses** section.

Describing a Project Milestones

> 🕮 *A project milestone defines an intermediate delivery step in the life cycle of the project life. A project deliverable can be associated with a project milestone if it is delivered during the project and on the project date.*

Between the scheduled start and end dates, intermediate milestones can be defined and associated with deliverables.

> ☛ *Associating a project deliverable with a milestone does not affect automatic initialization of its life cycle; it can be subject to a manual modification if appropriate.*

Within the framework of project progress follow-up, you can define the level of progress for each milestone.

To add a milestone to a project:

1. Open **Project Milestones** page of the Project that interests you.
2. In the **Milestones** section, click **New**.
   The window for creating a milestone appears.
3. Specify:
   - The local name
   - The scheduled date of the milestone
   - A comment if required
4. Click **OK**.

Using Projects Portolios

> 🕮 *A projects portfolio gathers all the projects of a given Project Domain in the defined project state (Demands, Candidate Projects, Ongoing Projects). In a Portfolio, Projects can be assessed and compared against Projects Criteria (derived from projects information or project qualitative assessment) or Portfolio criteria (additional criteria defined at portfolio level).*

The portfolio management process can be represented in three sub-steps:

- Project selection: a restricted list of projects is drawn up according to selection criteria (strategic, financial etc.). The projects are classified

according to the strategic perspectives (the domains) used in the organization.
- Analysis and arbitration: the best project combination is defined to maximize the objectives and the restrictions of the portfolio.
- Follow-up: the portfolio's performance indicators ensure the alignment of the portfolio with the strategy of the organization.

Accessing Projects Portfolios

To access Projects Portfolios:

1. From the **Project** navigation menu, select **Taxonomy**.
2. Click the **by Porfolio** tab.
   The list of the Enterprise Projects Portofilos is displayed.

Portfolio Types

There are two types of project portfolios:

- Arbitration portfolios, created automatically, which are divided into two groups:
  - project demand portfolios
  - candidate project and ongoing project portfolios, used to compare candidate projects with ongoing projects

    ☛ *For more details, see Project Domains*
- Analysis portfolios you can create later and which make up sub-sets within the arbitration portfolios.

Grouping Projects by Portfolios

Grouping projects by portfolio summarizes the information relating to different projects to facilitate decision-making.

Arbitration portfolio

Project arbitration portfolios group all the projects created according to their domain.

When you create a project domain, two types of arbitration portfolios are created by default and associated with this domain:

- Domain name - demand arbitration portfolio
- Domain name - arbitration portfolio of candidate projects and ongoing projects

Each new project appears in the dedicated portfolio.

In an arbitration portfolio, the projects can be assessed and compared according to a number of criteria:

- project criteria: these come from information on the projects (for example, the costs) or the qualitative evaluation of the project (for example, the level of strategic alignment).
- portfolio criteria: criteria that can be defined at the portfolio level, above the project criteria.

See also: Creating a Project Domain.

Analysis portfolio

You can create an analysis portfolio in an arbitration portfolio; it groups a sub-set of parent arbitration portfolio projects. It can be defined to assign certain projects to a specific portfolio manager.

Portfolio Lines

For each project added to a portfolio, a portfolio line is created.

A project portfolio line is used to assess the project in the context of a portfolio. It is linked to assessment criteria and provides the global note of the project in the context of the portfolio.

Assigning a Portfolio to Persons

The persons who can be assigned to a project portfolio are:

- the demand approver
- the portfolio manager
- the portfolio approver

For more information on profiles, see Roles in HOPEX Project Portfolio Management.

PROJECTS - STRUCTURE

The **Structure** aspect of the **Projects** view helps **UAF Architects** to classify the Entreprise projects using *Projets Domains* and to get a vision of the deliverables over the time.

Project Domains

A project domain gathers all the projects managed from the same perspective in terms of portfolio management decisions. Project Portfolios are therefore also attached to Project Domains; by default, one portfolio of each portfolio type is created when creating a new Project Domain.

Creating a Project Domain

To create a Project Domain:

1. From the **Projects** navigation menu, select **Structure**.
2. Select the **Project Domains** Tab.
   The list of Project Domains appears.
3. Click **New**.
   A creation dialog box opens.
4. Enter the name of the domain.
5. Click **OK**.

When you create a project domain, the two types of project portfolios that correspond to the different project statuses (project demands, candidate projects and projects in progress), are also created. They are visible in the **Projects > Structure** navigation menu, selecting the **Project Portfolios** tab.

For more details on Project Portfolios, see Using Projects Portolios.

Assigning a Domain to Persons

It is possible to define particular roles for users on a domain; these roles are then valid for all the projects in the domain.

To assign a person to a domain:

1. Display the domain properties.
2. Click the **Assignment** page.
3. Click **New**.
4. In the window that opens, select the person or person group.
5. Select their role. You can define the following roles:
   - Demand Approver
   - Project Portfolio Approver
   - Project Portfolio Manager
6. Click **OK**.

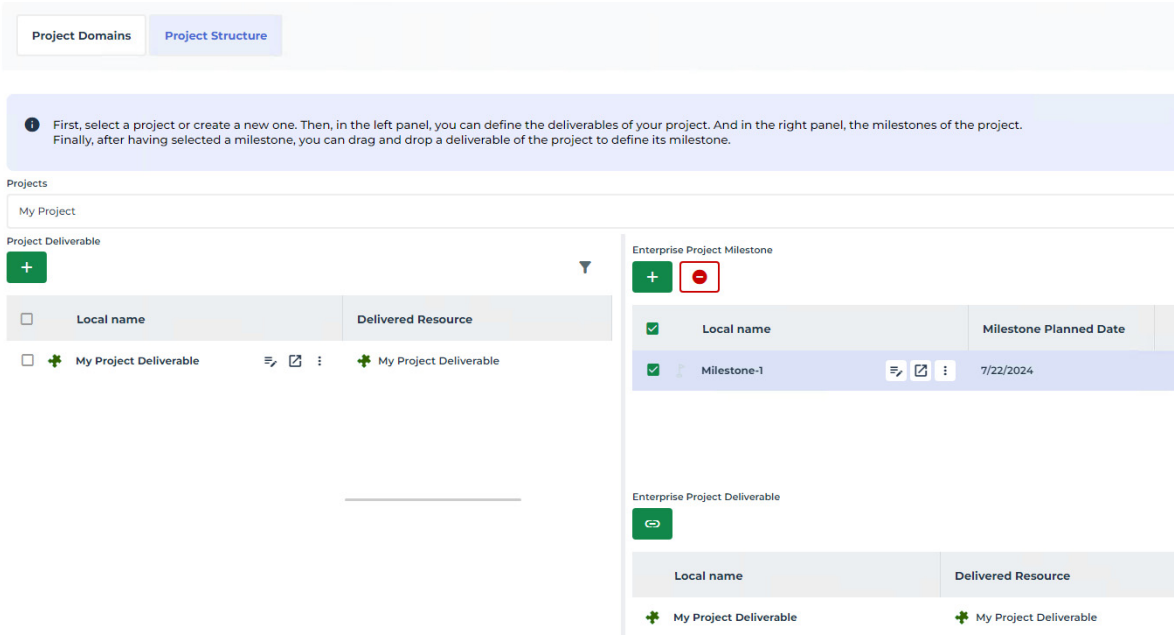
Project Structure

The **Projects Structure** tab helps **UAF Architects** to get a vision of a project deliverables over the time using *Project Milestones*.

