HOPEX ArchiMate

User Guide

HOPEX V2R1

CONTENTS

INTRODUCTION

HOPEX ArchiMate is a full-web implementation of the Open Group's ArchiMate 3.0.1 Enterprise Architecture standard http://www.opengroup.org/archimate/.

HOPEX ArchiMate provides a metamodel and a notation covering all the ArchiMate standard Enterprise Architecture layers: Motivation, Strategy, Business, Application, Technology, Physical, and Implementation & Migration.

HOPEX ArchiMate implementation also supports the viewpoints mechanism, so that the ArchiMate diagram contents can be filtered according to a defined perspective; additional viewpoints can be defined.

The points covered in **HOPEX ArchiMate**:

- ✓ "The HOPEX Implementation of ArchiMate", page 15;
- ✓ "HOPEX ArchiMate ViewPoints", page 51;
- ✓ "The HOPEX implementation for ArchiMate", page 69.

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

- ✓ "Presentation of HOPEX ArchiMate", page 6,
- ✓ "HOPEX ArchiMate Desktop", page 8,
- ✓ "About This Guide", page 12.

PRESENTATION OF HOPEX ARCHIMATE

HOPEX ArchiMate features Summary

HOPEX ArchiMate provides:

- All the concepts defined by the ArchiMate 3.0.1 Open Group Standard.
- One diagram enabling to describe an ArchiMate Model View, containing all the possible concepts and relationships. This diagram contents can be filtered according to user (or administrator) defined viewpoints.

  ☛ *The main viewpoints samples described in the standard are provided out of the box, additional ones can be defined by the administrator. For more details see: "Updating the ArchiMate diagram type setup", page 91.*

- A bridge between some ArchiMate concepts and equivalent **HOPEX** concepts, enabling compatibility and continuity with other **HOPEX** products. As a consequence, inventories can be shared with other **HOPEX** products and across ArchiMate Models.
  An ArchiMate object identified in an ArchiMate diagram, for example an ArchiMate Business Process, can be detailed as an **HOPEX** object in another diagram using the BPMN notation, for example a **HOPEX Business Process Analysis** Organizational Process diagram.

HOPEX ArchiMate implementation

The **HOPEX ArchiMate** metamodel implementation comes with its own ontology for ArchiMate in order to match, as closely as possible, with the ArchiMate language superstructure, as defined in the standard.

☛ *For more details on **HOPEX ArchiMate** implementation, see "The HOPEX MetaModel for ArchiMate", page 70.*

Pre-Requisites to HOPEX ArchiMate

If you want to use **HOPEX ArchiMate**, you need to import the **ArchiMate.exe** and **PPM.exe** Solution Packs

☛ *The Solution Packs that you want to import needs to be decompressed. To do that, connect to **HOPEX installation folder > Utilities > Solution Pack**, double-click the **Solution Pack** to extract it.*

To import the Solution Pack:

1. Using **HOPEX Administration.exe**, connect to the environment concerned.
2. Expand the **Repositories** folder.

3. Right-click the repository and select **Object Management > Import Solution Pack**.
   The Solution Pack Import dialog box appears.
4. Select the Solution Packs "ArchiMate.exe" and "PPM.exe".
5. Click **OK**.
   The Import dialog box displays import progress.
   The selected Solution Pack is imported into the repository.

HOPEX ARCHIMATE DESKTOP

Connecting to the solution

To connect to **HOPEX ArchiMate**, see **HOPEX Common Features**, "HOPEX Web Front-End Desktop".

HOPEX ArchiMate Profiles

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

☛ *For more details on using the Web platform for **HOPEX** solutions, see the **HOPEX Common Features** guide.*

In **HOPEX ArchiMate**, there are, by default, profiles with which specific activities are associated.

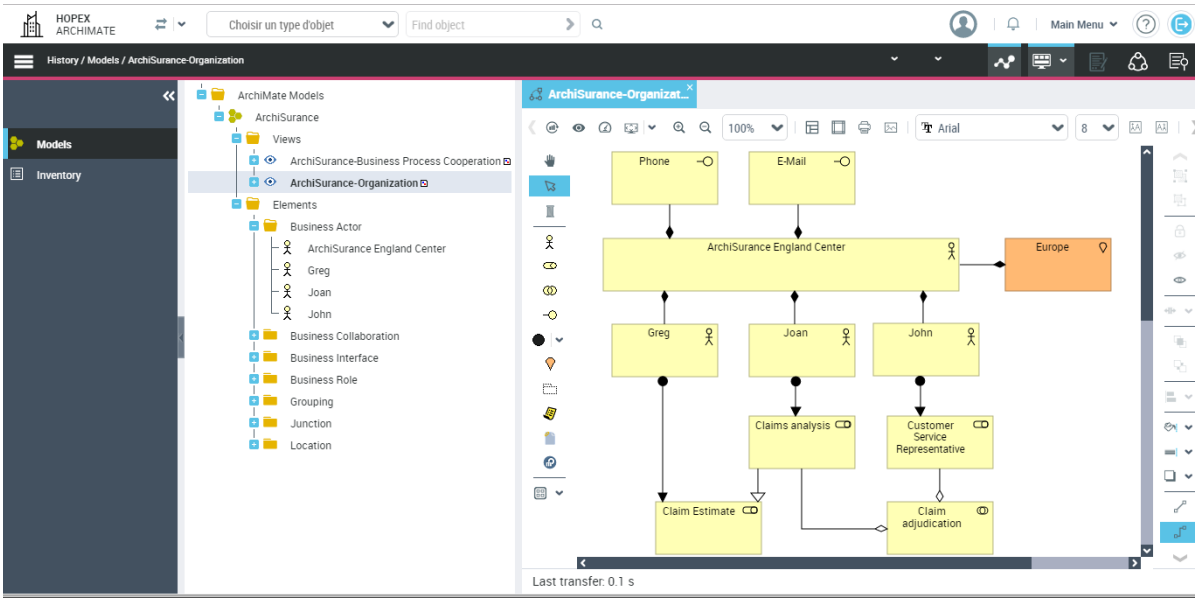
Profiles	Tasks
ArchiMate Functional Administrator	The ArchiMate Functional Administrator can create the working environments from Enterprises, assign the users, and define the necessary Viewpoints. For more details, see "Presenting the ArchiMate Functional Administrator workspace", page 10.
ArchiMate Enterprise Architect	The ArchiMate Enterprise Architect is the end user profile, entitled to create Models and View Diagrams according to the defined viewpoints. For more details, see "Presenting the ArchiMate Enterprise Architect workspace", page 9.

HOPEX ArchiMate Desktop Presentation

HOPEX ArchiMate provides its own Working Environment Template desktop, which provides:

- Access to ArchiMate Models and their elements in treeview (see **Models > ArchiMate Model Tree**),
- Access to ArchiMate EA Elements inventories,

  ☛ *To access to the shared inventories and relationships, select **Models > Inventories**,*
- Specific property pages for ArchiMate Elements (on the right side of the screen).



HOPEX ArchiMate Desktop

Presenting the ArchiMate Enterprise Architect workspace

The **ArchiMate Enterprise Architect** creates Enterprise Architectures Models.

The ArchiMate Enterprise Architect has access to the following panes:

- **Home** and **To-Do-List** that are common to all **HOPEX** solution users.
- **Reports**: produces access to all reports, improving understanding of models.
- **Models** that provides access to ArchiMate objects and ViewPoints.

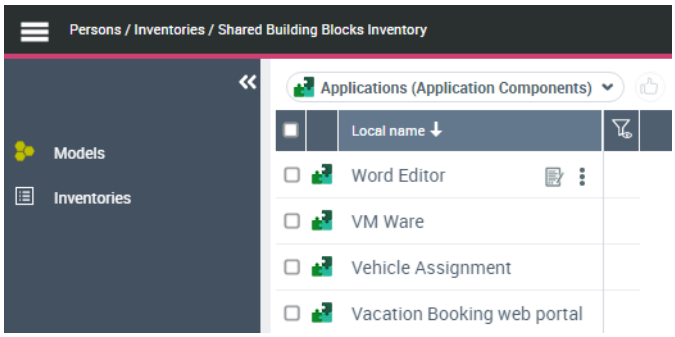
The Models pane

The **Models** pane provides access to the following menus:

- **Models**, to access the ArchiMate model, its views and the object used.

  ☛ *For more information on an ArchiMate model creation, see "Starting with HOPEX ArchiMate", page 33.*

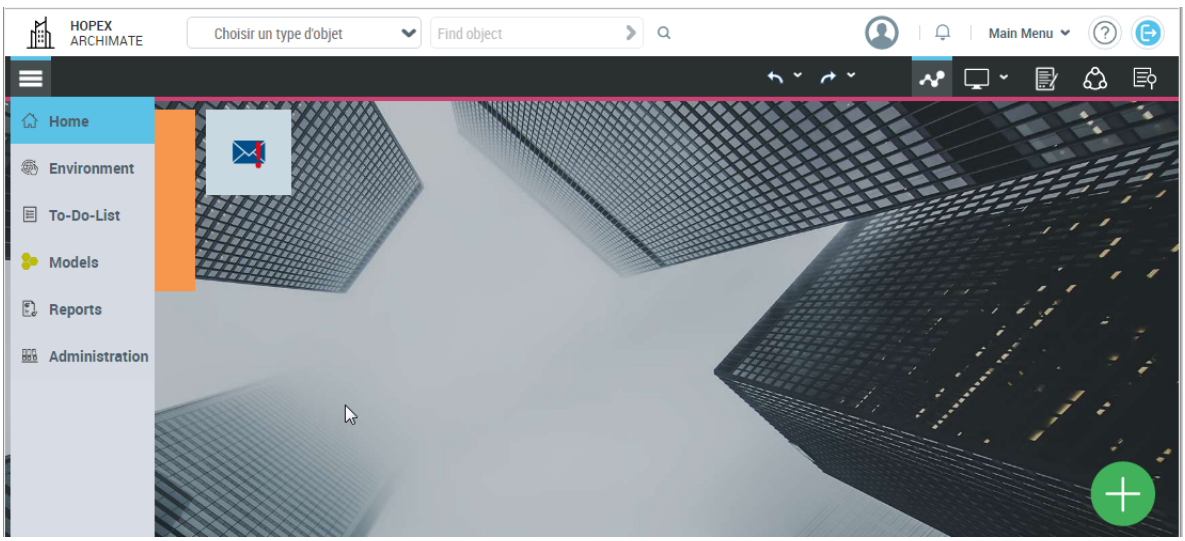
- **Inventories**, to access the shared objects inventories.
  The shared objects can be shared across ArchiMate Models and with other **HOPEX** products. For example, **Applications** inventory is shared with **HOPEX IT Portfolio Management** and can be used by **Application Components** in an ArchiMate Model.



Presenting the ArchiMate Functional Administrator workspace

The **ArchiMate Functional Administrator** has rights on all objects and Architectures. This profile prepares the work environment and creates elements required for management of projects. He manages:

- Enterprises.
- Users, assignment of profiles and access rights.



The Environment pane

In addition, the **ArchiMate Functional Administrator** profile, he has access to the **Environment** pane to create enterprises and allocate users to working environments.

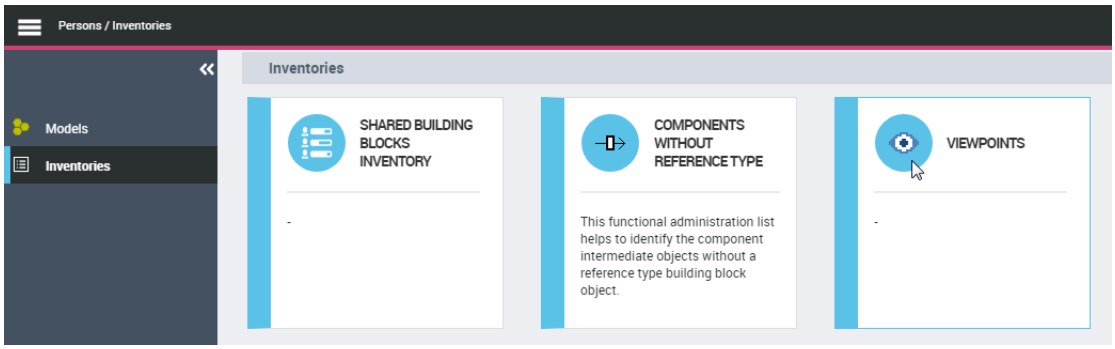
The **Environment** pane provides access to the following menu:

- **Standard Navigation**, to access the management functionalities for libraries and environments.

> ☛ *For more information on libraries and management, see the "Enterprise and Librairies" section in the **HOPEX Common Features** guide.*
>
> ☛

The Models pane

In addition, the **ArchiMate Functional Administrator** profile, he has access to **Viewpoints** tile of the **Inventories** menu to define viewpoints selecting which MetaClasses (Concepts: Element or Relationships) are available on the viewpoint.



> ☛ *For more details on viewpoints properties, see "The properties of a viewpoint", page 41.*

About This Guide

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

Guide Structure

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

- "The HOPEX Implementation of ArchiMate", page 15: describes the main principles that govern the ArchiMate implementation of **HOPEX ArchiMate**.
- "HOPEX ArchiMate ViewPoints", page 51: presents the functions offered by **HOPEX ArchiMate** to model the aspects of your enterprise architecture using the ArchiMate formalism.
- "The HOPEX implementation for ArchiMate", page 69: describes the **HOPEX** metamodel used to implement **HOPEX ArchiMate**.

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**.

THE HOPEX IMPLEMENTATION OF ARCHIMATE

HOPEX proposes an implementation based on the ArchiMate 3.0.1 specification which aims at ensuring continuity with other **HOPEX** products such as:

- ✓ **HOPEX Business Process Analysis** for the business layer,
- ✓ **HOPEX IT Architecture**, for the application, technology and implementation layers.

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 ArchiMate deliverables.

- ✓ "ArchiMate Layers and Relationships", page 16,
- ✓ "Starting with HOPEX ArchiMate", page 33,
- ✓ "Using HOPEX ArchiMate diagrams", page 40,
- ✓ "Using HOPEX ArchiMate reports", page 46.

ARCHIMATE LAYERS AND RELATIONSHIPS

This chapter provides definition and illustration of the generic set of concepts of ArchMate 3.0.1. They provide a proper basis for visualization, analysis, tooling, and use of these concepts.

The ArchiMate language defines three main layers, based on specializations of the core concepts:

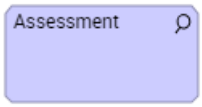
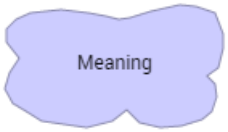
- The ***Business Layer*** offers products and services to external customers, which are realized in the organization by business processes performed by business actors. See "ArchiMate Business Layer Elements", page 20.
- The ***Application Layer*** supports the business layer with application services which are realized by (software) applications. See "ArchiMate Application Layer Elements", page 23.
- The ***Technology Layer*** offers infrastructure services (e.g., processing, storage, and communication services) needed to run applications, realized by computer and communication hardware and system software. See "ArchiMate Technology Layer Elements", page 25.

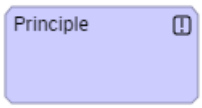
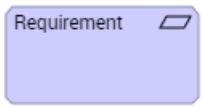
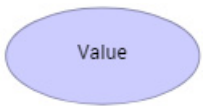
The other ArchiMate Layers are:

- The Motivation Layer, see "ArchiMate Motivation Layer Elements", page 17,
- The Strategy Layer, see "ArchiMate Strategy Layer Elements", page 19,
- The Physical Layer, see "ArchiMate Physical Layer Elements", page 27,
- The Implementation & Migration Layer, see "ArchiMate Implementation & Migration Layer Elements", page 28.

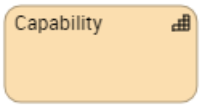
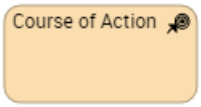
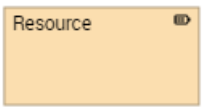
Examples of use of the elements included in each layer is described in "HOPEX ArchiMate ViewPoints", page 51.

ArchiMate Motivation Layer Elements

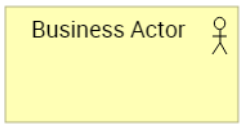
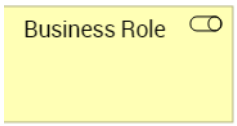
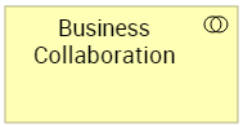
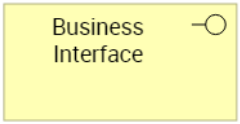
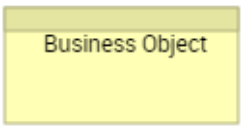
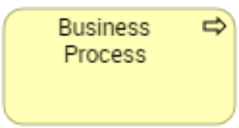
Concept Name	Notation	Comment
Assessment		An assessment represents the result of an analysis of the state of affairs of the enterprise with respect to some driver.
Constraint		A constraint represents a factor that prevents or obstructs the realization of goal.
Driver		A driver represents an external or internal condition that motivates an organization to define its goals and implement the changes necessary to achieve them.
Goal		A goal represents a high-level statement of intent, direction, or desired end state for an (organization and its stakeholders.
Meaning		Meaning represents the knowledge or expertise present in, or the interpretation given to, a core element in a particular context.
Outcome		An outcome represents an end result that has been achieved.

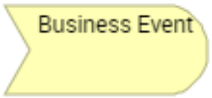
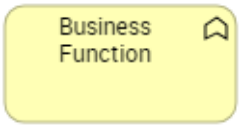
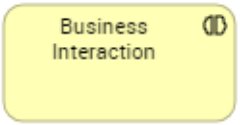
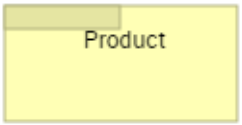
Concept Name	Notation	Comment
Principle		A principle represents a qualitative statement of intent that should be met by the architecture
Requirement		A requirement represents a statement of need that must be met by the architecture.
Stakeholder		A stakeholder is the role of an individual, team, or organization (or classes thereof) that represents their interests in the outcome of the architecture.
Value		Value represents the relative worth, utility, or importance of a core element or an outcome

ArchiMate Strategy Layer Elements

Concept Name	Notation	Comment
Capability		A capability represents an ability that an active structure element, such as an organization, person, or system, possesses.
Course of Action		A course of action is an approach or plan for configuring some capabilities and resources of the enterprise, undertaken to achieve a goal.
Resource		A resource represents an asset owned or controlled by an individual or organization.

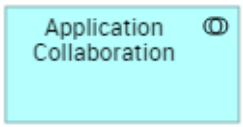
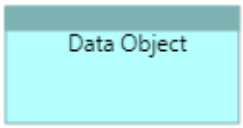
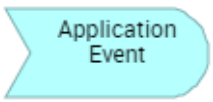
ArchiMate Business Layer Elements

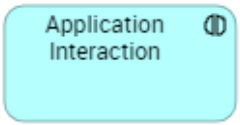
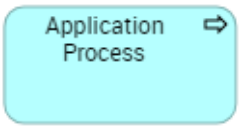
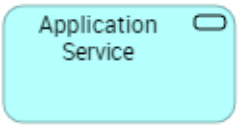
Concept Name	Notation	Comment
Business Actor		A business actor is a business entity that is capable of performing behavior.
Business Role		A business role is the responsibility for performing specific behavior, to which an actor can be assigned, or the part an actor plays in a particular action or event.
Business Collaboration		A business collaboration is an aggregate of two or more business internal active structure elements that work together to perform collective behavior.
Business Interface		A business interface is a point of access where a business service is made available to the environment.
Business Object		A business object represents a concept used within a particular business domain.
Business Process		A business process is defined as a unit of internal behavior or collection of causally related units of internal behavior intended to produce a defined set of products and services.