> 🕮 *A project milestone defines an intermediate delivery step in the life cycle of the project life. A project deliverable can be associated with a project milestone if it is delivered during the project and on the project date.*
>
> ☛ *For more details on Project Milestones, see Describing a Project Milestones.*
>
> ☛ *For more details on Project Deliverables, see Project deliverables.*

To specify the list of projects deliverables associated to *Project Milestones*:

1. From the **Projects** navigation menu, select **Structure**.
2. Select the **Project Structure** tab.
3. In the **Project** field, select the Project that interests you.
   The list of Project Deliverables appears in the left side of the edition area and le list of project Milestones appears on the right side.
   > ☛ *You can create a deliverable or a milestone from this wizard.*



4. Select a *Project Milestone*.
   The list of Deliverables associated to the Milestone is displayed in the **Enterprise Project Deliverable** section.
5. Select a Project Deliverables in the left side of the edition area, and drop it in the **Enterprise Project Deliverable** section corresponding to the selected Milestone.

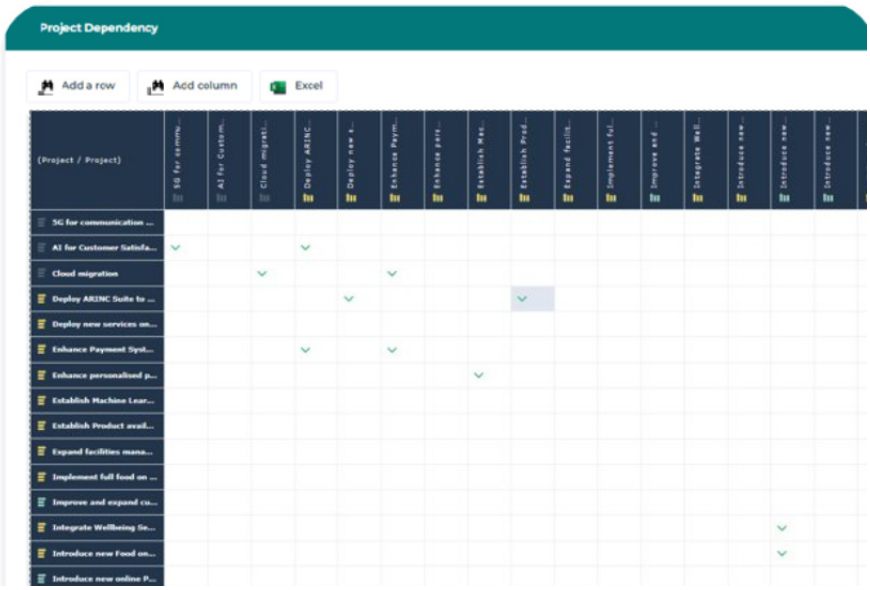
PROJECTS - CONNECTIVITY

The **Connectivity** aspect of the **Projects** view helps management of the dependencies between Projects to ensure that project deliverables are in the available milestones sequence.

Connectivity Matrix

Matrix enabling creation of Project Dependencies between two projects.

☛ *For more information, see: Project dependencies.*



The matrix shows that dependencies exists between projects. You can create a dependency or delete an existing one using this matrix.

To create a Connectivity Matrix:

1. From the navigation menu, select **Projects > Connectivity**.
2. Select the **Connectivity Matrix** tab.
   The list of existing matrix is displayed.
3. Click the **New** button.
   The new matrix is created.
4. Click the matrix to open it.
5. Use the **Add row** and **Add column** butons to add projects in the matrix.
   The dependencies are represented by check marks.

6. Click the box between projects to add or delete a dependency.

☛ *You can access the dependencies of a projet using the Properties button. See: Project dependencies*

Connectivity Reports

The Connectivity Report shows the dependencies of a specific list of projects.

☛ *For more information, see Project dependencies.*

To create a Connectivity Report:

1. From the navigation menu, select **Projects > Connectivity**.
2. Select the **Connectivity Reports** tab.
3. Click the **New** button.
   The creation wizard opens.
4. Enter the **Name** of the report and click **Next**.
5. Click **Connect** to select the list of projects that interest you.
6. Click **OK**.
   The Project Dependency report is displayed with the dependencies of the selected projects.

PROJECTS - PROCESSES

The **Processes** aspect of the **Projects** view provides access to the *Personnel Functions* of the current project.

> 🕮 *A Personnel Function is a set of operations performed by org-units within a company or organization, to produce a result. It is depicted as a sequence of operations, controlled by events and conditions. In the BPMN notation, the Personnel Function represents a sub-Personnel Function from the organizational point of view.*

To access the list of *Personnel Functions*:

1. From the **Projects** navigation menu, select **Processes**.
   The list of the Personnel Functions is displayed.
2. Click the **New** button to create a new one.

For more details on the use of *Personnel Functions* with **HOPEX UAF**, see Personnel Functions.

PROJECTS - ROADMAP

The **Roadmap** aspect of the **Projects** view provides reports dedicated to the Gantt chart for projects.

Accessing the Projects Roadmap

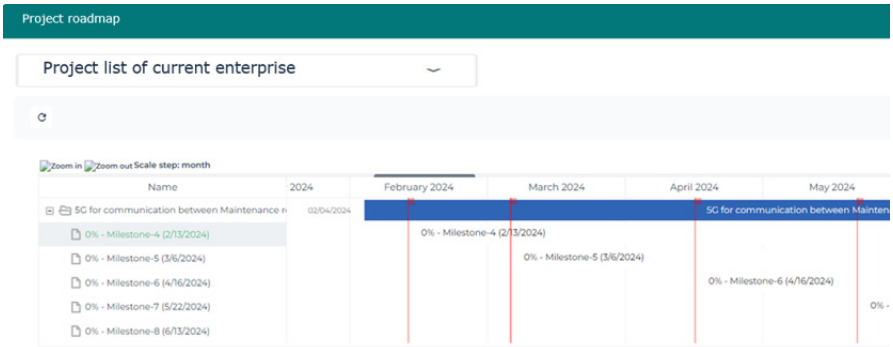
To access the Project RoadMap reports:

1. From the navigation menu, select **Projects > RoadMap**.
2. Click the **Projects Roadmap** tab.
3. In the **Projects** field, select the Project that interests you.
   The "Project RoadMap" report is displayed presenting the Gantt chart for projects and the road maps for project deliverables.

The Gantt chart presents one row per project. The following information is provided for each project:

- Start and end dates
- Progress
- Dependencies
- Declared delays

> ☛ *See also*
> - *Follow-up of Ongoing Projects*
> - *Project dependencies*.



Accessing the Project Portfolios Roadmap

To access the Project Portfolio RoadMap reports

1. From the navigation menu, select **Projects > RoadMap**.
2. Click the **Project Portofolios Roadmap** tab.

3. In the **Projects Portofolios** field, select the Project that interests you. The "Project Portoflio RoadMap" report is displayed presenting the Gantt chart for Portofolio projects and the road maps for Portofolioproject deliverables.

The following information is displayed for each portfolio:

- The projects included
- The dependencies
- The status of projects
- The project progress
- The dates defined for the project

☛ *See also Follow-up of Ongoing Projects.*

PROJECTS - TRACEABILITY

The **Traceability** aspect of the **Projects** view provides access to matrixes and reports presenting the Capabilities impacted by Projects.

☛ *For more details, see: Transformation Purpose*

- Projects are presented in rows

  🕮 *A project is a temporary endeavor undertaken by a specific team, to create a unique product, service or result. Its serves a purpose which can be expressed in terms of capability that is acquired (new), extended (improvement) or decommissioned (rationalization).*

- Capabilities are presented in columns

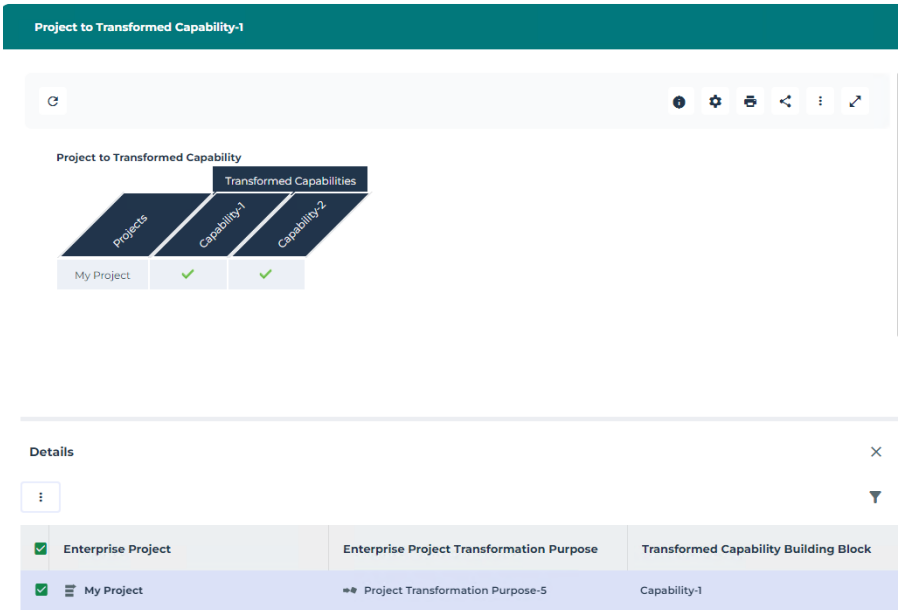
  🕮 *A Capability is a set of features that can be made available by an enterprise.*

For more details on **Traceability** matrixes and reports, see Using Traceability Aspect.

You can access the **Transformation Purposes** of a projet using the Properties button. See: Transformation Purpose

Two types of matrix and report are provided:

- The **project to Capability** matrix and report , concerning the capabilities associated to projects transformation purpose
- The **Project to Transformed Capability** matrix and report , concerning the exhibited capabilities associated to projects transformation purpose.



STANDARDS VIEWPOINT

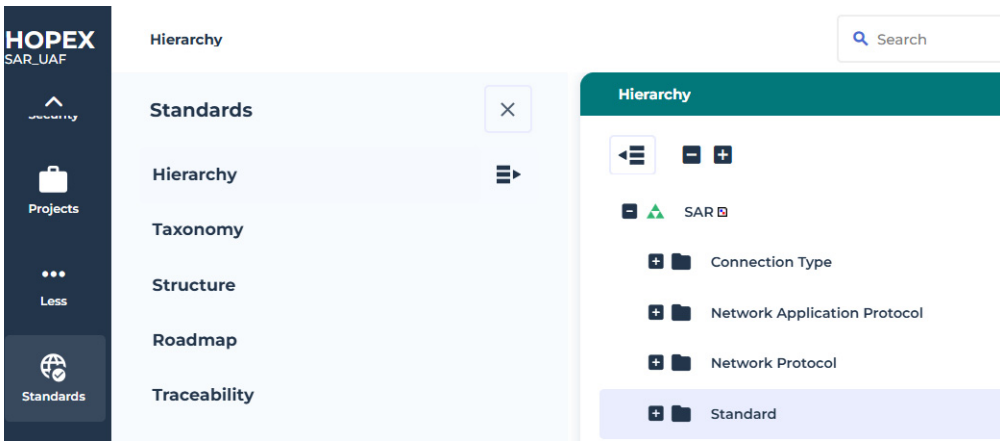
The **Standards** viewpoint is dedicated to the composition of standards required to achieve the architecture's objectives. It mainly addresses the specification of the protocol stack used in the architecture.

This chapter comprises the following sections:

- ✓ Standards - Hierarchy
- ✓ Standards - Taxonomy
- ✓ Standards - Structure
- ✓ Standards - Roadmap
- ✓ Standards - Traceability

STANDARDS - HIERARCHY

The **Hierachy** aspect of the **Standard** view helps UAF Architect to show the technical, operational, and business Standards applicable to the architecture. Defines the underlying current and expected Standards.



To get more information about the use of folders in **HOPEX** trees, see **HOPEX**, "Handling Trees" chapter of the **HOPEX Common Features** guide.

The hierarchy navigation enables to navigate on the main concepts of the view:

- Connection Type, see Defining Connection Types,
- Network Application Protocol, see Defining Network Application Protocols,
- Network Protocol, see Defining Network Protocols,
- Standards, see Defining Standards.