Concept Name	Notation	Comment
Business Event		A business event is a business behavior element that denotes an organizational state change. It may originate from and be resolved inside or outside the organization.
Business Function		A business function is a collection of business behavior based on a chosen set of criteria (typically required business resources and/or competencies), closely aligned to an organization, but not necessarily explicitly governed by the organization.
Business Interaction		Business interaction is defined as a unit of behavior performed by a collaboration of two or more business roles.
Business Service		A business service represents an explicitly defined exposed business behavior.
Product		A product represents a coherent collection of services and/or passive structure elements, accompanied by a contract/set of agreements, which is offered as a whole to (internal or external) customers.

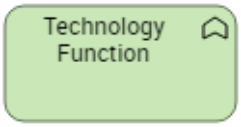
Concept Name	Notation	Comment
Representation		A representation represents a perceptible form of the information carried by a business object.
Contract		A contract represents a formal or informal specification of an agreement between a provider and a consumer that specifies the rights and obligations associated with a product and establishes functional and non-functional parameters for interaction.
Location		A location is a place or position where structure elements can be located or behavior can be performed.

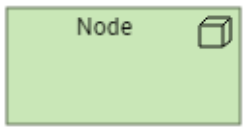
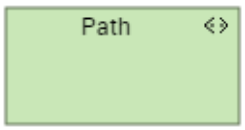
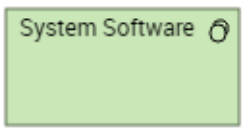
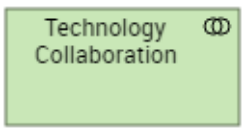
ArchiMate Application Layer Elements

Concept Name	Notation	Comment
Application Collaboration		An application collaboration represents an aggregate of two or more application components that work together to perform collective application behavior.
Application Component		An application component represents an encapsulation of application functionality aligned to implementation structure, which is modular and replaceable. It encapsulates its behavior and data, exposes services, and makes them available through interfaces.
Application Interface		An application interface represents a point of access where application services are made available to a user, another application component, or a node.
Data Object		A data object represents data structured for automated processing.
Application Event		An application event is an application behavior element that denotes a state change.
Application Function		An application function represents automated behavior that can be performed by an application component.

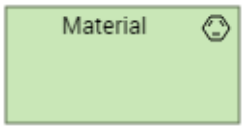
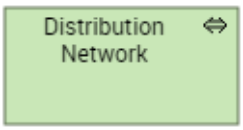
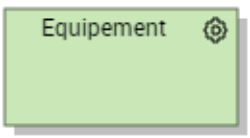
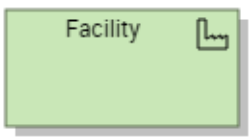
Concept Name	Notation	Comment
Application Interaction		An application interaction represents a unit of collective application behavior performed by (a collaboration of) two or more application components.
Application Process		An application process represents a sequence of application behaviors that achieves a specific outcome.
Application Service		An application service represents an explicitly defined exposed application behavior.

ArchiMate Technology Layer Elements

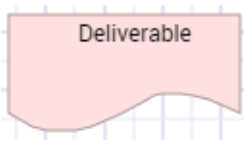
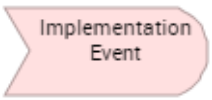
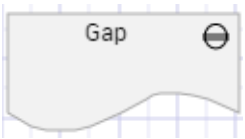
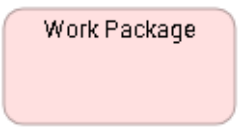
Concept Name	Notation	Comment
Artifact		An artifact represents a piece of data that is used or produced in a software development process or by deployment and operation of an IT system.
Technology Event		A technology event is a technology behavior element that denotes a state change.
Technology Function		A technology function represents a collection of technology behavior that can be performed by a node.
Technology Interaction		A technology interaction represents a unit of collective technology behavior performed by (a collaboration of) two or more nodes.
Technology Process		A technology process represents a sequence of technology behaviors that achieves a specific outcome.
Technology Service		A technology service represents an explicitly defined exposed technology behavior.

Concept Name	Notation	Comment
Communication Network		A communication network represents a set of structures that connects computer systems or other electronic devices for transmission, routing, and reception of data or data-based communications such as voice and video.
Device		A device is a physical IT resource upon which system software and artifacts may be stored or deployed for execution.
Node		A node represents a computational or physical resource that hosts, manipulates, or interacts with other computational or physical resources.
Path		A path represents a link between two or more nodes, through which these nodes can exchange data or material.
System Software		System software represents software that provides or contributes to an environment for storing, executing, and using software or data deployed within it.
Technology Collaboration		A technology collaboration represents an aggregate of two or more nodes that work together to perform collective technology behavior.
Technology Interface		A technology interface represents a point of access where technology services offered by a node can be accessed.

ArchiMate Physical Layer Elements

Concept Name	Notation	Comment
Material		A material represents tangible physical matter or physical elements.
Distribution Network		A distribution network represents a physical network used to transport materials or energy.
Equipment		An equipment represents one or more physical machines, tools, or instruments that can create, use, store, move, or transform materials.
Facility		A facility represents a physical structure or environment.

ArchiMate Implementation & Migration Layer Elements

Concept Name	Notation	Comment
Deliverable		A deliverable represents a precisely-defined outcome of a work package
Implementation Event		An implementation event is a behavior element that denotes a state change related to implementation or migration.
Gap		A gap represents a statement of difference between two plateaus.
Plateau		A plateau represents a relatively stable state of the architecture that exists during a limited period of time.
Work Package		A work package represents a series of actions identified and designed to achieve specific results within specified time and resource constraints.

Other ArchiMate Elements

Concept Name	Notation	Comment
Grouping		The grouping element aggregates or composes concepts that belong together based on some common characteristic.
Junction		A junction is used to connect relationships of the same type.

ArchiMate Relationships

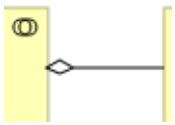
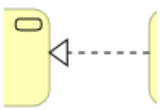
ArchiMate defines eleven types of relationships sorted out into four categories. Each relationship type has its own representation.

The relationships categories are:

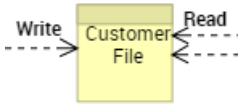
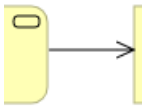
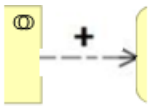
- "Structural Relationships", page 30,
- "Dependency Relationships", page 31,
- "Dynamic Relationships", page 31,
- "Other Relationships", page 32.

☛ *For more details on the use of ArchiMate Relationships in diagrams, see "Creating an ArchiMate Relationship", page 38.*

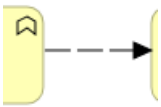
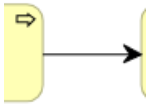
Structural Relationships

Concept Name	Notation	Comment
Composition		The composition relationship indicates that an element consists of one or more other concepts. The diamond indicates the owner element.
Assignment		The assignment expresses the allocation of responsibility, performance of behavior, or execution.
Aggregation		The aggregation relationship indicates that an element consists of one or more other concepts. The diamond indicates the main element.
Realization		The realization relationship indicates that an entity plays a critical role in the creation, achievement, sustenance, or operation of a more abstract entity. The arrow indicates the entity playing a role.

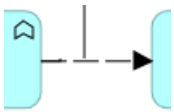
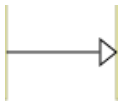
Dependency Relationships

Concept Name	Notation	Comment
Access		The access relationship models the ability of behavior and active structure elements to observe or act upon passive structure elements.
Serving		The serving relationship models that an element provides its functionality to another element. The arrow indicates the functionality user element.
Influence		The influence relationship models that an element affects the implementation or achievement of some motivation element. The arrow indicates the motivation element

Dynamic Relationships

Concept Name	Notation	Comment
Flow		The flow relationship describes the exchange or transfer of, for example, information or value between processes, function, interactions, and events
Trigger		The triggering relationship describes the temporal or causal relations between processes, functions, interactions, and events.

Other Relationships

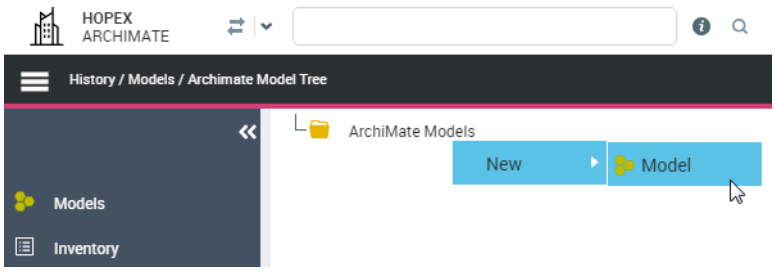
Concept Name	Notation	Comment
Association		Association is a specific Relationship which can associate any concepts (including other Relationships.
Specialization		The specialization relationship indicates that an element is a particular kind of another element. The arrow points to the particular element.

STARTING WITH HOPEX ARCHIMATE

Creating an ArchiMate Model

To create an ArchiMate model:

1. In the **Models > Models** navigation pane, select the **ArchiMate Model Tree** tile.
2. Right click the **ArchiMate Models** folder and select **New > Model**.



The **Creation Model** dialog box appears.

3. In the **Name** box, enter "ArchiSurance" and click **OK**.

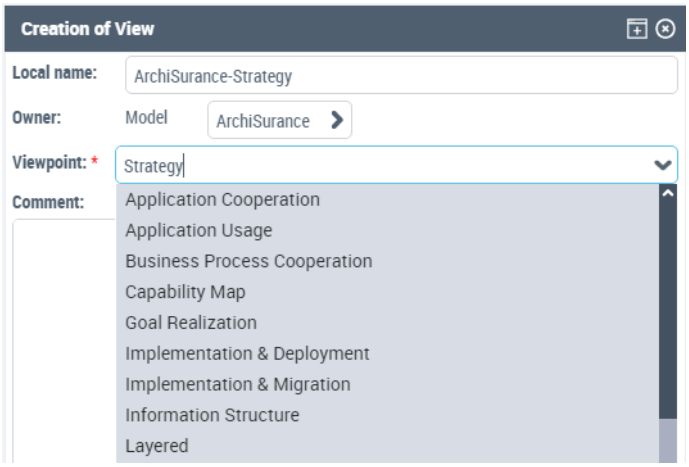
> ☛ *The ArchiMate model is the root object in ArchiMate and defines a namespace for ArchiMate Elements*

Creating an ArchiMate View

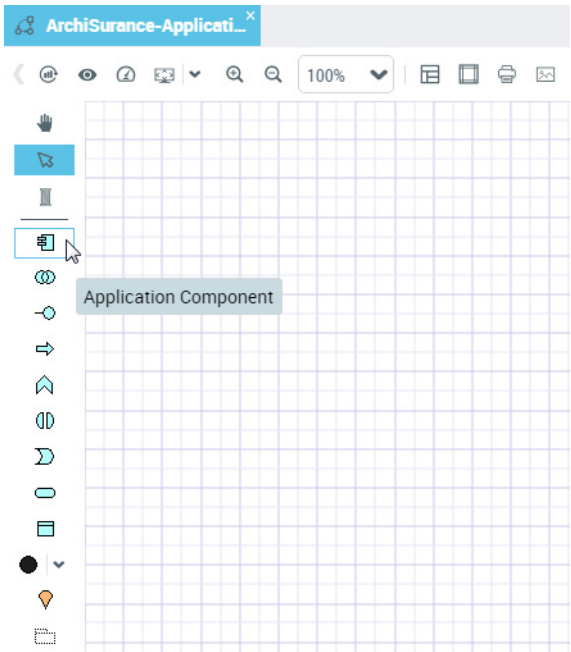
To create an ArchiMate View:

1. In the **Models > Models** navigation pane, select the **ArchiMate Model Tree** tile.
2. Expand the **ArchiMate Models** folder.

3. Right click your ArchiMate Model and select **New > View**.
The **Creation of View** dialog box appears.



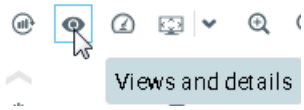
4. In the **Viewpoint** field, select the viewpoint that interest you and click **OK**.
The **Name** of the new view is automatically computed and can be modified.
The view diagram opens in the edition area.



The diagram views are activated based on the selected Viewpoint setup. Additional views can be manually activated, or default views can be deactivated.

To activate the views window:

1. In a diagram, click **Views and details** button.



The list of object types to be displayed appears.

2. Select (or clear) the object types you wish to display (or not).

> ☛ *For more details on object types available in a viewpoint, see "The properties of a viewpoint", page 41.*

Creating an ArchiMate Element in a diagram view

ArchiMate Standalone Elements have a plain creation wizard where only the **Name** is requested.

> ☛ *For more details, see "Creating ArchiMate Standalone Elements", page 35.*

ArchiMate EA Elements are based on shared inventory objects. Those elements have a specific creation behavior with a dedicated wizard enabling the reuse or the creation of shared object.

> ☛ *For more details, see "Creating an ArchiMate EA Element", page 36.*

Creating ArchiMate Standalone Elements

ArchiMate **Application Functions** are ***Standalone Elements***.

> ☛ *For more details on ArchiMate Standalone Elements implementation in **HOPEX**, see "ArchiMate Elements", page 75.*

Creating an ArchiMate Standalone Element

To create an **Application Function**, for example:

1. In the insert toolbar, click the **Application Function** button.
2. Click in the diagram.
   The **Creation of Application Function** dialog box appears.
3. Enter the **Name** of the Application Function.
4. Click **OK**.
   The Application Function appears in the diagram.

Creating several ArchiMate Standalone Elements

To create the other Application Functions:

1. Double-click the **Application Function** button and then click in the diagram
   The new Application Function appears in the diagram.
2. Press key <Esc> to stop the **Application Functions** creation.
3. To rename the Application Functions, click the name of the Application Function, press key <F2> and enter a new name.

Reusing an ArchiMate Standalone Element

To reuse an existing ArchiMate Standalone Element, you must use the navigation tree.

For example, to reuse an existing **Business Service**:

1. In the navigation pane **Models**, expand your ArchiMate Model folder.
2. Expand the **Elements** folder and the **Business Service** folder.
3. Click the Business Service that interest you and, holding down the mouse button, drag the cursor to the in the diagram and release the mouse button.
   The corresponding **Business Service** appears in the diagram.

> **You can reuse only the Standard Elements of the current model. Reusing across ArchiMate models is not permitted.**

Creating an ArchiMate EA Element

ArchiMate EA Elements have a specific creation behavior with a dedicated wizard enabling the reuse or the creation of shared inventory objects.

> *For more details on ArchiMate EA Elements implementation in **HOPEX**, see "ArchiMate Elements", page 75.*

> *For more details on the ArchiMate Elements in **HOPEX**, see "Concepts mapping", page 99.*

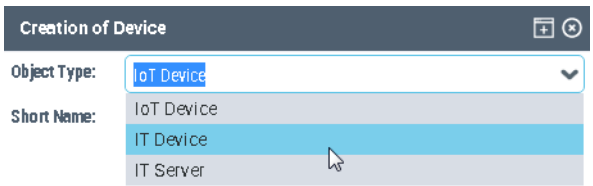
With **HOPEX ArchiMate**, two categories of mapping are considered:

- ***Simple mapping***: the existing inventory objects are proposed in a drop-down list with autocompletion.
- ***Complex mapping***: several inventory object subtypes are possible and must be chosen when creating a new object. For example, creating a new **Device** will request to choose whether the new device is an **IT Server**, an **IT Device** or an **IoT Devices**, in order to be consistent with the **HOPEX IT Architecture** solution infrastructure MetaModel.

Creating a shared inventory object

To create a **Device**, for example:

1. In the insert toolbar, click the **Device** button.
2. Click in the diagram.
   The **Creation of Device** dialog box appears.



3. Select the type of the Device.
4. Enter the **Name** of the New Device.

5. Click **OK**.
   The device appears in the diagram.

> ☛ *For simple mapping EA Elements, you have to enter the new **Name** and clicking **Next** to create a new inventory objects.*

Depending on the type of the created **ArchiMate EA Element**, more information may be requested in a specific wizard. For example:

- **Business Object** is mapped to the **Concept** inventory, so creation of a new concept will display the concept creation wizard, suggesting existing terms, for instance.
- **Work Package** is mapped to the **Enterprise Project** inventory, so creation of a new **Work Package** will display the **Enterprise Project** creation wizard, requesting the following additional information:
  - The current project state: demand, candidate or ongoing,
  - The Project Domain in which the project is defined, for example Business, IT or R&D projects,
  - The project planned dates.

> ☛ *The **HOPEX IT Portfolio Management** solution pack must be imported in order to enable **Work Package** creation*

Creating a shared inventory object using the navigation tree

You can also create ArchiMate Elements from the navigation tree.

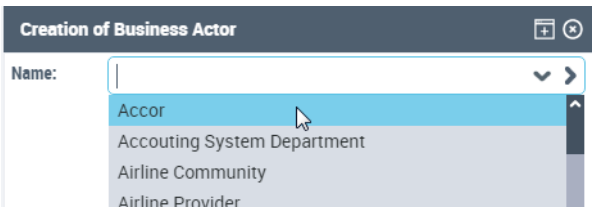
For example, to create a **Device** from the navigation tree:

1. In the navigation pane **Models**, expand your ArchiMate Model folder.
2. Expand the **Elements** folder.
3. Right-click on the **Device** folder and select **New**.
   The **Creation of Device** dialog box appears.

Reusing a shared inventory object using the insertion toolbar

To reuse a **Business Actor**, for example:

1. In the insert toolbar, click the **Business Actor** button.
2. Click in the diagram.
   The **Creation of Business Actor** dialog box appears.
3. Click the **Name** down arrow.
   Several inventory object subtypes are possible: **Org-Unit** or **Position Type** elements must be chosen.



> ☛ *The shared inventory object is added in the **Elements** folder of the ArchiMate Model.*

Reusing an element based on a shared inventory object using the navigation tree

You can also use the navigation tree to reuse an existing **Org Unit**, for example:

1. In the navigation pane **Models**, expand your ArchiMate Model folder.
2. Expand the **Elements** folder and the **Business Actor** folder.
3. Click the **Business Actor** that interests you and, holding down the mouse button, drag the cursor to the in the diagram and release the mouse button.
   The corresponding **Business Actor** appears in the diagram.

   ☛ *The shared inventory object is reused but a new **EA Element** is created.*

Creating an ArchiMate Relationship

All ArchiMate relationships are created using the **Link** button  available in the toolbar of any diagram view.