STANDARDS - TAXONOMY

The **Taxonomy** aspect of the **Standards** view shows the taxonomy of types of Communication Agents.

Defining Standards

A standard is a definition or format that has been approved by a recognized standards organization or is accepted as a de facto standard by the industry.

Accessing Standards

HOPEX UAF helps you access *standards* in a list or in a navigation tree.

To access the list of *standards*:

1. In the navigation menu click **Standards > Hierachy**.
2. The list of *standards* apperas in the corresponding folder.

> *For more details on a Standard sub folders, see Specifying standard components and Specifying Elements Using standard*

> *A Standard Type can be referenced by a Standard or by a Standard Component.*

Creating a Standard

To create a Standard in **HOPEX UAF**:

1. From the navigation menu, select **Standards > Taxonomy**.
2. Click the **Standards** tab.
3. Click **New** button.
4. Enter the name and click **OK**.
   The Standard appears in the list.

> *A Standard Type can be referenced by a Standard.*

Specifying standard components

A *standard* can be decomposed into sub-parts. Each part is called a *Standard Component*.

To specify the components of the standard:

- In the **Characteristics** page of the standard, specify components in the **Owned Standard Components** section.

Specifying Elements Using standard

A standard can be used by several types of object.

The **Standard Traceability Matrix** provides the list of elements using a standard.

☛ *For more details, see Standards - Traceability*

To specify that an object uses a standard:

1. In the **Characteristics** page of the *standard*, expand the **Element Using Standard** section.
2. Click **Connect**.
3. Select the objet type and the object.
4. Click **Connect**.

Defining Connection Types

📖 *A connexion type defines a local use of a standard network application protocol (protocol and port number) which can be override and completed by source port number restriction and transport protocol (TCP, UDP).*

Creating a Connection Type

To create a *Connection Type* in **HOPEX UAF**:

1. From the navigation menu, select **Standards > Taxonomy**.
2. Click the **Connection Types** tab.
3. Click **New** button.
4. Enter the name.
5. Select the **Network Application Protocol**.

   ☛ *For more details, see Defining Network Application Protocols*
6. Enter the Port numbers.
7. Select the **Transport Protocol**.
8. Click **OK**.
   The Connection Type appears in the list.

Specifying Standard used by a Connection Type

A standard can be used by a Connection Type.

The **Standard Traceability Matrix** provides the list of elements using a standard.

☛ *For more details, see Standards - Traceability*

To specify the standard used by a Connection Type:

1. Open the **Standards** page of the Connection Type.
2. Click **Connect**.
3. Select the standard that interest you.
4. Click **Connect**.

Defining Network Application Protocols

📖 *A network application protocols supported by a communication port must be compatible with the protocols supported by communication ports to which they are connected.*

To create a **Network Application Protocol** in **HOPEX UAF**:

1. From the navigation menu, select **Standards > Taxonomy - Structure**.
2. Click the **Network Application Protocol** tab.
3. Click **New** button.
4. Enter the name and click **OK**.
   The Network Application Protocol appears in the list.

Defining Network Protocols

A network protocol is a set of standardized rules for transmission of information (voice, data, images) on a communication channel. The different layers of protocols can handle the detection and processing of errors, authentication of correspondents, management of routing.

Creating a Network Protocol

To create a **Network Protocol** in **HOPEX UAF**:

1. From the navigation menu, select **Standards > Taxonomy - Structure**.
2. Click the **Network Protocol** tab.
3. Click **New** button.
4. Enter the name and click **OK**.
   The Network Application Protocol appears in the list.

Specifying Standard used by a Network Protocol

A standard can be used by a Network Protocol.

The **Standard Traceability Matrix** provides the list of elements using a standard.

☛ *For more details, see Standards - Traceability*

To specify the standard used by a Network Protocol:

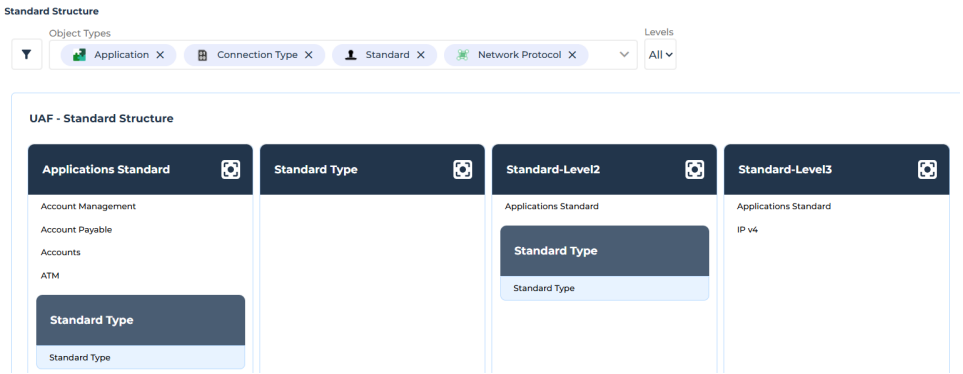
1. Open the **Standards** page of the Network Protocol.
2. Click **Connect**.
3. Select the standard that interest you.
4. Click **Connect**.

STANDARDS - STRUCTURE

The **Structure** aspect of the **Standards** view helps UAF Architect to see the applicability of standards to specific elements in the architecture.

The breakdown report shows the standard and their standard composition as boxes.

The user can hide or display the Asset types.



In boxes, you can see:

- The Agent using the standard, see *Specifying Elements Using standard*,
- The *Standards* using a *Standard component*, see *Specifying standard components*,
- The *Standard Type* refrenced by the *Standard* or the *Standard component.*

STANDARDS - ROADMAP

The **Roadmap** aspect of the **Standards** defines the underlying current and *expected Standards*. *Expected Standards* are those that can be reasonably forecast given the current state of technology and expected improvements / trends.

Standard Asset Catalogs Access

To access a Standard Asset Catalog:

1. From the navigation menu, select **Standards > RoadMap**.
2. Using the arrow at the right of the **Standard Asset Catalog** field, select the Catalog that interests you.
   The tree of catalog components appears below.

Standard Asset Catalogs Creation

To create a Standard Asset Catalog:

1. From the navigation menu, select **Standards > RoadMap**.
2. Click the arrow at the right of the **Standard Asset Catalog** field, select **New**.
   A Standard Asset Catalog is created, the corresponding tree appears below.

Creating a Standard Catalogued Asset

To create a Standard Catalogued Asset:

1. Select Standard Asset Catalog that interests you.
2. In the tree of catalog components, click **Plus** button.
   An **Creation of Object Life** window opens.
3. Using the arrow at the right of the **Catalogued Asset** field, click **Create** button.
4. Select the **Object Type** and click **Next**.
5. Create a Capability or reuse existing ones and click **Add**.
6. Enter the name of the object and click **OK**.
   The Catalogued Asset is created.
7. Specify the **Lifecycle to follow**.
8. Specify the **Begin Date** and the **End Date**.
9. Click **OK**.

Standard Asset Catalog Report

To access a Standard Asset Catalog Report:

- Open the **Gantt Chart** property of the Standard Asset Catalog that interests you.

STANDARDS - TRACEABILITY

The **Traceability** aspect of the **Standards** view provides access to the **Standards to Assets** matrixes and reports that help UAF Architect to see the applicability of standards to specific elements in the architecture.

- *Standards* are presented in rows

  > 🕮 *A standard is a definition or format that has been approved by a recognized standards organization or is accepted as a de facto standard by the industry.*

- The assets are presented in columns

For more details on Assets using standards, see Specifying Elements Using standard.

For more details on **Traceability** matrixes and reports, see Using Traceability Aspect.

Standard Traceability reports

Two reports are provided in the **Traceability** aspect of the **Standards** view.

- Standard Traceability Matrix,
- Standard Traceability Table.

Standard Traceability Matrix

The **Standard Traceability Matrix** defines a matrix report presenting the connections between standards and Enterprise Assets.

> ☛ *For more details, see Specifying Elements Using standard*

In this matrix, standards are presented in lines and columns provide the list of the architecture elements using the standards.

> ☛ *For more details, see Specifying Elements Using standard*

Standard Traceability Table

The **Standard Traceability Table** details the elements using a list of standards.

Standard Traceability Table

Standard	Standard Description	Object Type	Element Using Standard	Description
Applications Standard		Application	Account Management	Organize client interactions, track progress, and nurture relationships
			Account Payable	Manage invoices, track payments, and streamline vendor communication.
			Accounts	Monitor transactions, categorize expenses, and track balances
		Connection Type	ATM	
Standard-Level2		Standard	Applications Standard	
Standard-Level3		Network Protocol	IP v4	
		Standard	Applications Standard	

This report presents a table with standards in lines. For each one, columns provide:

- The standard Description
- The type of the element using the standard
- The name of the element using the standard
- The element description

☛ *For more details, see Specifying Elements Using standard*

THE UAF TERMINOLOGY

As an enterprise architecture tool, **HOPEX** can be used to implement a UAF 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** 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 not exactly equivalent to those described in UAF standard.

This section details the mapping made between the **HOPEX** and UAF concepts used. The aim of this renaming is to make the mapping between the UAF concepts and the **HOPEX** concepts as invisible as possible for the user.

Strategic

UAF Concept	HOPEX Concept	Definition
capability	**business capability**	A Capability is a set of features that can be made available by an enterprise.
capability composition	**business capability composition**	A Capability Composition is the involvement of a Capability in the context of a Capability map (one and only one) linked to an enterprise.
capability dependency	**business capability dependency**	A Capability Dependency is a relationship which asserts that a dependent Capability depends upon a needed capability in the context of a capability dependency owner, with respect to its needed effect (business outcome).

UAF Concept	HOPEX Concept	Definition
capability map	**business capability map**	A Capability Map is an assembly of Capabilities and their Dependencies that, together, provide a capability scope for an Enterprise Stage.
capability process fulfillement	**business capability process fulfillement**	A Capability Process Fulfillement defines the Personnel Function that realizes the Capability.
enterprise	**enterprise**	An Enterprise is a purposeful undertaking, conducted by one or more organizations, aiming at delivering goods and services, in accordance with the enterprise mission in its changing environment. During its development over time, an enterprise has to adapt to its environment and sets up transformation goals and objectives along with course of action to achieve these objectives. The design and realization of the resulting transformation stages may transcend organizational boundaries and consequently require an integrated team working under the direction of a governing body to involve stakeholders in transformation initiatives.
enterprise event	**enterprise event**	An Enterprise event is something that happens at a given place and time and that marks the end of one Enterprise State and possibly the beginning of another.
enterprise stage	**enterprise stage**	An Enterprise stage is a past, current or future stage of an enterprise.
enterprise stage component	**enterprise stage component**	An Enterprise Stage Component is a composition of a sub-enterprise state in the context of an Enterprise or Enterprise Stage.
enterprise stage end event	**enterprise stage end event**	An Enterprise stage End Event is the end event of an Enterprise State.
enterprise stage start event	**enterprise stage start event**	An Enterprise stage Start Event is the start event of an Enterprise State.
exhibited capability	**exhibited business capability**	An exhibited Capability is a Capability that is exhibited by an Enterprise Stage with quantified measure (KPI) and potential geopolitical scope (Site) for a defined market segment (Operational Partner).
fulfillement	**fulfillment**	A Fulfillment describes the relationship between a logical entity and a physical entity that implements it. The physical entity gives the list of logical entities that fulfill it.

UAF Concept	HOPEX Concept	Definition
goal	**goal**	A Goal tends to be longer term, and defined qualitatively rather than quantitatively. It should be sufficiently narrow-focused that Objectives can be defined for it.
IT transformation stage	**IT transformation stage**	An IT Transformation Stage is a type of Enterprise Transformation Stage aiming at the alignment of its IT System to expected functionalities required for the business to operate effectively at the right cost.
library	**library**	Libraries are collections of objects used to split repository content into several independent parts. They allow creation of virtual partitions of the repository. In particular, two objects owned by different libraries can have the same name.
measurable property	**measurable property**	A measurable property expresses the nature of indicators (duration, mass, cost, etc.) and defines the unit used to measure them (minutes, kilograms, euros, etc.). Measurable properties are used to define indicators, they can be elementary or composite. Elementary measurable properties are described by measurement units: kg, Liter, Gallon, Hour, Minute.
mission	**mission**	A mission indicates the ongoing operational activity of the enterprise. The Mission describes what the business is or will be doing on a day-to-day basis. A Mission makes a Vision operative; that is, it indicates the ongoing activity that makes the Vision a reality. A Mission is planned by means of Strategies.
objective	**objective**	An objective is a goal that a company/organization wants to achieve, or the target set for a Personnel Function or an operation. An objective allows you to highlight the features in a Personnel Function or operation that require improvement.
partner operational domain role	partner business functional area use	A Partner Operational Domain Role is a usage of a Operational Domain which is external to the considered environment.
qualifying Values	**qualifying value**	A qualifying value (key indicator) defines how much of something can be quantified, either as a singular value or as range of values, according to a Measurable Property. Key indicators are valued. Example: Response Time < 20 secondes.
set of constraint values	**set of constraint values**	A set of constraint values defines the grouping of elementary Qualifying values that should be examined together in order to appreciate the actual performance of a KPIed item. E.g.: a delivery must take place in less than 20 minutes and cost less than 5 euro.