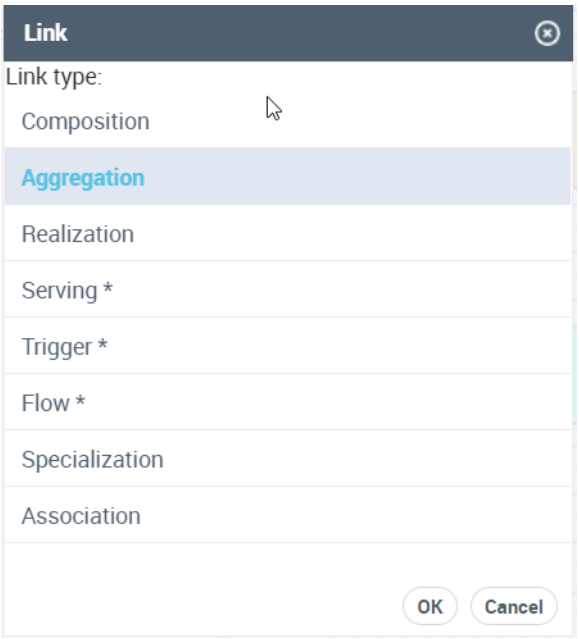
When creating a relationship, a dialog box displays the list of possible relationship types available between the two elements in the current viewpoint.

☛ *For more details on the possible ArchiMate Relationship types, see "ArchiMate Relationships", page 29.*

To create an ArchiMate Relationship in a diagram view:

1. In the insert toolbar, click the **Link** button  .
2. Click an ArchiMate Element and, holding down the mouse button, drag the cursor to the ArchiMate Element to be connected and release the mouse button.
   The two ArchiMate Elements are highlighted and a dotted line indicates the path that will be taken by the graphic link.

3. In the **Link** a dialog box, select the relationship type you want to create.



☛ *Indirect relationships appear with a star suffix: **Serving ***, for example, and are displayed in grey in the diagram. For more details on Indirect Relationship, see "ArchiMate Relationships MetaModel", page 73.*

USING HOPEX ARCHIMATE DIAGRAMS

Using Libraries

Libraries are collections of objects used to split HOPEX 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.

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

To access the list of libraries from the **Environment** navigation pane:

1. Click **Enterprises > Standard Navigation** the navigation menu.
2. Click **Structure View** tile.
   The repository tree appears. The library tree appears underneath.

Models, Views and Viewpoints

HOPEX ArchiMate supports the viewpoints mechanism defined by the ArchiMate standard:

- Viewpoints define which concepts (elements or relationships) are useful in a particular type of view,
- Views can be created after a given viewpoint within an ArchiMate Model.

The diagram views of the diagram describing a view are initialized according to the viewpoint definition; it is then possible for the user to restrict / extend this selection to hide or add necessary additional concepts.

The diagram views are defined based on the generic metamodel and layers (e.g. Business Layer Active Structures activates 'Business Actor', 'Business Collaboration', 'Business Role' and 'Business Interface' concepts).

☛ *The number of diagram views being limited in **HOPEX**, the additional layers (motivation, implementation & migration, strategy) diagrams views are more limited so that requiring one concept (e.g. outcome) in one viewpoint will activate the whole view (e.g. motivation), so that additional concepts are also made available (e.g. value, meaning)*

One diagram, several views

HOPEX ArchiMate implementation provides a single diagram type containing all the possible elements and relationships defined in ArchiMate 3.0.1.

The 'layered' viewpoint provides access to all the concepts.

- Direct relationships appear in black,
- Indirect relationships appear in dark grey.

Customizing viewpoints

Viewpoints can be defined by the **ArchiMate Functional Administrator** by selecting the MetaClasses (concepts) which are included in the viewpoint. The MetaClasses (concepts) can be Elements and Relationships.

Note that abstractions can be used in this definition to ease the setup:

- e.g. the 'layered' viewpoint only contains 'ArchiMate Concept' root abstract MetaClass, thus enabling access to all concepts
- 'ArchiMate Flow Relationship' generic relationship can be selected to add all kind of flow relationships in the viewpoint's views diagram.

☛ *For more details on viewpoints access, see "Creating or Modifying a Viewpoint", page 83.*

Consulting HOPEX ArchiMate Property pages

HOPEX ArchiMate provides information about each object in the **property pages** associated to the object.

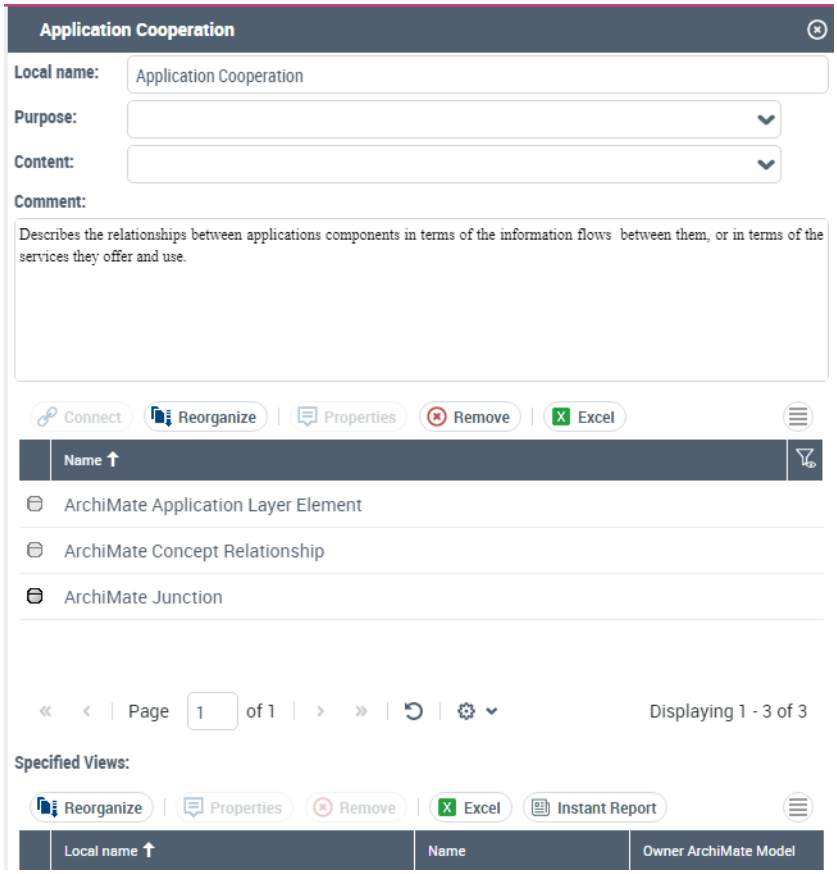
The properties of a viewpoint

The **ArchiMate Functional Administrator** can define viewpoints selecting which MetaClasses (Concepts: Element or Relationships) are available on the viewpoint.

☛ *For more details on **HOPEX ArchiMate** viewpoints customization, see "Creating or Modifying a Viewpoint", page 83.*

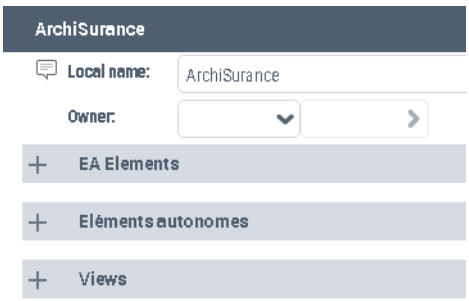
The viewpoint property page provides access to the basic characteristics and contained elements:

- its **Name**,
- its **Purpose**, the possible values are:
  - Deciding
  - Designing
  - Informing
- its **Content**, the possible values are:
  - Coherence
  - Details
  - Overview
- the text of the viewpoint **Description**.
- the list of MetaClasses defining the type of ArchiMate objects available for the viewpoint,
- the list of **Specified Views**.


Application Cooperation
Local name:
Application Cooperation
Purpose:
Content:
Comment:
Describes the relationships between applications components in terms of the information flows between them, or in terms of the services they offer and use.
Connect
Reorganize
Properties
Remove
Excel
Name
ArchiMate Application Layer Element
ArchiMate Concept Relationship
ArchiMate Junction
Page
1
of 1
Displaying 1 - 3 of 3
Specified Views:
Reorganize
Properties
Remove
Excel
Instant Report
Local name
Name
Owner ArchiMate Model

The properties of a model

The model property page provides access to the basic characteristics and contained elements:



- its **Name**,
- its **Owner**, by default on creation of the model, the current Library or Enterprise.
- the list of its **EA Elements**,
- the list of its **Standalone Elements**,
- the list of its **Views**.

The properties of ArchiMate Elements

The **Characteristics** property page of an ArchiMate Element provides access to:

- its **Name**,
- its **Owner**, by default on creation of the ArchiMate Element, the current model.
- the text of its **Description**.

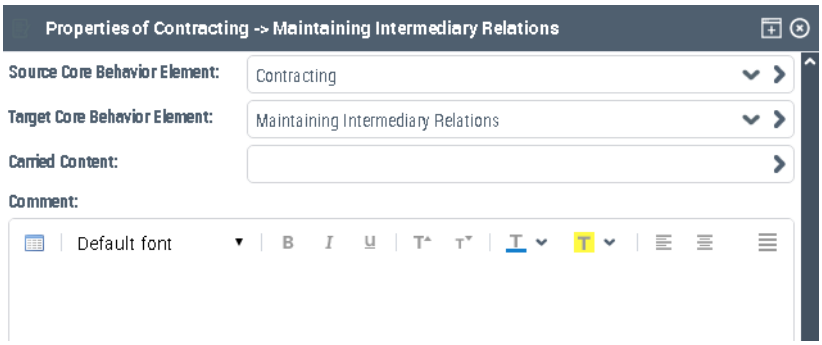
With **HOPEX ArchiMate**, an ArchiMate Element is described by the following pages:

- the **Outgoing Relationships** property page provides access to outgoing relationships of the various types presented in separate sections
  - Structural relationships,
  - Dependency relationships,
  - Dynamic relationships,
  - Other relationships.
- **Incoming Relationships** property page provides access to incoming relationships, **same as for outgoing.**

The properties of Relationships

The property page of an ArchiMate relationships provides access to:

- its **Source Element** or concept,
- its **Target Element** or concept,
- an attribute or additional object link, if any
  - influence type
  - access type
  - flow **Carried Contents**.
- the text of its **Description**.



The properties of Junction

The property page of an ArchiMate junction provides access to:

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

With **HOPEX ArchiMate**, an ArchiMate junction is described by the following pages:

- the **Outgoing Relationships** property page provides access to outgoing relationships of the various types presented in separate sections
  - Structural relationships,
  - Dependency relationships,
  - Dynamic relationships,
  - Other relationships.
- **Incoming Relationships** property page provides access to incoming relationships, **same as for outgoing.**

USING HOPEX ARCHIMATE REPORTS

HOPEX provides some reports specifically designed for ArchiMate.

Accessing HOPEX ArchiMate Reports

Several report templates are provided with **HOPEX ArchiMate**:

- Archimate - TOGAF / ArchiMate Stakeholder Map - Stakeholders Influence Matrix, see "Stakeholder Map Matrix", page 47.
- Archimate - TOGAF / ArchiMate Stakeholder Map - Stakeholders Associated Motivation Element Matrix, see "Stakeholder Map Matrix", page 47.
- Archimate - TOGAF / Capability x Active Structure Matrix (via Resources), see "Business Service / Function Catalog", page 48.
- Archimate - TOGAF / Capability x Active Structure Matrix (via Servicesx,see "Business Service / Function Catalog", page 48.
- Archimate - TOGAF / Service x Information, see "Business Service / Information Diagram", page 48.
- Archimate - TOGAF / Stakeholder / Driver / Goal / Requirement Catalog, see "Driver / Goal / Objective Catalog", page 48.
-
- ArchiMate Application Component Catalog,
- ArchiMate Capability X Active Structure Matrix (via Resoures),
- ArchiMate Capability X Active Structure Matrix (via Services),
- ArchiMate Stakeholder Map - Stakeholder X Associated Motivation Element Matrix,
- ArchiMate Stakeholder Map - Stakeholder Influence Matrix.

To access to **HOPEX ArchiMate** reports:

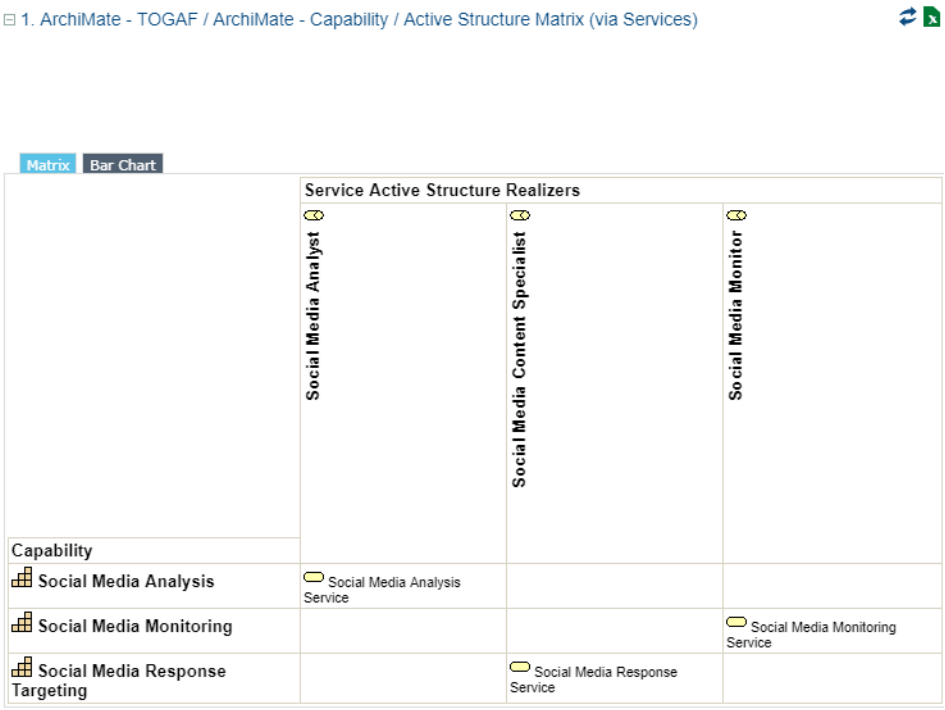
1. Click the navigation menu and select **Reports > Other Reports**.
2. Click **My Reports** tile.
   The list of existing reports is displayed.

Example of HOPEX ArchiMate Reports outputs

To use the Report Template "ArchiMate - TOGAF / ArchiMate Capability X Active Structure Matrix (via Services)" provided with **HOPEX ArchiMate** reports:

1. Click the navigation menu and select **Reports > Other Reports**.
2. Click **My Reports** tile.
3. Click **New** button.
4. The Creation of Report opens, click **Next**.
5. Select the Report Template "ArchiMate - TOGAF / ArchiMate Capability X Active Structure Matrix (via Services)".

6. Click **Next**.
7. In the **ArchiMate Capability List**, select the capabilities that interest you and click **Connect**.
   The report is dispalyed in the edition area.



Presentation of HOPEX ArchiMate Report Templates

The report templates provided with **HOPEX ArchiMate** are coming from the reports provided by TOGAF Standard.

The **HOPEX ArchiMate** report templates are presented below depending on the corresponding TOGAF artifact

Stakeholder Map Matrix

The purpose of the **Stakeholder Map matrix** is to identify the stakeholders for the architecture engagement, their influence over the engagement, and their key

questions, issues, or concerns that must be addressed by the architecture framework.

- ***Supported TOGAF ADM Phase***: Phase A - Architecture Vision
- ***Topic***: Motivation
- **HOPEX ArchiMate** report templates:
  - Archimate - TOGAF / ArchiMate Stakeholder Map - Stakeholders Influence Matrix
  - Archimate - TOGAF / ArchiMate Stakeholder Map - Stakeholders Associated Motivation Element Matrix

Business Service / Function Catalog

The purpose of the **Business Service/Function catalog** is to provide a functional decomposition in a form that can be filtered, reported on, and queried, as a supplement to graphical Functional Decomposition diagrams.

The **Business Service/Function catalog** can be used to identify capabilities of an organization and to understand the level that governance is applied to the functions of an organization. This functional decomposition can be used to identify new capabilities required to support business change or may be used to determine the scope of change initiatives, applications, or technology components

- ***Supported TOGAF ADM Phase***: Phase B: Business Architecture
- ***Topic***: Strategy /Architecture
- **HOPEX ArchiMate** report templates:
  - Archimate - TOGAF / Capability x Active Structure Matrix (via Resources)
  - Archimate - TOGAF / Capability x Active Structure Matrix (via Servicesx

Business Service / Information Diagram

During the Business Architecture phase, a **Business Service/Information diagram** was created showing the key data entities required by the main business services. This is a prerequisite to successful Data Architecture activities.

- ***Supported TOGAF ADM Phase***: Phase B: Business Architecture
- ***Topic***: Application and Data
- **HOPEX ArchiMate** report template :
  - Archimate - TOGAF / Service x Information

Driver / Goal / Objective Catalog

The purpose of the **Driver/Goal/Objective catalog** is to provide a cross-organizational reference of how an organization meets its drivers in practical terms through goals, objectives, and (optionally) measures..

- ***Supported TOGAF ADM Phase***: Phase B: Business Architecture
- ***Topic***: Application and Data
- **HOPEX ArchiMate** report template :
  - Archimate - TOGAF / Stakeholder / Driver / Goal / Requirement Catalog.

Data Entity / Business Function Matrix

During the Business Architecture phase, a **Business Service/Information diagram** was created showing the key data entities required by the main business services. This is a prerequisite to successful Data Architecture activities.

- ***Supported TOGAF ADM Phase***: Phase C: Data Architecture
- ***Topic***: Active Structure and Data
- **HOPEX ArchiMate** report template :
  - Archimate - TOGAF / Service x Information

Application/Technology Matrix

The **Application/Technology matrix** documents the mapping of applications to technology platform.

- ***Supported TOGAF ADM Phase***: Phase D: Technology Architecture
- ***Topic***: Application & infrastructure
- **HOPEX ArchiMate** report template :
  - Archimate - TOGAF / Application Components x Technology Nodes Matrix

Requirements Catalog

The **Requirements catalog** captures things that the enterprise needs to do to meet its objectives. Requirements can also be used as a quality assurance tool to ensure that a particular architecture is fit-for-purpose (i.e., can the architecture meet all identified requirements).

- ***Supported TOGAF ADM Phase***: Phase E: Opportunities and Solutions
- ***Topic***: Requirements Analysis
- **HOPEX ArchiMate** report template :
  - Archimate - TOGAF / Requirements Catalog

HOPEX ARCHIMATE VIEWPOINTS

ArchiMate is composed of a set of viewpoints, which address different parts of an enterprise architecture. This chapter presents the functions offered by **HOPEX ArchiMate** to model your enterprise through the diagrams proposed by ArchiMate.

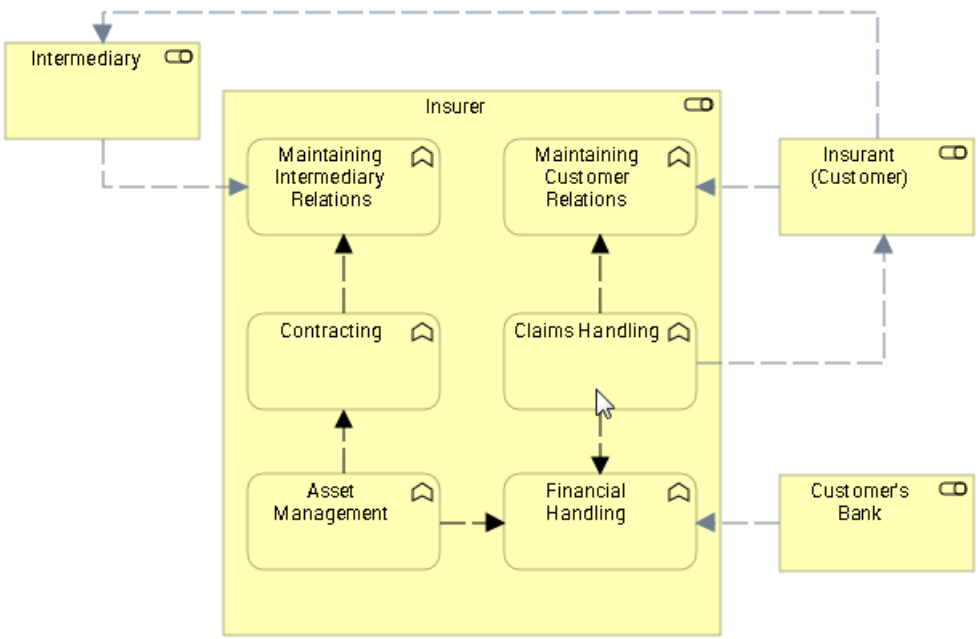
This presentation is based on the example of a fictional Insurance company which is used in the "ArchiMate Specification", "ArchiMate Language Primer" and "ArchiSurance business case" documents.

Below is the description of some of these viewpoints as outlined in the ArchiMate Specification documents. The Organization ViewPoint paragraph will help you to use **HOPEX ArchiMate**.

ARCHIMATE EXAMPLE CONTEXT PRESENTATION

"ArchiSurance" is a company that provides home and travel insurance services. The main business functions are:

- Maintaining Customer Relations and Intermediary Relations: these business functions are responsible for the contacts of the company with its customers and the intermediaries that sell its products. It handles customer questions and incoming claims, and performs marketing and sales.
- Contracting: this function handles the 'back-office' processing of contracts. It performs risk analysis and ensures legally and financially correct contracts.
- Claims Handling: this function is responsible for handling insurance claims.
- Financial Handling: this function performs the regular premium collection, according to the insurance policies with customers as produced by Contracting, and handles the payment of insurance claims.
- Asset Management: this function manages the financial assets of ArchiSurance, e.g. by investing in stocks and bonds.



Example of Business Function viewpoint diagram

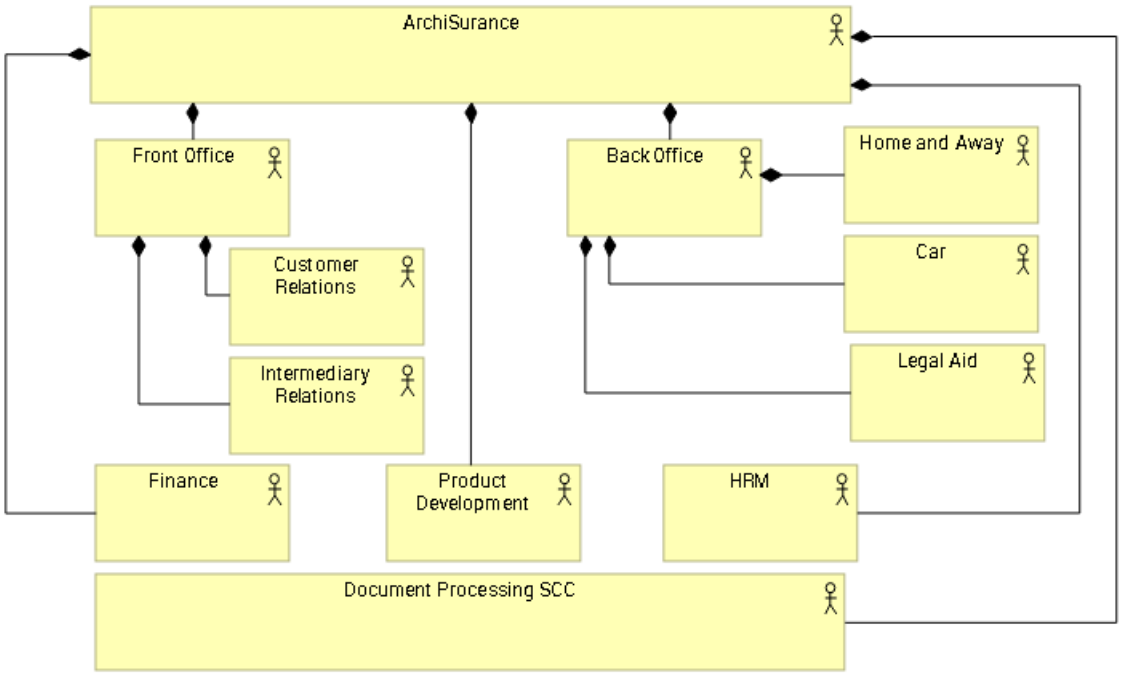
ORGANIZATION VIEWPOINT

The Organization viewpoint focuses on the (internal) organization of a company, a department, a network of companies, or of another organizational entity (i.e. a Business Actor). It is possible to present models in this viewpoint as organizational charts. The Organization viewpoint is very useful in identifying competencies, authority, and responsibilities in an organization.

With this viewpoint, you will learn how to describe your first viewpoint diagram.

Example Presentation

The diagram below describes the "ArchiSurance" organization.



Organization viewpoint diagram

"ArchiSurance" is a business actor composed by:

- A front office, comprising departments for managing relations with customers on the one hand, and intermediaries on the other hand.

Three separate back offices:

Home & Away: this department was the original pre-merger ArchiSurance, responsible for home and travel insurance.

Legal Aid: this is the old LegallyYours, responsible for legal aid and liability insurance.

Car: this department is the core of the old PRO-FIT and handles car insurance, including some legal aid.

Furthermore, ArchiSurance is in the process of setting up a Shared Service Center for document processing, which will handle all document streams and performs scanning, printing, and archiving job.

Creating an Organizational Viewpoint Diagram

In **HOPEX ArchiMate**, the "organizational viewpoint" is a materialized by diagram describing a Business Actor.

Creating an organizational View and its diagram

To create an the organizational view:

1. In the **Models > Models** navigation pane, select the **ArchiMate Model Tree** tile.
2. Expand the **ArchiMate Models** folder.
3. Right click your ArchiMate Model, "ArchiSurance" for example, and select **New > View**.
   The **Creation of View** dialog box appears.
4. In the **Viewpoint** field, select **Organizational** and click **OK**.
   The new view appears in the **Views** folder. Its **Name** is automatically computed, but it can be modified.
   The organizational view diagram opens in the edition area.

Creating Business Actors

You will define the Business Actors of the "ArchiSurance" organization structure.

To create a **Business Actor** in the organizational view diagram:

1. In the insert toolbar, click the **Business Actor** button.
2. Click in the diagram.
   The **Creation of Business Actor** dialog box appears.
3. Select the **Name** of the Business Actor, "ArchiSurance" for example.
4. Click **OK**.
   The Business Actor appears in the diagram.

Assigning Sub-Actors to a Business Actor

To assign a Business Actor to another Business Actor, for example "Front Office" to "ArchiSurance" actor:

1. Click the **Link** button 
2. Click the parent business actor and, holding down the mouse button, drag the cursor to the child business actor to be connected and release the mouse button.
   The two Business Actors are highlighted and a dotted line indicates the path that will be taken by the graphic link.
3. Select the **Composition** relationship type.
   The **Composition** link appears in the diagram.

Saving a Diagram

To save your drawing, click the **Save** button  .

BUSINESS PROCESS COOPERATION VIEWPOINT

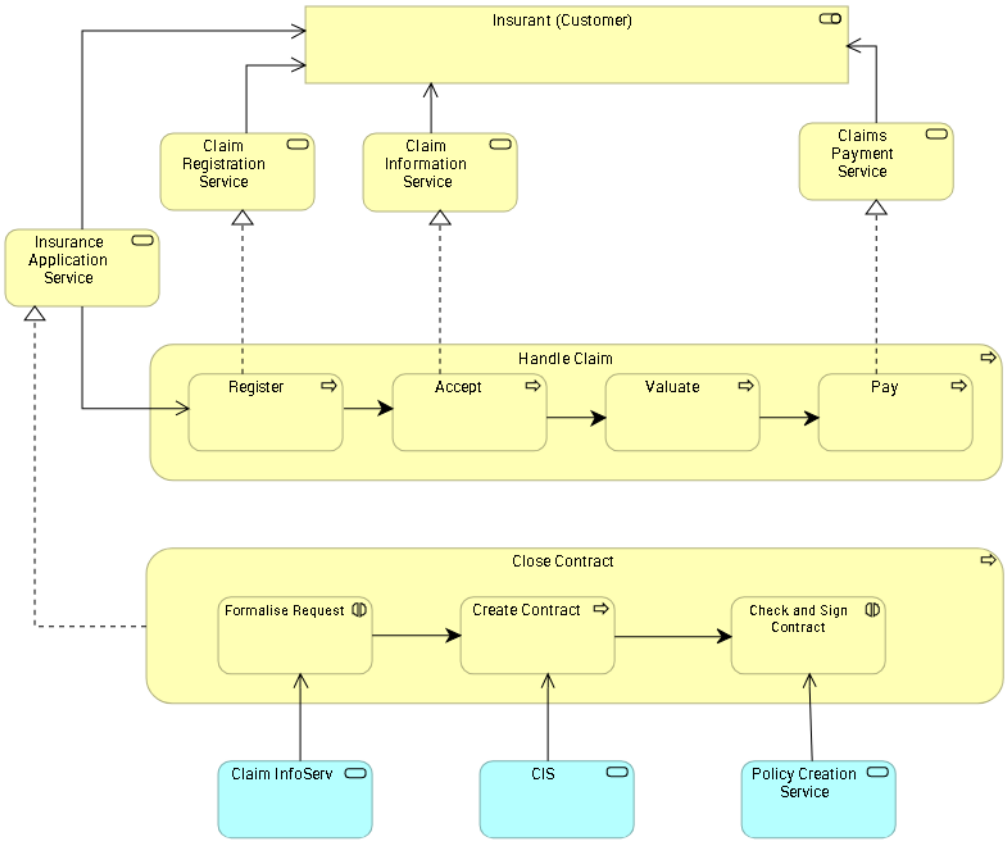
The business process cooperation viewpoint is used to show the relationships of one or more business processes with each other and/or with their environment. It can be used both to create a high-level design of business processes within their context and to provide an operational manager responsible for one or more such processes with insight into their dependencies. Important aspects of business process cooperation are:

- Causal relationships between the main business processes of the enterprise,
- Mapping of business processes onto business functions,
- Realization of services by business processes,
- Use of shared data.

Each of these can be regarded as a "sub-viewpoint" of the business process cooperation viewpoint.

Example Presentation

The diagram below represents the functional description of "Damage Claiming process".



Example of Business Process Coopération viewpoint diagram

Managing a Business Process Cooperation Viewpoint Diagram

You will create Business Services and connect each of these to the Business Role responsible for their execution.

Creating Business Services

To create a Business Service:

1. In the insert toolbar, click the **Business Service** button.
2. Click in the diagram.
   The **Creation of Business Service** dialog box appears.

3. Enter the **Name** of the Business Service, "Claim Registration Service" for example.
4. Click **OK**.
   The business service appears in the diagram.

Creating several Business Services

To create the other Business Services:

1. Double-click the **Business Service** button and then click in the diagram.
   A new business service appears in the diagram.
2. Click again in the diagram.
   Other new business service appear in the diagram.
3. Press key <Esc> to stop the **Business Services** creation.
4. To rename the business service, click the name of the business service, press key <F2> and enter a new name.

Creating Business Roles

To create a Business Role:

1. In the insert toolbar, click the **Business Role** button.
2. Click in the diagram.
   The **Creation of Business Role** dialog box appears.
3. Enter the **Name** of the Business Role, "Insurant (Customer)" for example.
4. Click **OK**.
   The business role appears in the diagram.

To connect a **Business Service** to a **Business Role**, for example "Claim Registration Service" to "Insurant (Customer)":

1. Click the **Link** button 
2. Click the business Service and, holding down the mouse button, drag the cursor to the business role to be connected and release the mouse button.
   A dotted line indicates the path that will be taken by the graphic link.
   The **Serving** link appears in the diagram.

   ☛ *You could also connect a **Business Role** to another **Business Role**, using a **Specialization** relationship. For example "Car Insurant" as a specialization of "Insurant (Customer)".*

Creating Business Processes

Similarly create the **Business Process** "Register".

To specify that "Register" **Business Process** is realized by the "Claim Registration Service" **Business Service**:

1. Click the **Link** button 
2. Click the "Register" business process and, holding down the mouse button, drag the cursor to the business service to be connected and release the mouse button.

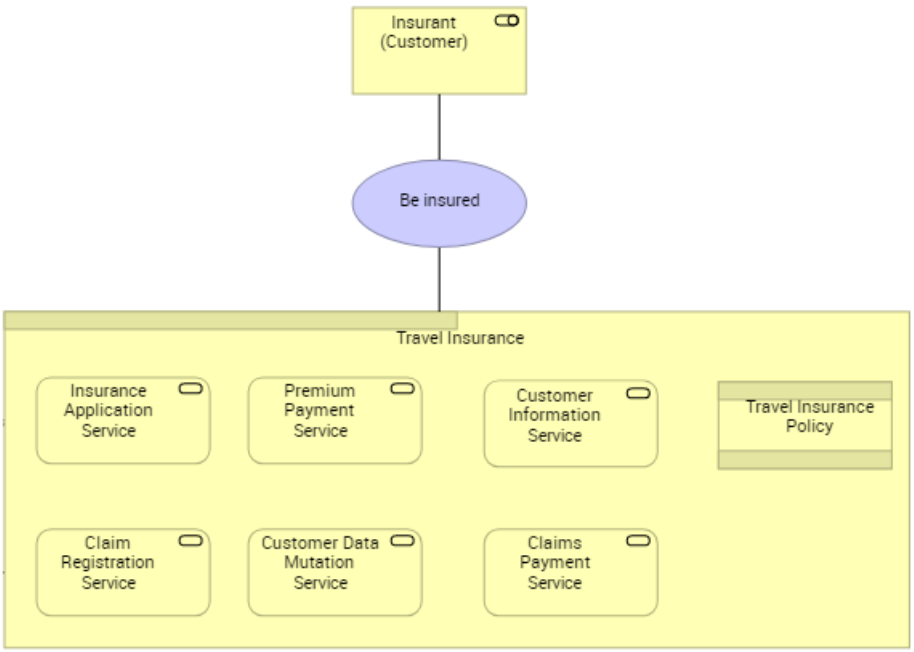
3. Select the **Realization** relationship type.
   The **Realization** link appears in the diagram.

   ☛ *The Realization link appears in grey because it is an indirect relationship.*

PRODUCT VIEWPOINT

The product viewpoint depicts the value that these products offer to the customers or other external parties involved and shows the composition of one or more products in terms of the constituting (business, application, or technology) services, and the associated contract(s) or other agreements. It may also be used to show the interfaces (channels) through which this product is offered, and the events associated with the product. A product viewpoint is typically used in product development to design a product by composing existing services or by identifying which new services have to be created for this product, given the value a customer expects from it. It may then serve as input for business process architects and others that need to design the processes and ICT realizing these products.

The diagram below shows how Business Services are used to describe the "Travel Insurance" product.

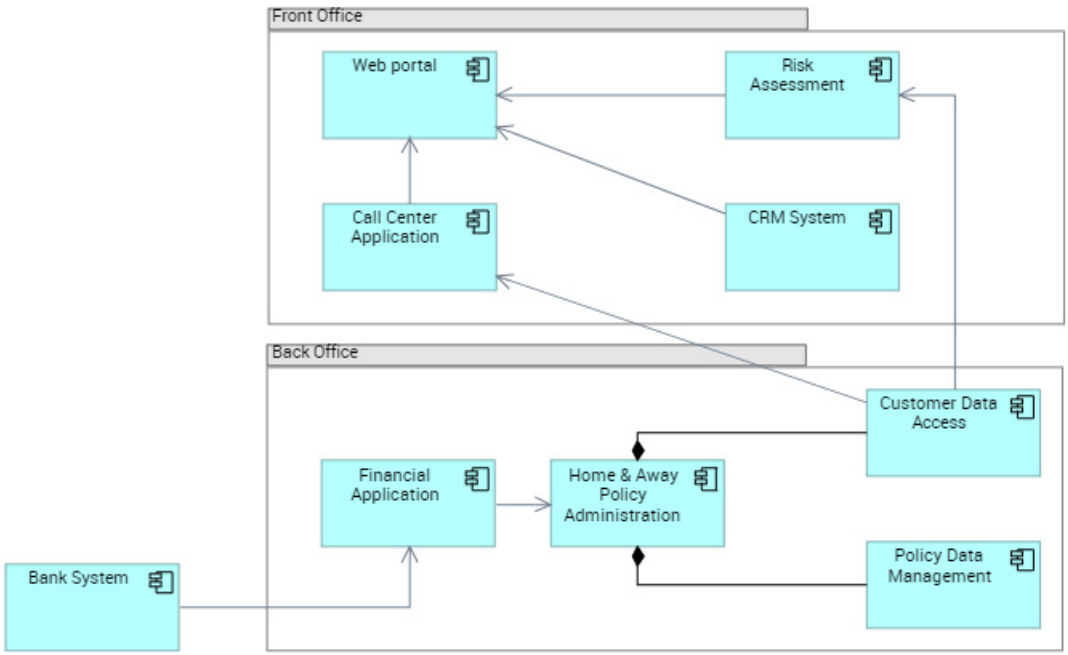


Example of Product viewpoint diagram

APPLICATION COOPERATION VIEWPOINT

The application cooperation viewpoint describes the relationships between applications components in terms of the information flows between them, or in terms of the services they offer and use. This viewpoint is typically used to create an overview of the application landscape of an organization. This viewpoint is also used to express the (internal) cooperation or orchestration of services that together support the execution of a business process.

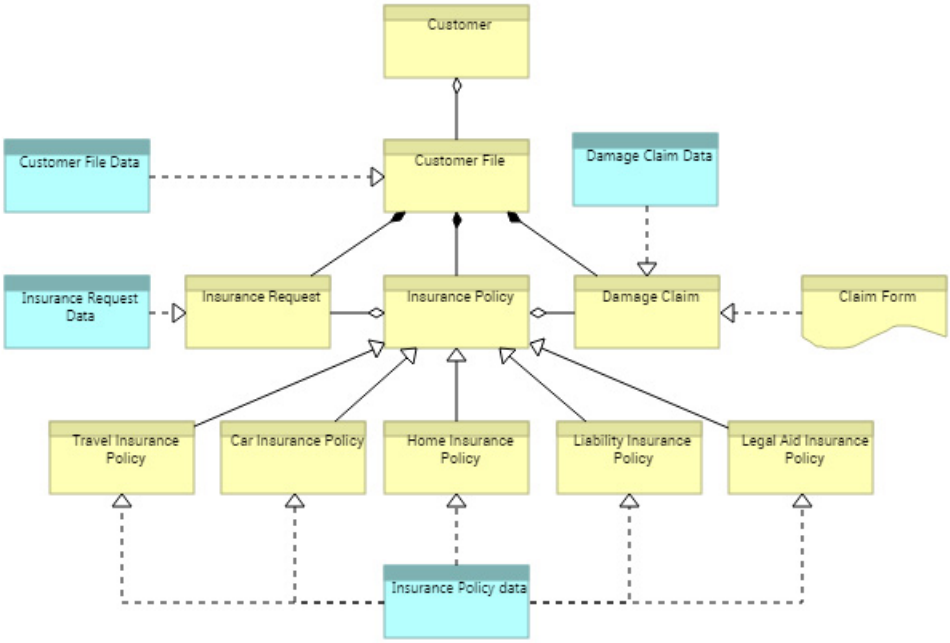
The diagram below details how the involved Application Components interacts with each other so as to provide the Application Services.