UAF Concept	HOPEX Concept	Definition
state	**state**	A state is a condition or situation in the life of an object, during which it satisfies some condition, performs some activity, or waits for some event. A state represents an interval of time delimited by two events. It is a phase an object passes through during its life cycle.
state machine	**state machine**	A state machine is a set of states and transitions governing the state changes that can match any time-dependent object.
strategy	**strategy**	A Strategy is one component of the plan for the Mission. A Strategy represents the essential Course of Action to achieve Ends (Goals in particular). A Strategy usually channels efforts towards those Goals. A Strategy is accepted by the enterprise as the right approach to achieve its Goals, given the environmental constraints and risks.
tactic	**tactic**	A Tactic is a Course of Action that implements part of a Strategy. Tactics generally channel efforts towards Objectives.
transformation stage	**transformation stage**	A transformation stage is a type of enterprise transformation stage aiming at the alignment of the enterprise operating model to its strategy and corresponding exhibited Capabilities.
transition	**transition**	A transition is passage of an object from one state to another. A transition is the response of an object to an event it receives. When an event occurs and certain conditions are satisfied, the object executes certain actions while still in the first state, before passing to the second state.
vision	**vision**	A Vision is the ultimate, possibly unattainable, state the enterprise would like to achieve. A Vision is often compound, rather than focused toward one particular aspect of the business problem. A Vision is supported or made operative by Missions. It is amplified by Goals.

Operational

UAF Concept	HOPEX Concept	Definition
arbitrary connector	**sketching link**	An Arbitrary Connector is a link between the two Concept Roles, represented in the form of a stroke.
concept	**concept**	A concept is a statement expressing the essential nature of a being, an object, a word through its essential properties and characteristics or its specific qualities.
concept domain	**concept domain**	A concept domain is a sub-set of elements of a subject area that reduces the scope of a fields.
concept property	**concept property**	A concept property is a category of linguistic unit used to represent properties of a concept: name, description, etc.s.
concept role	**sketching item**	A Concept Role part of a High Level Operational Concept represented as a shape.
concept type	**concept type**	A concept type enables classification of concepts. Relationships between concept types are represented by concept type components.
concept view	**concept view**	A concept view enables representation of the semantic scope covered by a business object. A concept view is based on the selection of several concepts specific to the view.
high level operational concept	**sketch**	A High Level Operational Concept is a rough draft of a model; unformal representation.
interface	**service interface**	An Interface is a template of a contract between entities (organizational, IT …). The contract is described by available operations which can be triggered trough messages exchanged by roles (vendor, buyer..).
interface use	**service interface use**	An Interface use is associated to an interface. It enables representation of complex exchanges.
operational activity	**value stream**	An Operational Activity is an end-to-end collection of Activity Actions that creates an outcome for a customer, who may be the ultimate customer or an internal end-user of the Operational Activity.

UAF Concept	HOPEX Concept	Definition
operational activity action	**value stage**	An Operational Activity Action is a distinct, identifiable phase or step within an Operational Activity that has a unique entrance criteria, exit criteria, and identifiable participating Operational Performer or Operational Domain.
operational activity fulfill-ment	**value stream fulfillment**	A Operational Activity Fulfillment follows the Class of Building Block Realization pattern and materializes the fulfillment of a Operational Activity by a BPMN Personnel Function, for instance.
operational architecture	**business architecture environment**	A Operational Architecture represents the relationships of a Operational Domain with its partners.
operational constraint	**business rule**	An Operational Constraint is a rule that is under business jurisdiction.
operational domain	**business functional area**	A Operational Domain is a grouping of Operational Performers and their associated Operational Activities on the conjunction of two main criteria. 1.Their need in accomplishing one or more Capabilities. 2.The common skills and functionalities required to accomplish these Capabilities.
operational domain opera-tional activity performance	**business functional area operational activity performance**	An Operational Domain Operational Activity Performance is the relationship between a Operational Domain and a Operational Activity that it performs.
operational domain role	**business functional area Use**	An Operational Domain Role is the usage of an Operational Domain which is internal to the considered environment.
operational partner	**business partner**	An Operational Partner is a conceptual stakeholder type that interacts with the enterprise in the context of the enterprise Operational Architecture. Examples : 'Private Sector Customer', 'Regulation Authority', 'Supplier'.
operational per-former	**business function**	An Operational Performer is a skill or grouping of skills of interest for the enterprise.
operational role	**business function component**	An Operational Role represents the configuration of a Operational Performer within another one.
partner opera-tional domain role	**partner business functional area use**	A Partner Operational Domain Role is the usage of a Operational Domain which is external to the considered environment.

UAF Concept	HOPEX Concept	Definition
request point	**request point**	A request point is a point by which an agent requests a service from potential suppliers.
role	**role**	A role is a participant in an interaction service, workflow or Personnel Function. It can be the initiator, that is the requester of a service, or it can represent a sub-contractor carrying out processing outside the service. A role is an integral part of the object that it describes, and is not reusable. It can subsequently be assigned to an org-unit internal or external to the organization or to an IT component. Examples: client, traveler.
service interaction	**service interaction**	A Service Interaction represents an interaction for service purpose between entities in a specific context inside or outside a company. These entities can be enterprise org-units, applications, activities or Personnel Functions, as well as external org-units. The content of this interaction is described in an Interface.
service operation	**service operation**	A service operation specifies exchanges between participants.
service operation use	**service operation use**	A service operation use represents the usage of a service operation in an service interface.
service point	**service point**	A service point is a point from which a system receives a request from another system and provides the requested service.
signal	**content**	A Signal designates the content of a message or a message flow, independently of its structure. Signals may be used by several messages or message flows, since it is not associated with an sender and a destination.
term	**term**	A term is a word or group of words that is used for a specific meaning in a specific context.