Example of Application Cooperation viewpoint diagram

INFORMATION STRUCTURE VIEWPOINT

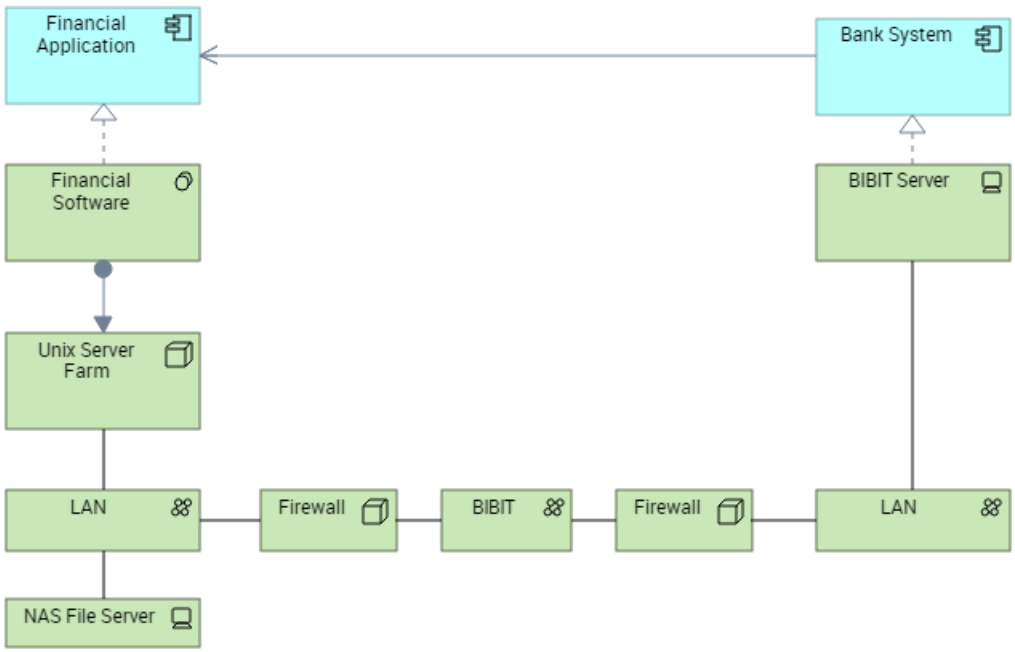
The information structure viewpoint is comparable to the traditional information models created in the development of almost any information system. It shows the structure of the information used in the enterprise or in a specific business process or application, in terms of data types or (object-oriented) class structures. Furthermore, it may show how the information at the business level is represented at the application level in the form of the data structures used there, and how these are then mapped onto the underlying technology infrastructure; e.g., by means of a database schema.



Example of Information Structure viewpoint diagram

IMPLEMENTATION AND DEPLOYMENT VIEWPOINT

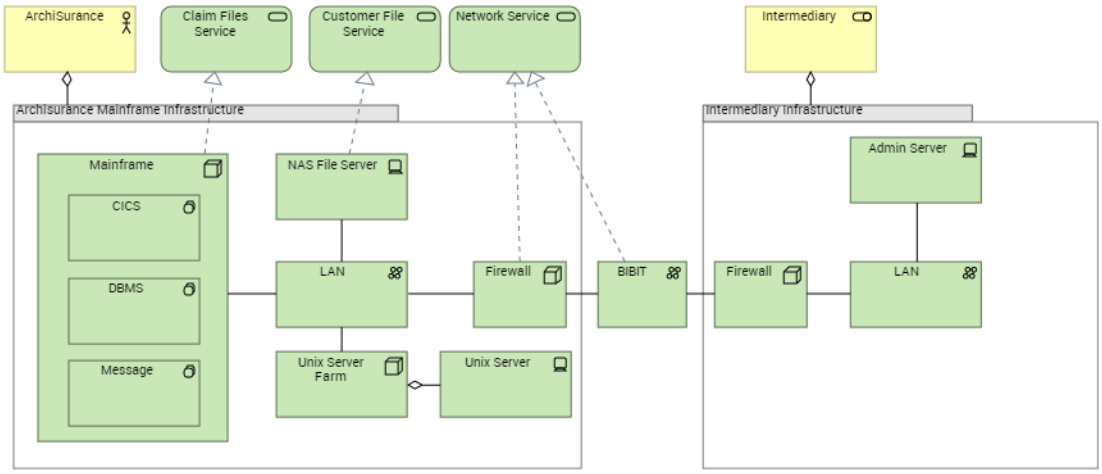
The implementation and deployment viewpoint shows how one or more applications are realized on the infrastructure. This comprises the mapping of applications and components onto artifacts, and the mapping of the information used by these applications and components onto the underlying storage infrastructure.



Example of implementation and deployment viewpoint diagram

TECHNOLOGY VIEWPOINT

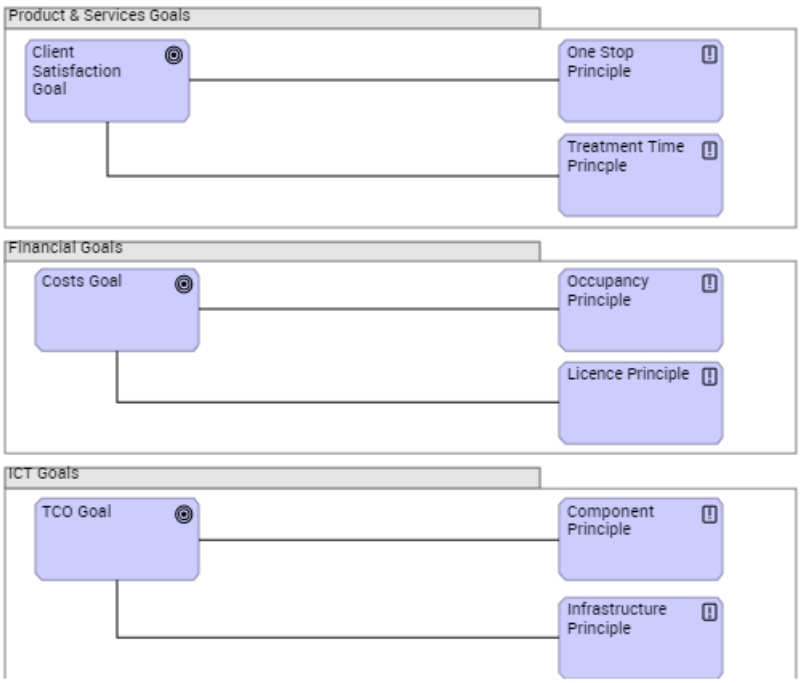
The technology viewpoint contains the software and hardware technology elements supporting the Application Layer, such as physical devices, networks, or system software (e.g., operating systems, databases, and middleware).



Example of Technology viewpoint diagram

MOTIVATION VIEWPOINT

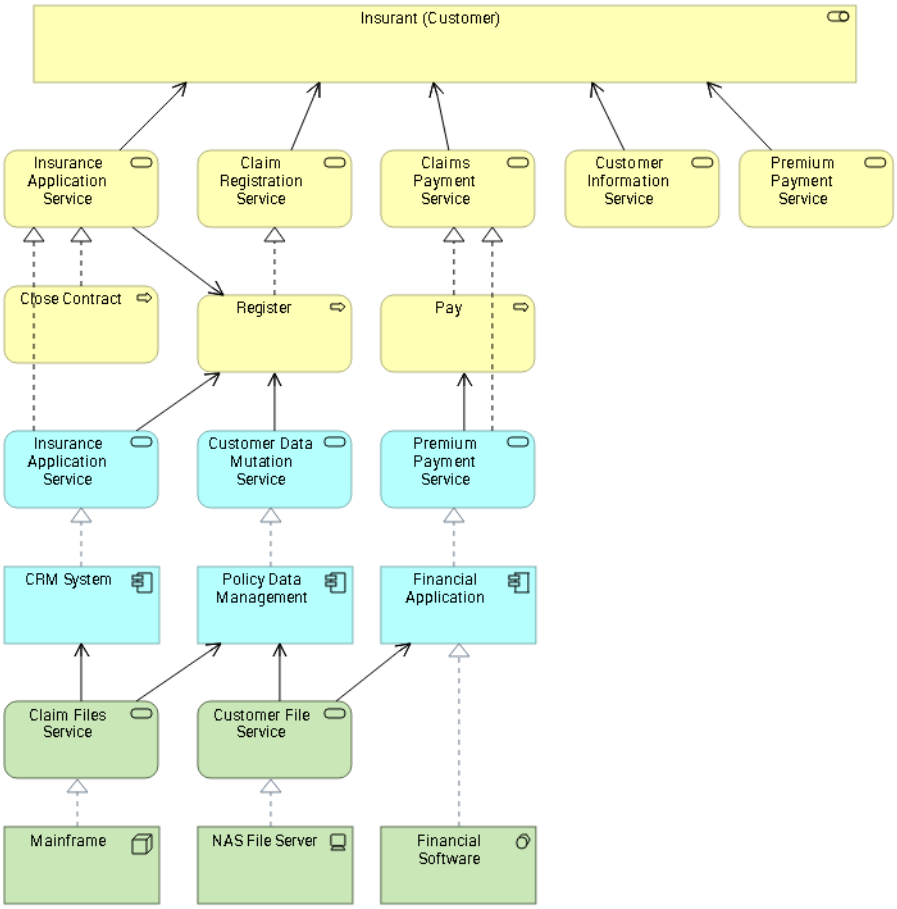
The motivation viewpoint allows the designer or analyst to model the motivation aspect, without focusing on certain elements within this aspect. For example, this viewpoint can be used to present a complete or partial overview of the motivation aspect by relating stakeholders, their primary goals, the principles that are applied, and the main requirements on services, processes, applications, and objects.



Example of Motivation viewpoint diagram

SERVICE REALIZATION VIEWPOINT

The service realization viewpoint is used to show how one or more business services are realized by the underlying processes (and sometimes by application components). Thus, it forms the bridge between the business products viewpoint and the business process view. It provides a "view from the outside" on one or more business processes.



Example of Service Realization viewpoint diagram

THE HOPEX IMPLEMENTATION FOR ARCHIMATE

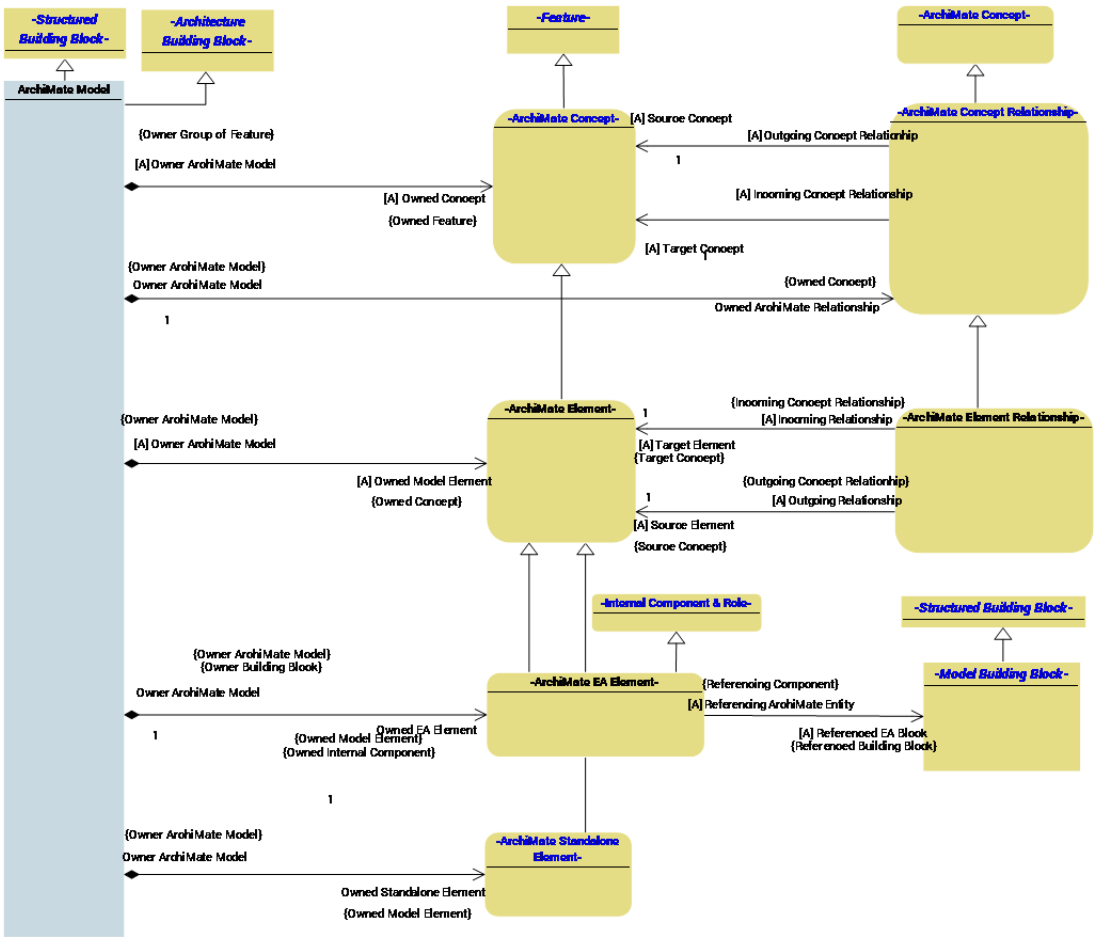
This chapter presents the **HOPEX** MetaModel used to implement the ArchiMate Framework. It presents also the customization possibilities.

☛ *To extend **HOPEX ArchiMate** to meet specific needs, we must use **HOPEX Power Studio** product.*

THE HOPEX METAMODEL FOR ARCHIMATE

Upper ArchiMate Ontology

The upper level ArchiMate abstract MetaModel provides a framework for the implementation of ArchiMate formalism in **HOPEX**.



ArchiMate Upper Ontology abstract MetaModel

ArchiMate Model (on the left side of the figure) is the container which owns the concepts. The main concepts are: **ArchiMate Concept** and **ArchiMate Elements**.

ArchiMate Concept is the root abstract MetaClass of the MetaModel.

ArchiMate Concept has two sub MetaClasses:

- **ArchiMate Elements**, see "ArchiMate Elements", page 61,
- **ArchiMate Concept Relationships**, see "ArchiMate Concept Relationships", page 61.

ArchiMate Elements

ArchiMate Elements MetaClass gathering all the ArchiMate elements implemented to get **HOPEX ArchiMate.** To manage the bridge with **HOPEX ArchiMate** and other **HOPEX** products and concepts, **ArchiMate Elements** MetaClass is divided into sub-MetaClasses:

- **ArchiMate Standalone Elements** for the concepts which are ArchiMate specific. For more details, see "Creating ArchiMate Standalone Elements", page 31.
- **ArchiMate EA Elements** for the concepts which can refer to existing **HOPEX** Building Blocks used in other **HOPEX** solutions. For example:
  - **Application Component** refers to an **HOPEX Application** (available in **HOPEX Business Process Analysis** or **HOPEX IT Architecture**)
  - **ArchiMate Work Package** refers to an **HOPEX Enterprise Project** (available in **HOPEX Project Portfolio Management**)

  These **EA Elements** have a specific behavior for creation and deletion. For more details, see "Creating an ArchiMate EA Element", page 32.

ArchiMate Concept Relationships

ArchiMate Concept Relationships MetaClass is divided into sub-MetaClasses:

- **ArchiMate Association**: a specific Relationship which can associate any concepts (including other Relationships),
- **ArchiMate Element Relationship**: gathering all ArchiMate Relationships, sorted by subtypes, for example: **ArchiMate Composition** or **ArchiMate Aggregation**.

ArchiMate Generic MetaModel

The generic MetaModel serves as a pattern for all the Core Layers MetaModels. It helps identifying the Passive / Behavior / Active elements and also provides abstractions for the relationships defined as part of the ArchiMate generic MetaModel.

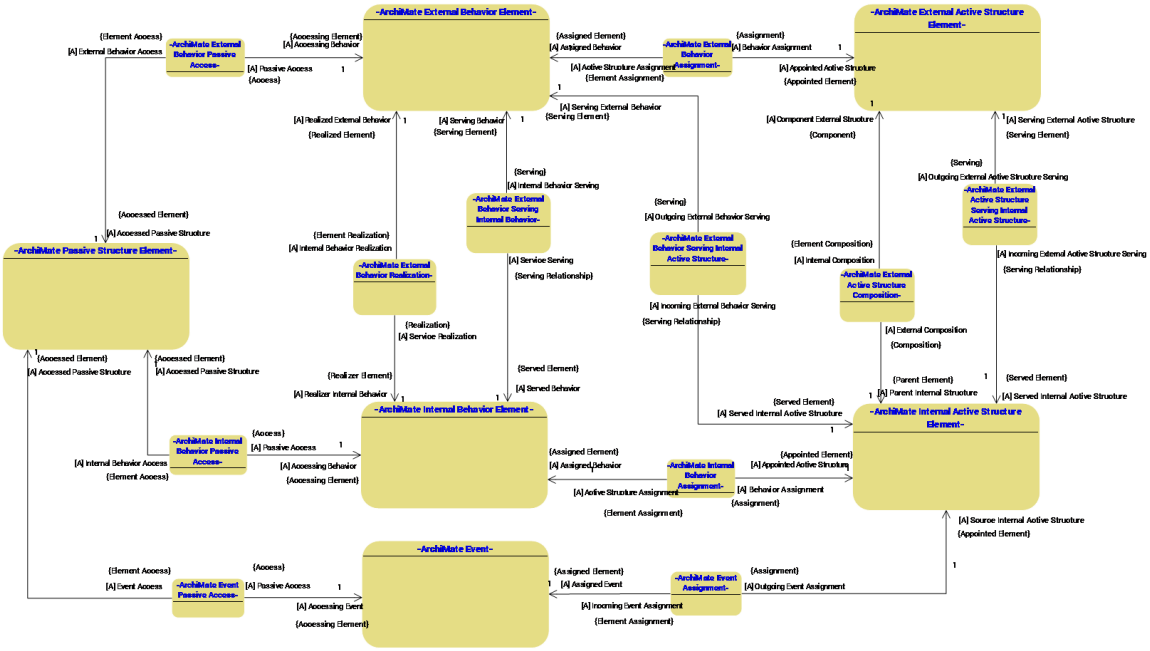
The language consists of **active** structure elements, **behavioral** elements and **passive** structure elements.

These three aspects - active structure, behavior, and passive structure - have been inspired by natural language, where a sentence has a subject (active structure), a verb (behavior), and an object (passive structure).

- The **passive** structure elements are the objects on which behavior is performed. In the domain of information-intensive organizations, which

is the main focus of the language, these are usually information or data objects, but they may also be used to represent physical objects.

- The **behavioral** corresponds to the dynamic aspect. The active structure concepts are assigned to behavioral concepts, to show who or what performs the behavior.
- The **active** structure elements are the business actors, application components and devices that display actual behavior, i.e., the 'subjects' of activity.

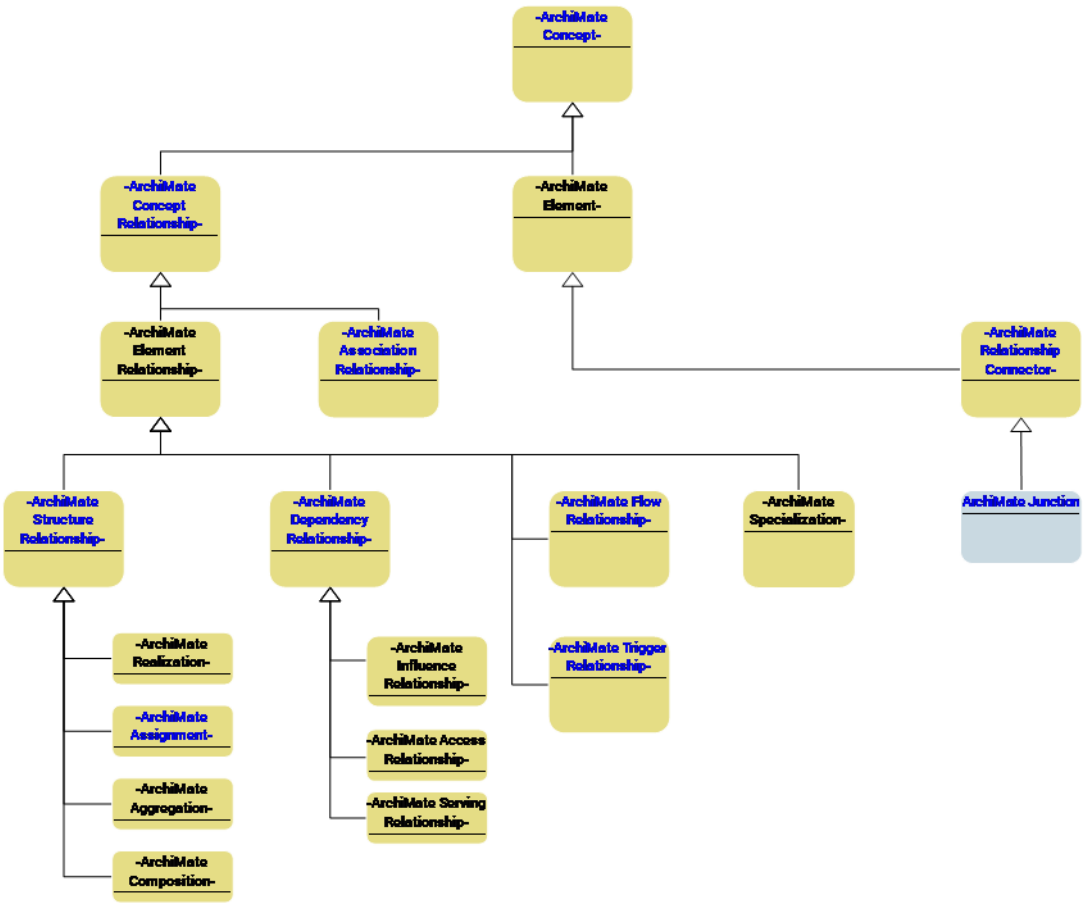


ArchiMate 'Generic MetaModel' implementation

ArchiMate Relationships MetaModel

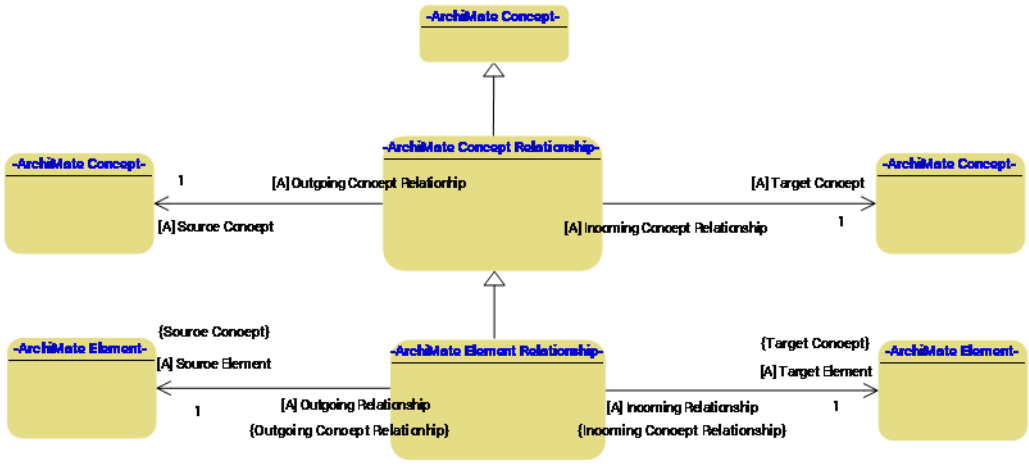
ArchiMate relationships are all sub-MetaClasses of the **ArchiMate Concept Relationship** MetaClass.

- **ArchiMate Association** can link any ArchiMate concepts (elements or relationships), as can structural relationships to or from **Groupings**.
- **ArchiMate Element Relationships** can be used to link two **ArchiMate Elements**. The Relationship subtypes are then available for each relationship type defined in ArchiMate



ArchiMate Relationships

As a general rule, relationships are oriented and follow a source / target generic pattern. They are presented accordingly in properties as outgoing / incoming relationships.



ArchiMate Relationships pattern

☛ *For more details on the use of ArchiMate Relationships in diagrams, see "Create an ArchiMate Relationship", page 33.*

Direct relationships

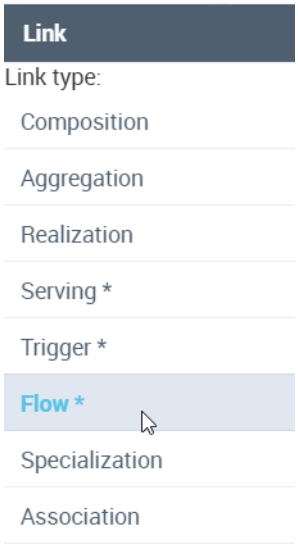
In ArchiMate, some relationships are ***direct*** and form the core structure of the language; they are typically contained in the MetaModel diagrams throughout the ArchiMate standard specification.

Indirect relationships

Indirect relationships are also available; these are obtained by applying derivation rules to the direct relationships. They are defined in the tables in appendix to the ArchiMate standard.

Indirect relationships are available as relationships in **HOPEX ArchiMate**, they are identified in the interface

- By a star '*' symbol following the relationship type name in the relationship creation box,



- By a dark grey color in the diagrams

Indirect relationships can be identified through an **ArchiMate Indirect Relationship** abstract MetaClass and are activated, and desactivated, in the diagram thanks to the ***Indirect relationships*** diagram view.

> ☛ ***HOPEX ArchiMate*** *does NOT provide the means to infer the derived relationships based on the model occurrences (objects), which is the other use case mentioned in the ArchiMate standard.*

ArchiMate Elements

In **HOPEX ArchiMate**, the **ArchiMate Elements** can be used to enable bridging with other **HOPEX** EA products and solutions.

These objects come with a specific creation wizard enabling to reuse **HOPEX** inventory objects or to add new objects to the inventory.

In some cases, the creation wizard will also require additional information that goes beyond the sole ArchiMate universe purpose. For example, when creating a **Work**

Package, the 'project type' (demand, candidate project, ongoing project) is requested, in accordance to **HOPEX Project Portfolio Management** requirements.

This information is required to enable integration in the **HOPEX** product universe.

☛ *The right-click menu shows the **HOPEX** object name instead of the ArchiMate object name. For more details, see "Concepts mapping", page 72.*

To be able to create work packages, the HOPEX Project Portfolio Management solution pack must be imported; since the Work packages are mapped to the Enterprise Project MetaClass from the HOPEX Project Portfolio Management feature, the solution pack is required.

ArchiMate Standalone

The **ArchiMate Standalone** abstract MetaClass gathers the **ArchiMate Elements** which are only available as part of an **ArchiMate Model**.

To add a new ArchiMate specific object, you only need to create a new MetaClass and to define it as a subtype of the **ArchiMate Standalone** MetaClass.

☛ *You may also define it as a subtype of the appropriate ArchiMate Generic MetaModel.*

☛ *For more details, see "Add an ArchiMate Standalone Element", page 68.*

ArchiMate EA Element

The **ArchiMate EA Element** abstract MetaClass gathers the ArchiMate element which are also available as a shared inventory both across ArchiMate models and in other **HOPEX** products, such as **Risks**, for example.

☛ *For more details on the ArchiMate Elements in **HOPEX**, see "Concepts mapping", page 72.*

☛ *For more details on the creation of an ArchiMate EA Element Type, see "Add an ArchiMate EA Element", page 68.*

ArchiMate Element relationship

To add a new relationship, you have to:

- Create a new relationship MetaClass as a subtype of **ArchiMate Concept Relationship**, using the appropriate subtypes to defines its relationship category.
- Define the outgoing **MetaAssociations** as subtypes of the relevant Source / Target relationships.

> ☛ *For more details on the creation of an ArchiMate Element Relationship Type, see "Add an ArchiMate Relationship", page 69.*

Querying the ArchiMate MetaModel

HOPEX ArchiMate defines a high number of relationships to be compliant with the ArchiMate standard.

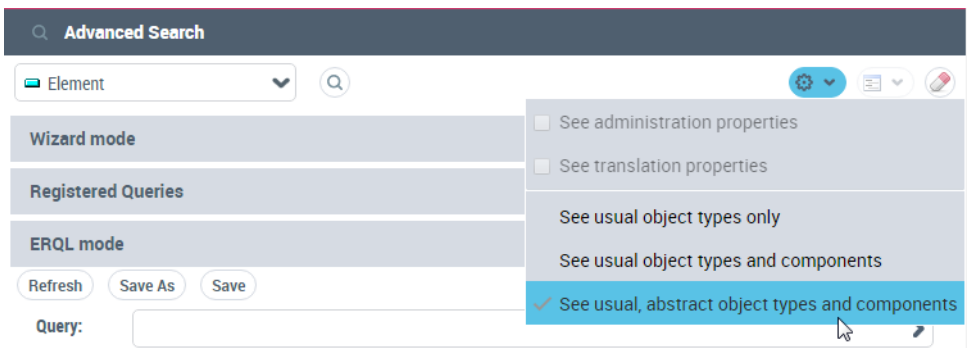
A typical recommendation is to use the provided abstractions when building reports and querying the MetaModel, instead of looking directly at the concrete (lower) level implementation, which may be difficult to read.

Using the Advanced Search tool

A set of useful queries can be found in the **Registered Queries** section of the **Advanced Search** tool, in the selecting the **Element** target, for example.

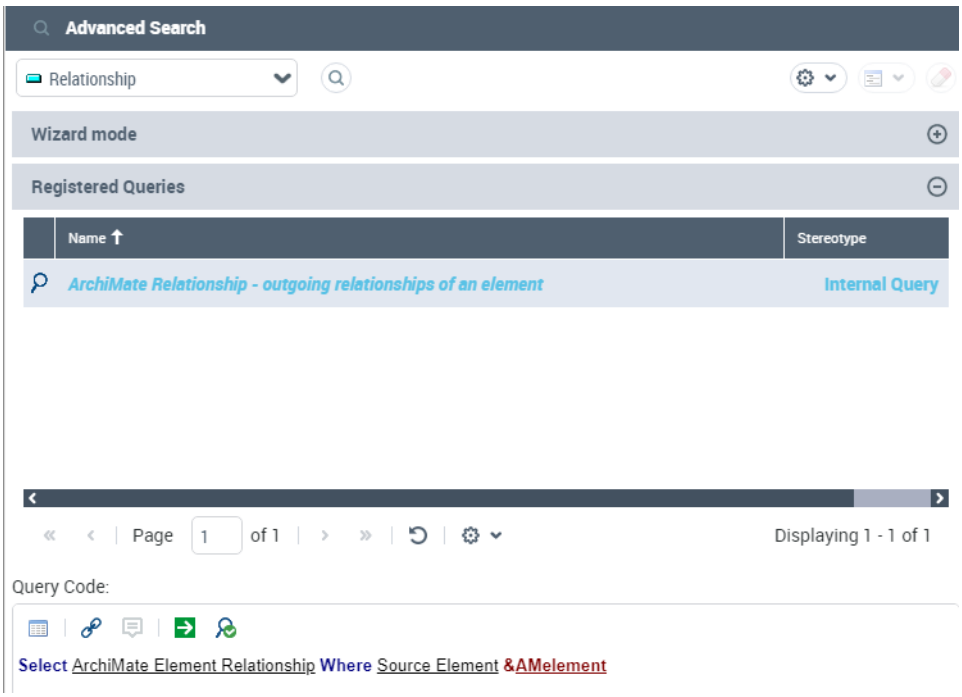
> ☛ *For more information on Query Tools, see the "Presentation of the Advanced Query Tool" section in the **HOPEX Common Features** guide.*
>
> ☛

To be able to select abstract MetaClasses, you must select **See usual abstract objects, types and components** in the advanced query **Display** options.



Querying using Relationship Target

Several typical queries can be used to access the relationships from an object according to the relationships super types.



To get all the outgoing relationships the following query can be used:

Select [ArchiMate Element Relationship] Where [Source Element] &AMelement

For example to get all realization relationships, which are relationships of the realization types 'from' the object, defining which object are realized by the object),

~)G8Ql7d7Qz57[ArchiMate - outgoing realization relationships]

For example:

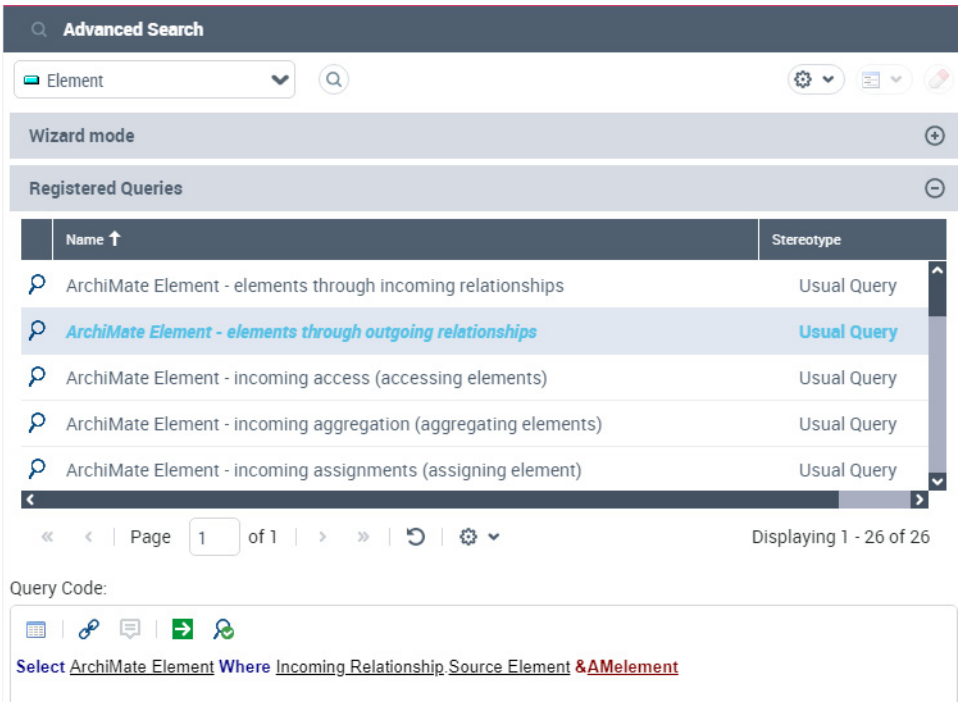
```
Select [ArchiMate Realization] Where [Source Element]
&"AMelement"
```

Querying using Element Target

Similarly, to browse a relationship and to reach the target object through the relationship:

1. In the **Advanced Search** tool, select the **Element** target,

2. Open the **Registered Queries** section to get the typical useful queries.



The following syntax can be used:

```
Select [ArchiMate Element] where [incoming
Relationship].[Source Element] &AMelement
```

Or, conversely, to select the source elements:

```
Select [ArchiMate Element] where [outgoing
Relationship].[Target Element] &AMelement
```

To get, for example, all the realizer elements of an element:

```
Select [ArchiMate Element] Where [outgoing
Relationship]:[ArchiMate Realization].[Realized Element]
&AMelement.
```

Querying using other target types

Some relationships, like **Association**, are specific. In this case the direction does not matter, so if you want to get the associated elements looking both ways; this can be done using the following: **~qRT7gJ2VQv)7[ArchiMate Element - neighboring elements through incoming or outgoing associations]**

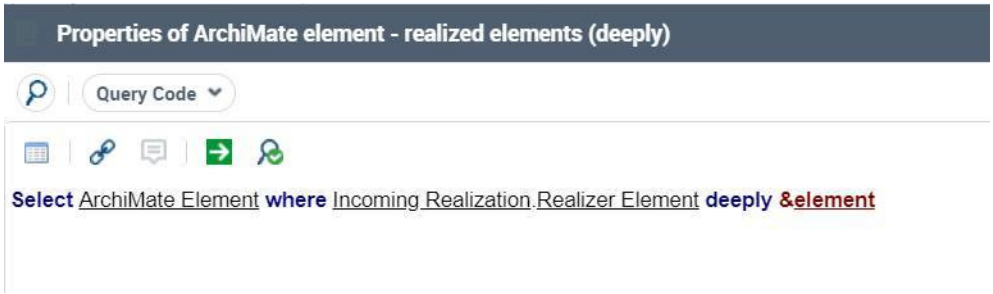
```
Select [ArchiMate Concept] Into @in Where [Incoming
Association].[Associated Source Concept] &"concept"

Select [ArchiMate Concept] Into @out Where [Outgoing
Association].[Associated Target Concept] &"concept"

Select [ArchiMate Element] From @in Or @out
```

Using the deeply ERQL clause

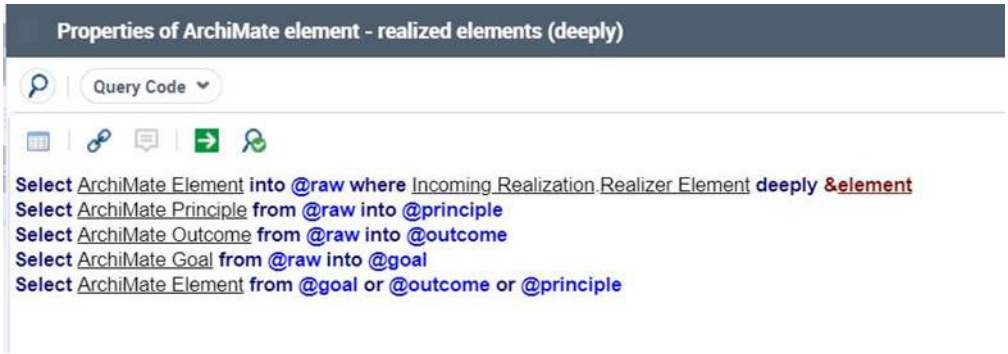
The ***deeply ERQL*** clause can be used with the ArchiMate metamodel, for instance to retrieve the objects through a type of relationship, and recursively the rank "n" objects links to this rank 1 neighboring objects, through the same relationship type.