Services

UAF Concept	HOPEX Concept	Definition
business service catalog	**business service catalog**	A business service catalog provides a centralized information source for the business services offered by the service provider organization. It contains a customer-oriented view of the services associated to business capabilities, how they are supposed to be used, the processes that they support as well as the expected service quality level. The business service catalog presents the list of functionalities mentioned as well as implementation recommendations.
functionality	**functionality**	A Functionality is a service required by an org-unit in order to perform its work. This functionality is generally necessary within an activity in order to execute a specific operation. If it is a software functionality, it can be provided by an application.
functionality map	**functionality map**	A Functionality Map is an assembly of functionalities and their dependencies that, together, defines the scope of a hardware or software architecture.
hardware capability	**hardware capability**	A Hardware Capability is the ability to deliver a physical outcome which is required by an organizational resource in order to perform its work. This hardware capability is generally necessary within a process in order to execute a specific operation.
hardware capability map	**hardware capability map**	A Hardware Capability Map is an assembly of hardware capabilities and their dependencies that, together, defines the scope of a hardware or software architecture.
hardware service catalog	**hardware service catalog**	A hardware service catalog provides a centralized information source for the hardware services offered by the service provider organization. It contains a customer-oriented view of the hardware used, how they are supposed to be used, the processes that they support as well as the expected service quality level. The hardware service catalog presents the list of hardware functionalities mentioned as well as implementation recommendations.
hardware service catalog item	**hardware service catalog item**	A hardware service catalog item defines which hardware functionality is in the catalog and which hardware artifacts provide the hardware functionality.

UAF Concept	HOPEX Concept	Definition
functionality map	**functionality map**	A Technology Capability Map is an assembly of functionalities and their dependencies that, together, defines the scope of a hardware or software architecture.
implementation	**implementation**	An implementation describes the relationship between a logical entity and a physical entity that implements it. The physical entity gives the list of logical entities that it implements.
service catalog	**service catalog**	A service catalog contains a list of key service offers for which solutions are recommended.
service point	**service point**	A service point is a point from which a system receives a request from another system and provides the requested service.
technology capability	**technology capability**	A Technology Capability is the ability to deliver a technology service which is required by a technology artifact in order to function. This technology capability is generally required within a technical process handle by the technology artifact.
technology service catalog	**technology service catalog**	A technology service catalog provides a centralized information source for the technology services offered by the service provider organization. It contains a customer-facing view of the technology services in use, how they are intended to be used, the process they enable, and the levels and quality of service the customer can expect from each service. The technology service catalog provides the list of reference technology capabilities and their recommended implementation.

Personnel

UAF Concept	HOPEX Concept	Definition
business dictionary	**business dictionary**	A Business Dictionary is a branch of knowledge, a discipline or a field of study. It is used to describe all the elements defining your information architecture.
business skill	**business skill**	A Business Skill is a Capability of an Organization (human resource) that has been acquired by training.
event	**event**	An Event represents a fact or an action occurring in the system, such as updating client information. It is managed by a broker. An application indicates that it can produce the event by declaring that it publishes it. If an application is interested in an event, it declares that it subscribes to the event.
flow	**flow**	A flow represents the circulation of information between components or within a component. A flow can carry a content.
Gateway	**gateway**	A Gateway represents elements that are used to control how sequence flows interact as they converge and diverge within a process.
message	**message**	A Message represents information flowing within an enterprise or exchanged between the enterprise and its business environment. Messages can be information flows such as orders or invoices. For convenience, financial and material flows such as payments or product deliveries are also represented by messages.
message flow	**message flow**	A Message Flow represents circulation of information within a service interface. A message flow transports its content.
operational flow channel	**business flow channel**	An Operational Flow Channel is used to graphically group a number of flows into a single flow.

UAF Concept	HOPEX Concept	Definition
organization	**org-Unit**	An organization represents a person or a group of persons that intervenes in the enterprise Personnel Functions or information system. An organization can be internal or external to the enterprise. An internal organization is an organizational element of the enterprise structure such as a department, a service, or a workstation. An internal organization 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 organization is an organization that exchanges flows with the enterprise. Example: Customer, Supplier, Government Office.
personnel function	**process**	A Personnel Function is a set of operations performed by org-units within a company or organization, to produce a result. It is depicted as a sequence of operations, controlled by events and conditions. In the BPMN notation, the Personnel Function represents a sub-Personnel Function from the organizational point of view.
personnel function Action	**operation**	A Personnel Function Action is an elementary step in process executed by an org-unit. It cannot be broken down. An operation can be industrial (manufacturing a component), logistical (receiving a delivery), or can involve information processing (entering an order).
post	**position type**	A Post is a type of status assigned to an individual or a group of individuals which defines a job in an organization or hierarchy.
sequence flow	**sequence flow**	A Sequence Flow is used to show the order in which steps of an service contract will be performed. A sequence flow has only one source and only one target.

Resource

UAF Concept	HOPEX Concept	Definition
application	**application**	An Application is a software component that can be deployed and provides users with a set of functionalities.
application deployment architecture	**application deployment architecture**	An Application Deployment Architecture describes one possible deployment configuration of an application. It contains the deployment architectures to be hosted, recommends hosting architectures and identifies required communication techniques (communication protocols and port numbers) they use to communicate with each other. . An application may have several deployment architectures (E.g.: autonomous installation, horizontal or vertical deployment, etc.)
application environment	**application environment**	An application environment is used to represent a use context of an application. An application environment allows presenting the other application systems, applications, microservices or actors with which this application can interact.
application system	**application system**	An Application System is an assembly of other application systems, applications and end users interacting with application components to implement one or several functions.
application system deployment architecture	**application system deployment architecture**	An Application System Deployment Architecture describes one of the configurations possible for deploying an application system. It contains the deployment architectures of application components and specifies the communication protocols (and port numbers) they use to communicate with each other.
capability configuration	**resource architecture**	A Capacity Configuration is a combination of physical assets and organization configured to provide a capability.
client port	**technical client port**	A Client Port is a point used to request the opening of communications from a technical architecture or an application technical area in compliance with a particular communication protocol (SMTP, HTTP, etc.).
communication port	**communication port**	A Communication Port is a physical point of communication with a resource. It adheres to the specific communication protocol. A communication port implements service and requests points.

UAF Concept	HOPEX Concept	Definition
data domain	**data domain**	A Data Domain is a sub-set of enterprise data designed to be used together in the context of a business operation. For example, the "Sales" data domain contains at least the following data: Clients, Orders, Products. Each entity of a data domain has CRUD characteristics. The data domains define the functional data frontiers used both for allocating data to applications (see Data stores) and for the governance or data in data management.p
database	**database**	A Database is the physical object that enables storage and organization of logical data for use by programs corresponding to distinct applications, to facilitate the independent evolution of the data and the application programs.
deployable application package	**deployable application package**	A Deployable Application Package is a split of application code according to technical criteria for hosting purpose. For example, it may be N tiers, Front End/Back End/… or GUI/ Business Logic/Database etc... Each deployable application package is associated to required technologies (for running) and can host code for several IT services. Architect can also prescribes a kind of hosting artefact (IaaS/PaaS cloud service or IT server model).
deployable data package	**deployable data package**	A Deployable Data Package represents a data part of an application deployment that must be hosted and accessed by application services (code) to run. Each deployable data package is associated to required technologies (for data hosting and access) and can host several data structures. Architect can also prescribes a kind of hosting artefact (IaaS/PaaS cloud service or IT server model). Architect can also prescribes a kind of hosting artefact (IaaS/PaaS cloud service or IT server model).
facility	**facility**	A Facility is a model of site of interest for the enterprise. Examples: Data Center, Factory or Outlet
Hardware technology	**hardware**	Non-IT Hardware can embed computers. Together with their embedded computers, they provide information and IS services. Examples: Connected Truck with Delivery Calendar Application and connected Drone with Online Payment Application. Hardware device can also provide hardware functionalities. Example: Connected fridge providing ordering functionalities and of course a freezing hardware functionality and connected drones fly and provide Online Payment.