For instance, to retrieve the realized items of an object (a requirement, for example) and the realized object of these realized objects, the following query can be used:

```
Select [ArchiMate Element] Where [Incoming
Realization].[Realizer Element] deeply &element
```



Sets can also be used to narrow down the list of resulting objects, for instance, to retrieve the **Goals**, **Outcomes** and **Principles** realized by a given **Requirement**, but filtering the intermediate junctions objects, for instance, the following query can be used:



Using the Environment (dendogram) report template

The dendogram report template can be used to visualize the relationships to and from an object, at a defined depth.

This can be achieved by using the appropriate query to browse the neighboring objects, in a recursive manner.

☛ *For more details on the usage of deeply clause which gives similar results, see "Using the deeply ERQL clause", page 80.*

You can, for example, use the "ArchiMate - element through outgoing relationships" query to retrieve all the elements which are target of an outgoing relationship from the object set as parameter:

1. Create a new report, using the **Environment** report template.
2. Define the "ArchiMate - element through outgoing relationships" query as the query to be used.
3. Generate the report.

HOPEX ARCHIMATE CUSTOMIZATION

HOPEX ArchiMate can be customized relying on **HOPEX Power Studio** standard features.

The following MetaModeling conventions and accelerators (macro) shall be used to benefit from the common defined behavior.

Creating an ArchiMate Standalone Element

To create a new ArchiMate Standalone element:

1. Create a new **MetaClass** and define it as a subtype of **ArchiMate Standalone Element**.
2. If necessary, define a **GUIName**.
3. Add **GenericLocalName** as a substitute for **Name**.
4. Add a **MetaPicture** with an icon and a shape for the new MetaClass.
5. Connect your new MetaClass to the "ArchiMate - ArchiMate Element Characteristics" **PropertyPagePresentation**.
6. Update the ArchiMate diagram setup using predefined macros. For more details, see "Updating the ArchiMate diagram type setup", page 91.

> ☛ *If applicable, connect to the appropriate ArchiMate super-type; in this case, applicable relationships connectivity may be inherited automatically based on the generic MetaModel (e.g. passive structure elements can be accessed by behavior elements)*

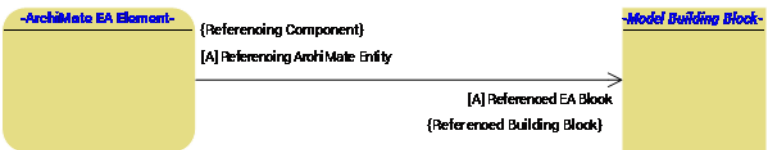
Creating an ArchiMate EA Element

To create a new ArchiMate EA element:

1. Create a new **MetaClass** and define it as a subtype of **ArchiMate EA Element**.

   > ☛ *Only MetaClasses subtypes of **Model Building Block**.*

2. If necessary, define a **GUIName**.
3. Add **GenericLocalName** as a substitute for **Name**.
4. Create a new **MetaAssociation** referring to an existing **HOPEX** MetaClass.
5. Define this new MetaAssociation as subtype of the **ArchiMate EA Entity Reference** MetaAssociation



6. Add a **MetaPicture** with an icon and a shape for the new **ArchiMate EA Element** MetaClass.
7. Connect the MetaClass to the "ArchiMate - ArchiMate Element Characteristics" **PropertyPagePresentation**.
8. Update the ArchiMate diagram setup using predefined macros. For more details, see "Updating the ArchiMate diagram type setup", page 91.

```
Example: add ArchiMate Risk bridging to the Risks inventory
```

Creating or Modifying a Viewpoint

Viewpoints can be added or modified by the **ArchiMate Functional Administrator**.

The modification of viewpoints are available for any ArchiMate Model.

To access to the list of existing viewpoints:

- In the **Models > Inventories** navigation pane, select the **ViewPoints** tile.
  The list of existing viewpoints is displayed.

To modify an existing viewpoint you must be authorized to modify HOPEX Data.

> ☛ *In the options window, select **Repository** and, in the field **Authorized HOPEX Data Modification**,select **Autorize**.*

Creating a Viewpoint

To create a new Viewpoint:

1. In the **Models > Inventories** navigation pane, select the **ViewPoints** tile.
   The list of existing viewpoints is displayed.
2. Click the **New** button.
   The Creation of Viewpoint opens.
3. Select the **Purpose** and **Content** values.

   > ☛ ***Purpose** and **Content** values can be defined for information,*
4. Enter the **Comment** which describes the intent of the viewpoint.
5. Connect the **MetaClasses in the scope** and click **OK**.

   > ☛ *For more details on lues can be defined for information,*

Defining the scope of the Viewpoint

The scope of a Viewpoint is specified by the list of the MetaClasses which are available in the corresponding diagram.

To specify the scope of a Viewpoint:

1. In the **Models > Inventories** navigation pane, select the **ViewPoints** tile.
   The list of existing viewpoints is displayed.
2. Open the property page of the viewpoint that interests you.
3. In the **MetaClasses in the scope** section, click **Connect**.
   The **Connecting** window opens.

4. Select the concepts to add to the scope and click **Content**. The concepts can be:
   - Elements or
   - Relationships : by default, use the generic (abstract metaclasses) to easily define which relationship types are included in the viewpoint definition.

Finalizing Viewpoint creation

To finalize the setup, either when creating a new viewpoint or when modifying an existing one:

➢ Click on the **Refresh Active DiagramTypeViews** button.
A popup menu notifies when the setup is ready.

☛ *The finalization of the viewpoint speeds up the viewpoint diagrams creations and also the model Export.*

Adding properties on ArchiMate concepts

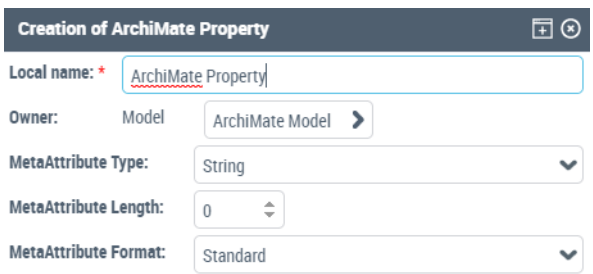
ArchiMate properties can be defined by the **ArchiMate Functional Administrator** for an ArchiMate Model. The new property is available for a set of ArchiMate Concepts (elements or relationships) or for all of them.

Defining properties for an ArchiMate Model

Create a new property

To create a new ArchiMate Property:

1. Open the **Model Properties** property page of your ArchiMate Model.
2. In the **Properties Definition** section, click on **New**.
   A creation wizard opens.



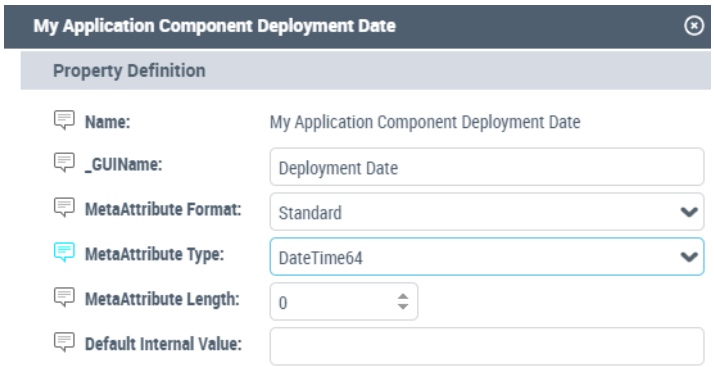
3. Enter the technical **Local name** of the Model Property, for example "My Application Component Deployment Date".

4. Enter information about the type of the new property:
   - **MetaAttribute Type**,
   - **MetaAttribute Format** (set "Standard", excepted for "Currency" MetaAttribute Type),
   - **MetaAttribute Length** (required only for "Strings" MetaAttribute Type).

     ☛ *For more details on these MetaAttributes, see "Specifying the ArchiMate Property MetaAttributes", page 85.*
5. Click **OK**.

To specify the name that will appear in the ArchiMate Concepts property page:

1. Open the Characteristic property page of the ArchiMate property.
2. Enter the **GUIName**, for example "Deployment Date".



Specifying the ArchiMate Property MetaAttributes

A new ArchiMate Property must be compliant with the **ArchiMate Standard**.

Only the types mentioned in the table below are available with the **HOPEX ArchiMate** standard. The other possible types, available in **HOPEX**, are not compliant with the **ArchiMate Standard**.

ArchiMate	MetaAttribute Type	MetaAttribute Length	MetaAttribute Format
String	String	e.g; 63 (short) or 255 (long)	Standard
Boolean	Boolean	n/a	Standard
Currency	Currency	n/a	- Standard (single currency) / - Currency (multi currency)
Date / Time	AbsoluteDateTime64	n/a	Standard
Number	Short/Long/Float	n/a	Standard

☛ *The import only takes into account properties as strings (e.g. the expected serialization format for boolean or dates is not defined in the ArchiMate standard). For more details, see "Properties management", page 97.*

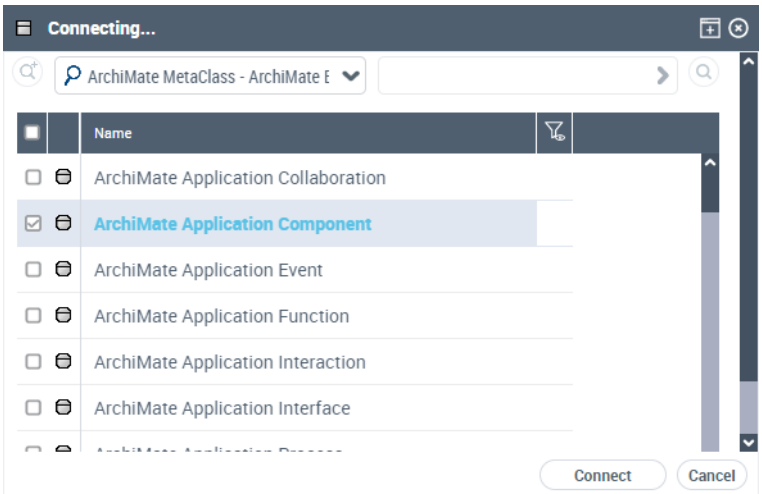
Defining the concepts characterized by the property

By default, the new property is available for all concepts used in **HOPEX ArchiMate**. It is possible to restrict the ArchiMate Property to a limited set of concepts: ArchiMate Elements or Relationships.

```
For example, the "Deployment Date" Property can be used only
for Application Component instances.
```

To specify the applicable concepts:

1. Open the **Model Property** property page.
2. In the **Characterized Concept** section, click on **Connect**.
   A connection window opens.
3. Select the **ArchiMate Element** tab and select the Metaclasses for which the Property applies.

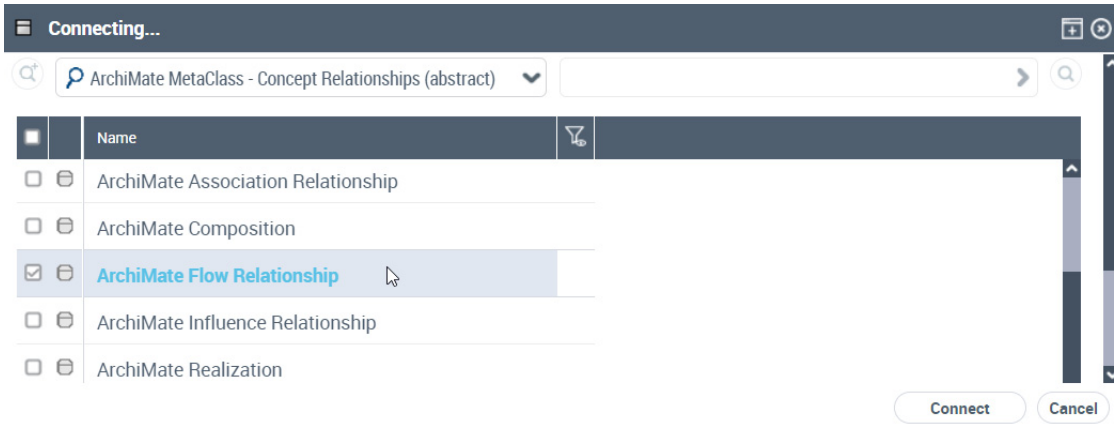


To specify the characterized Concept Relationships:

1. Open the **Model Property** property page.
2. In the **Characterized Concept** section, click **Connect**.
   A connection window opens.

3. Select the **Concept Relationships (abstract)** tab and select the Concept relationships concerned by the Model Property.

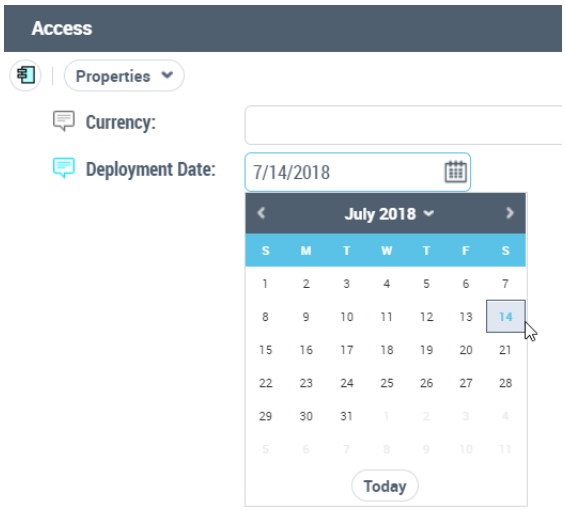
   For example, Archimate Flow Relationships.



☛ *Abstractions are managed by the mechanism, so that a property can be defined for all 'internal active structures', or instances, using the abstract metaclasses of the ArchiMate generic metamodel. The property will therefore be inherited by the subtypes instances.*

Setting properties values for a specific ArchiMate Concept

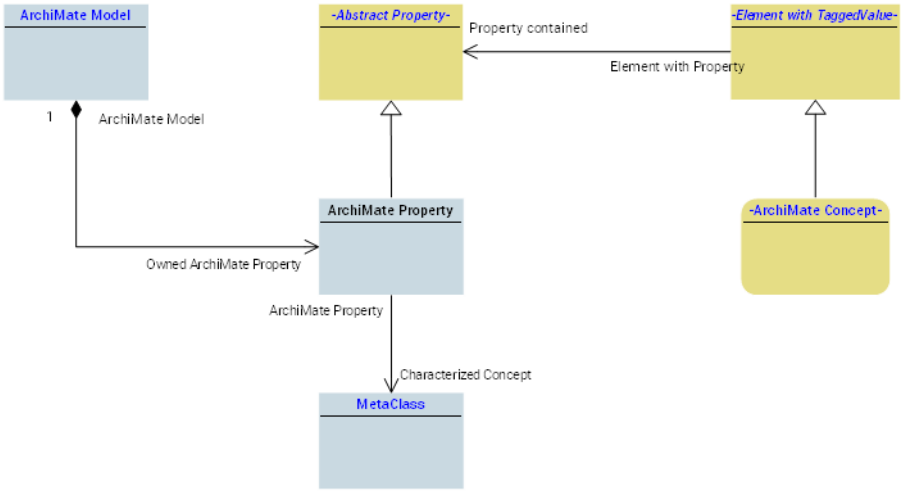
The **Model Properties** are automatically displayed in the **Properties** tab of the characterized concepts instances. Values can be set in the field using the appropriate control (e.g. date picker for dates).



Using Properties in query and reporting

ArchiMate Properties MetaModel

HOPEX ArchiMate standard Metamodel is presented below. The ArchiMate properties MetaModel instantiates the **Abstract Property** MetaClass.



☛ *Values are stored as link attributes between the object instance of **ArchiMate Concept** (**Element with Property** subtype) and the property.*

Query properties

The query syntax to select objects based on the property values is:

- To get an object whose property **Property Name** is "This Name":
  Select [*ArchiMate Concept Name*] Where [Property contained].[Name] = "*ThisName*"
- To get an object whose "This Name" **Property Name** has a given value noted "This Value":
  Select [*ArchiMate Concept Name*] Where [Property contained].[Name] Like "*ThisName*" and [Refers-To] = "*ThisValue*")

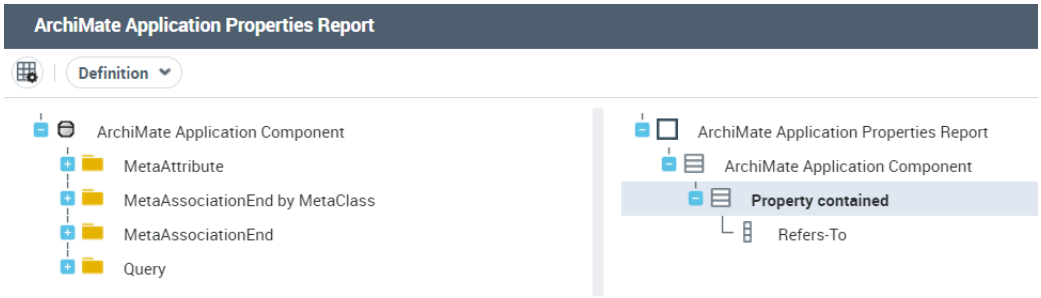
  For instance:

  ```
  Select [ArchiMate Application Component] where [Property contained].([Name] like "#DeploymentVersion#" And [Refers-To] = "1.0")
  ```

Using properties in Report DataSets

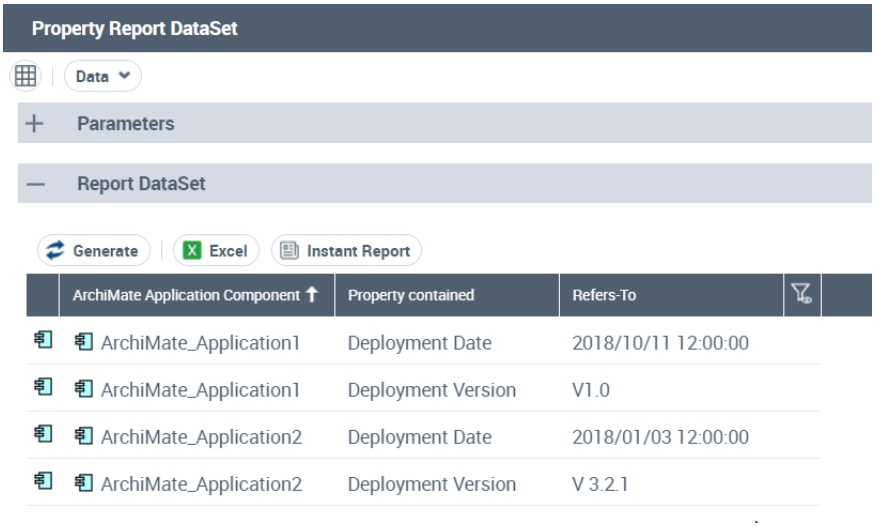
To use the **Model properties** in Report DataSets, you need to:

- Browse the **property contained** MetaAssociationEnd as **Report DataSet Collection** from the root concept
- Display the **Refers-To** MetaAttribute as **Report DataSet Property**.