UAF Concept	HOPEX Concept	Definition
IoT device	**IoT device**	An IoT Device is both a hardware device and a computing device which provides cvombined hardware and information services to the users using it directly. As a hardware device, it embeds sensors - e.g. accelerometer - which provide data to the embedded computing device. As a computing device, it can host data stores or run applications. Examples: smartwatch with GPS tracker, on-line surveillance video camera with live IP video feed, connected weighting scale with weight history management
IT Infrastruc-ture	**IT infrastructure**	An IT Infrastructure is composed of several connected computer devices (IT network nodes or computing systems) and computer networks.
IT network	**IT network**	An IT network is set of IT equipment components (e.g.: routers, switches, firewalls) that allow remote communications between computing devices (e.g.: IT server). An IT network can be broken down into sub-networks.
IT server	**IT server**	An IT Server is an IT component providing a service to users connected via an IT network. This IT component can house databases and run applications.
IT service	**IT service**	An IT Service is a software component of an application, that can't be deployed alone and that realizes a sub-set of the functionalities of this application either for end users of this application or inside the application (or another application). This includes batch programs.
microservice	**microservice**	A Microservice is a software component that can be deployed autonomously, but which does not directly provide an end user service. It can interact with other application services, applications or application systems. This is a deployable software component that uses software technologies. For example: an authentication service, a PDF file printing service.
network chan-nel	**network channel**	A Network Channel is a physical connector between resource elements. It supports service interactions defining communication protocols between physical resources It connects external resource elements through their Communication Ports.
network device	**network device**	A Network Device can host and run Software Technology. Conjointly with its hosted software, it provides services. This consists of, for example: Wifi Access Point, Firewall, router, switch, printer, Hard Drive.

UAF Concept	HOPEX Concept	Definition
package connection	**package connection**	A Package Connection represents a connection requirement between deployable packages. A client (package) requires a connection to a server (package). The connection is defined by a Connection Type and can be characterized with technical flow measures.
relational scheme	**relational schema**	A Relational schema represents a set schema stored in a database management system and used in the technical architecture of an application.
resource configuration	**resource configuration**	A Resource Configuration is a set of physical and human resources configured to provide a business capability.
resource function	**system process**	A Resource Function is the executable representation of a process. the events of the workflow, the tasks to be carried out during the processing, the algorithmic elements used to specify the way in which the tasks follow each other, the information flows exchanged with the participants.
resource function action	**task**	A Resource Function Action is an elementary step that is included within a Resource Function. A Resource Function Action is used when the work in the Resource Function is not broken down to a finer level of the process. Generally, an end-user and/or an IT service are used to perform the Resource Function Action when it is executed.
server port	**technical server port**	A Server Port is a point used to open communications with a technical architecture or an application technical area in compliance with a particular communication protocol (SMTP, HTTP, etc.).
software technology	**software technology**	A Software Technology is a basic component necessary for operation of business applications. Software technologies include all basic software such as: application server, electronic mail server, software components for presentation, data entry, storage, business information sharing, operating systems, middleware, navigators, etc.

Security

UAF Concept	HOPEX Concept	Definition
regulatory framework	**regulatory framework**	A regulatory framework is an authority document falling under any of following categories: regulations (rules of law that, if not followed, can result in penalties), guidelines, standards, best practices.
policy framework	**policy framework**	A policy framework represents internal documents issued by the Organization, such as code of conducts, standard security measures and similar.
control directive	**control directive**	A control directive is an interpretation of the law and contributes to the enforcement of any regulation article your organization has to comply with.
risk	**risk**	A risk is a hazard of greater or lesser probability to which an organization is exposed.
risk type	**risk type**	A risk type defines a risk typology standardized within the context of an organization.
business policy	**business policy**	A business policy is a directive whose purpose is to govern or guide the company. A business policy serves as the basis for defining business rules and governing corporate processes. A business policy is always under the control of the company. It allows to control, guide and formalize the strategies and tactics of the company.

Projects

UAF Concept	HOPEX Concept	Definition
project	**project**	A project is a temporary endeavor undertaken by a specific team, to create a unique product, service or result. Its serves a purpose which can be expressed in terms of capability that is acquired (new), extended (improvement) or decommissioned (rationalization).
projects portfolio	**projects portfolio**	A projects portfolio gathers all the projects of a given Project Domain in the defined project state (Demands, Candidate Projects, Ongoing Projects). In a Portfolio, Projects can be assessed and compared against Projects Criteria (derived from projects information or project qualitative assessment) or Portfolio criteria (additional criteria defined at portfolio level).
project domain	**project domain**	A project domain gathers all the projects managed from the same perspective in terms of portfolio management decisions. Project Portfolios are therefore also attached to Project Domains; by default, one portfolio of each portfolio type is created when creating a new Project Domain.
life cycle	**life cycle**	A life cycle is represented by all the states an abject can take over time, as well as the transitions enabling change of state.
object life	**object life**	An object life is a set of time periods representing the updated calendar of object life cycle states.
project milestone	**project milestone**	A project milestone defines an intermediate delivery step in the life cycle of the project life. A project deliverable can be associated with a project milestone if it is delivered during the project and on the project date.

Standards

UAF Concept	HOPEX Concept	Definition
connexion type	**connexion type**	A connexion type defines a local use of a standard network application protocol (protocol and port number) which can be override and completed by source port number restriction and transport protocol (TCP, UDP).
standard	**standard**	A standard is a definition or format that has been approved by a recognized standards organization or is accepted as a de facto standard by the industry.
network application protocol	**network application protocol**	A network application protocols supported by a communication port must be compatible with the protocols supported by communication ports to which they are connected.
network protocol	**network protocol**	A network protocol is a set of standardized rules for transmission of information (voice, data, images) on a communication channel. The different layers of protocols can handle the detection and processing of errors, authentication of correspondents, management of routing.