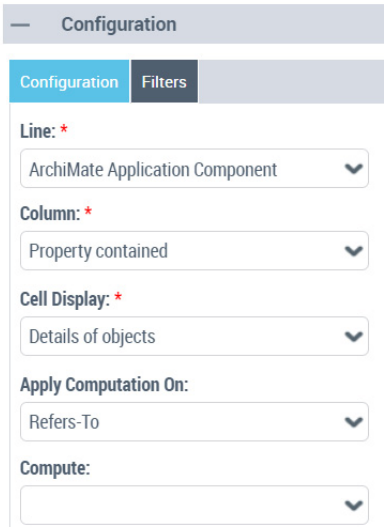
The corresponding Report DataSet presents all the properties of the selected objects:

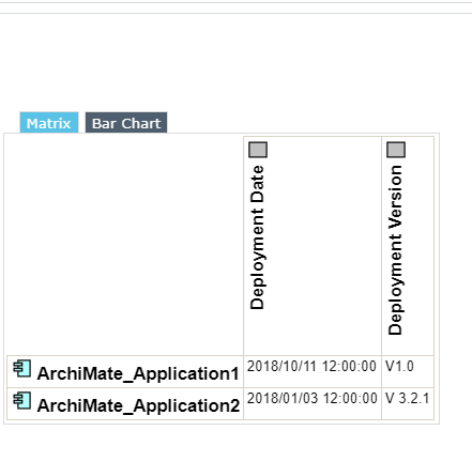
☛ *A query will be used to isolate one of the properties to display it as a separate column in the report data set structure.*



An instant report can be created using **Table** or **Matrix** rendering.

With the corresponding configuration parameters enables to get the report DataSet below.



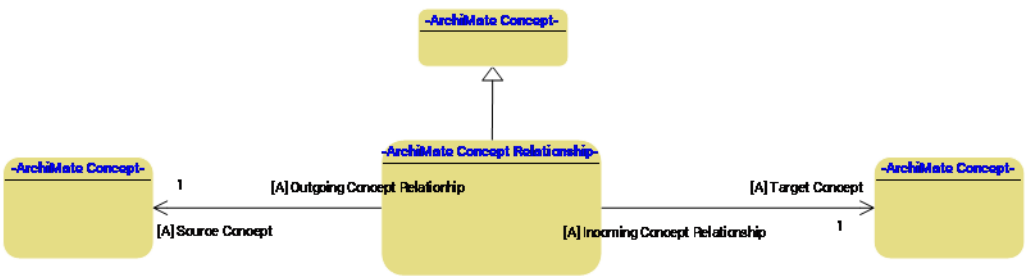


Adding an ArchiMate Relationship

☛ *The association relationship is available by default for all new concepts. So creating specific relationships is not systematically needed.*

To create a new relationship:

1. Create a new **MetaClass** and define it as a subtype of **ArchiMate Concept Relationship** at the appropriate subtype level.
   ☛ *Only MetaClasses subtypes of **Model Building Block**.*
2. If necessary, define a **GUIName** according to the relationship types.
3. Add **GenericLocalName** as a substitute for **Name**.



4. Create two new **MetaAssociations** pointing to the source / target concepts.

5. Define this MetaAssociations as subtype of the **Source / Target** MetaAssociations.
6. Add a **MetaPicture** with an icon of the desired type.
7. Connect the MetaClass to the "ArchiMate - ArchiMate Element Characteristics" **PropertyPagePresentation**.
8. Update the ArchiMate diagram setup using predefined macros. For more details, see "Updating the ArchiMate diagram type setup", page 91.

Updating the ArchiMate diagram type setup

The following macros automatically update the ArchiMate diagram type setup once the MetaModel has been set up using the provided abstractions and patterns as described above.

- **ArchiMate - Diagram Type Update with New Elements and Relationships**: updates the ArchiMate diagram definition with the new elements and relationships.

In case new relationships have been defined, the following macro shall be run.

- **ArchiMate - Diagram Type Relationships Paths Style Update**: updates the ArchiMate diagram definition with desired look and feel for ArchiMate relationship types representation,
- **ArchiMate - Diagram Type Indirect Relationships Paths Style Update**,
- **ArchiMate - Diagram Type Relationships Paths Order Update**.

ARCHIMATE MODEL IMPORT

The Standard for exchange of ArchiMate 3.0/3.0.1 models can be obtained from the Open Group publications catalog at https://www.opengroup.org/bookstore/catalog/C174/.

A dedicated information site with resources including examples is available at https://www.opengroup.org/xsd/archimate/.

ArchiMate import Overview

ArchiMate design can be serialized into an xml file that supports the standard objects and diagrams. The **HOPEX ArchiMate** import aims at importing ArchiMate Models from files so that architectures modeled in other ArchiMate modeling tools can be reused by **HOPEX**.

Pre-Requisites

The ArchiMate import feature is available with **HOPEX ArchiMate** and supports ArchiMate 3.0/3.0.1 models.

Necessary solution packs should have been imported.

☛ *For more details about solution pack, see "Pre-Requisites to HOPEX ArchiMate", page 6.*

The root target container (enterprise or library) should be selected when triggering the import.

☛ *For more details about target container, see "Container management", page 94.*

Scope of ArchiMate Import

The import cover the requirements of the ArchiMate Exchange format:

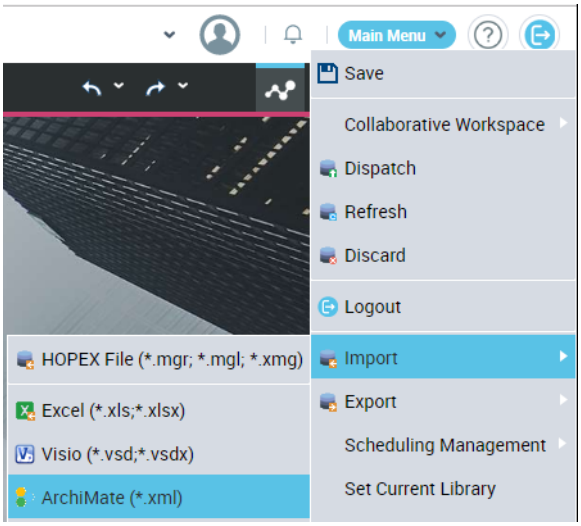
- ArchiMate Elements,
- Relationships,
- Views and diagrams,
- Properties.

For more details on the list of supported mappings, see *"Objects mapping", page 94*.

Importing an ArchiMate File

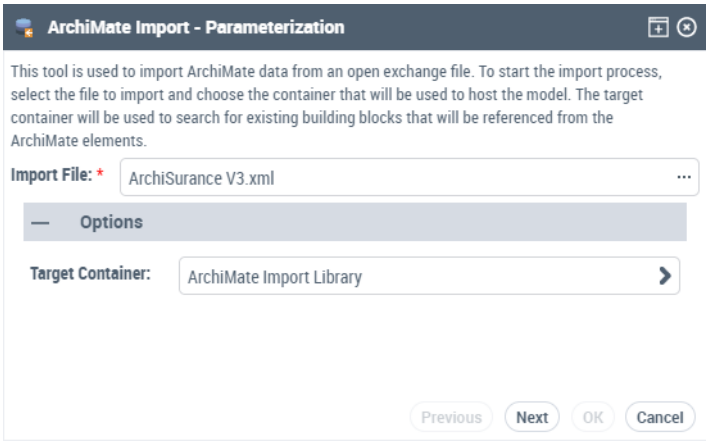
To import an ArchiMate Model file from **HOPEX ArchiMate**:

1. From **Main Menu**, select **Import > ArchiMate (*.xml)**.

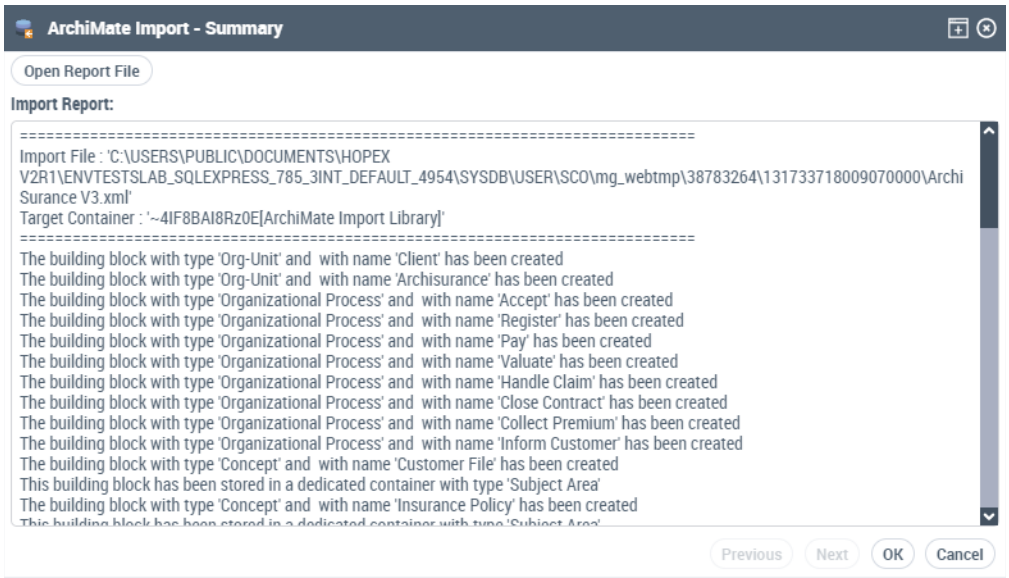


2. In the **Import File** of the **ArchiMate Import - Parameterization** dialog box, specify the path of the file to be imported.
3. (Optional) In the **Option** section, select a **Target Container**.

☛ *For more details about target container, see "Container management", page 94.*



4. Click **Next**.
   The file is uploaded and a summary is edited.



An import report file is displayed.

5. Click **Close**.
   The ArchiMate model, views, elements, relationships, properties and views' diagrams are created in the **HOPEX ArchiMate** repository

Objects mapping

Container management

The objects imported may be stored in a specific container, for example: an enterprise or a library. This container is set in the **Target container** field during the ArchiMate file import.

☛ *For more details about the **Target container** specification, see "Importing an ArchiMate File", page 93.*

ArchiMate Elements import

The **ArchiMate standalone elements** are imported as new standalone elements in the ArchiMate model.

☛ *For more details **ArchiMate standalone elements**, see "ArchiMate Standalone", page 76.*

The **ArchiMate EA elements** import is divided into two steps:

☛ *For more details **ArchiMate EA elements**, see "ArchiMate EA Element", page 76.*

1. The **ArchiMate EA element** creation.

2. A specific processing regarding the **HOPEX Building Block** referenced by the new **ArchiMate EA element**:
   - If an **HOPEX Building Block** (with the same name) is defined in the target scope (container or imported container), it is retrieved and linked as referenced building block.
   - If no **HOPEX Building Block** is available, or if ***many*** are available with this name in the defined scope, then a new objet is created in the defined target container, and a warning is added to the import report

     ☛ *Depending on the referenced **HOPEX Building Block**, a specific creation wizard is activated. For more details, see "Specific Creation Behavior", page 96.*

ArchiMate Relationships

☛ *For more details **ArchiMate Element relationship**, see "ArchiMate Element relationship", page 77.*

The processing of **ArchiMate Concept Relationships** consists in considering the **type (metaclass)** of the source element of the relationship, to get the outgoing relationships (concrete path) of the suitable relationship type towards the suitable target concept.

- If no such relationship is available, then a new relationship of the 'association' type is created (*association from 'source_object' to 'target_object' created instead of the required 'relationship_type') and a* warning is added to the import report file.
- If the source (or target) object type does not exist, an error message is added to the import report file *(non-existing source / target object: relationship of 'relationship_type' from source_object to target_object could not be created)*.
- Otherwise, the relationship is created.

The processing of **Relationships with attributes** consists in importing the attributes as attribute or object.

- **Flow**: the flow relationship name attribute is defined by the carried content name:
  - If no name is defined for the flow relationship, then no content is associated with the flow relationship,
  - If a content, with the same name, exists in the scope (in target container or target container imported containers), the content is reused,
  - otherwise a new content is created within the target container.
- **Access**: Access Type
  - Access type is stored as a closed enumeration attribute (read, write or read/write).
- **Influence**: Impact Type

  ☛ *In **HOPEX**, the impact type (or strength) is an enumeration, so an opened list **+++** / **++** / **+** / **-** / **--** / **---***
  - If the value fits with this list, the corresponding attribute value of **HOPEX** is used,
  - Otherwise, the imported value is set outside of the other predefined values (e.g. low, middle, high).

Specific Creation Behavior

The processing of **ArchiMate EA elements** import depends on the referenced **HOPEX Building Block**.

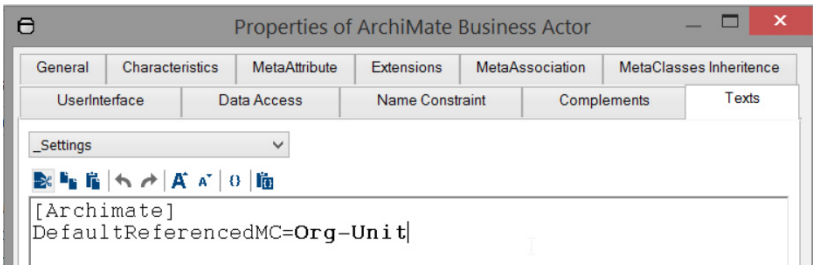
☛ *For more details on **ArchiMate EA elements** import, see "ArchiMate Elements import", page 94.*

The table below present the cases where a specific creation wizard is activated when the **ArchiMate EA elements** is imported.

☛ *For specific wizards, a simple creation is done. For example, an ArchiMate Project is created without workflow and status.*

Layer	ArchiMate EA Elements	HOPEX Metaclas	Specific creation wizard
Strategy	Capability	Business Capability	Simple case
Business	Business Object	Concept	Specific concept creation wizard
	Business Process	Organizational Process	Simple case
	Business Actor	Org-Unit, Position Type	Complex case: the appropriate concept must be selected
Application	Data Object	Class, Entity	Complex case: the appropriate concept must be selected
	Application Process	System Process	Simple case
	Application Component	Application	Simple case
Technology	Technology Process	System Process	Simple case
	Device	IT Device, IT Server, IoT Device	Complex case: the appropriate concept must be selected
	System Software	Software Technology	Simple case
	Communication	IT Network	Simple case
	Network	Facility	Simple case
	Facility	Hardware	Simple case
Implementation & migration	Work Package	Enterprise Project	Specific enterprise project creation wizard
Other	Location	Site	Simple case

In ***complex cases***, the default MetaClass used by the wizard is defined by a specific setting on the sourcce ArchiMate concept MetaClass; this can be customized if necessary. For example, a Business Actor references an Org-Unit by default.



Views and Diagrams

All views, of any suitable viewpoint type, are created.

> ☛ *In the ArchiMate format, views are diagrams while, in **HOPEX**, one view object is described by a diagram object (of the ArchiMate diagram type)*

In the view's diagrams, the imported file object coordinates are used to position the object in the diagram

If an object is added to the diagram while the viewpoint definition doesn't allowing the element (or relationship), then the diagram view is activated, and a warning added to the import report file.

Properties management

ArchiMate properties can be defined for an ArchiMate Model. The new property is available for a set of ArchiMate Concepts (elements or relationships) or for all of them.

> ☛ *For more details on standard properties management with **HOPEX ArchiMate**, see "Adding properties on ArchiMate concepts", page 84.*

Though in **HOPEX** the various properties formats are available, properties are imported in **HOPEX ArchiMate** in string format only.

In **HOPEX**, a property can be linked to a specific target MetaClass. This is not the case in the standard ArchiMate format. So, with **HOPEX ArchiMate**, a property can be allocated to the desired MetaClass after import.

Language management

To manage multiple language, **HOPEX** allows you to enter data in several languages. You can switch from one language to another to enter or consult data or to generate documents.

If your ArchiMate import file has been generated with a language available with **HOPEX**, you can set the appropriate **HOPEX** data langage before importing your file.

If the langage of the source file doesn't exist in **HOPEX**, the corresponding language specific darta is not imported.

To modify the **HOPEX** data language:

- On your **HOPEX** desktop, select **Main Menu > <current language> > <new language>**.
  The data (translated) appears directly in the selected language.

  ☛ *When you change the data language, this language is kept for your next connection.*

APPENDIX

Concepts mapping

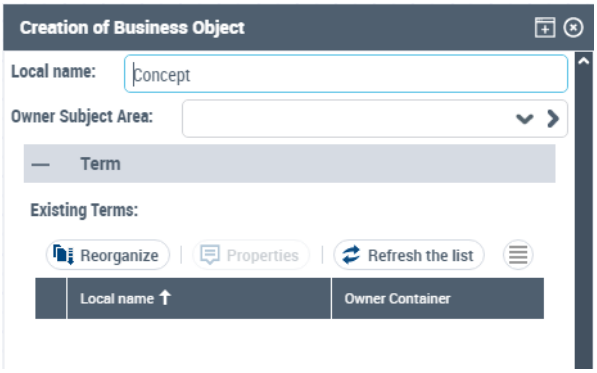
The list of ArchiMate concepts with their **HOPEX** equivalents is presented in the table below.

Layer	ArchiMate EA Element	HOPEX MetaClass	Specific creation wizard
Strategy	Capability	Business Capability	Simple Case
Business	Business Object	Concept	***Specific concept creation wizard***
	Business Process	Organisational Process	Simple Case
	Business Actor	Org-Unit, Position Type	***Complex case: the appropriate concept must be selected***
Application	Data Object	Class, Entity	***Complex case: the appropriate concept must be selected***
	Application Process	System Process	Simple Case
	Application Component	Application	Simple Case
Technology	Technology Process	System Process	Simple Case
	Device	IT Device, IT Server, IoT Device	***Complex case: the appropriate concept must be selected***
	System Software	Software Technology	Simple Case
	Communication Network	IT Network	Simple Case
Physical	Facility	Facility	Simple Case

Layer	ArchiMate EA Element	HOPEX MetaClass	Specific creation wizard
	Equipment	Hardware	Simple Case
Implementation & Migration	Work Package	Enterprise Project	***Specific Enterprise project creation wizard***
Other	Location	Site	Simple Case

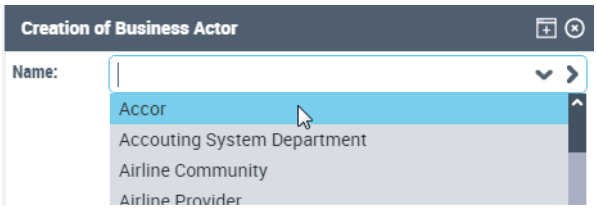
Specific creation wizards are provided for:

- **Business Object** using an existing **Concept** element:

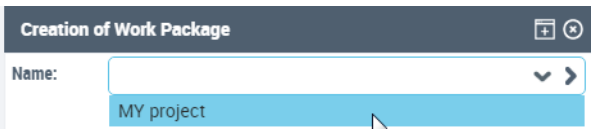


☛ *For more details on **Concept** in **HOPEX** solutions, see **HOPEX Information Architecture**.*

- **Business Actor** using an existing **Org-Unit** or **Position-Type** element:



- **Work Package** using an **Enterprise Project** existing element.



Sample ViewPoints Table

The table below present the list of ArchiMate Elements available in each ViewPoint in the standard **HOPEX ArchiMate** Solution.

ViewPoint	Defined MetaClass	Available ArchiMate Elements
Application Coorperation	Application Layer Element Junction Concept Relationship	Data Object Application Component Application Collaboration Application Interface Application Event Application Service Application Function Application Process Application Interaction Junction
Application usage and Business Process Coorperation	Application Layer Element Business Layer Element Junction Concept Relationship	Business Role Business Actor Business Collaboration Business Service Business Interface Business Event Business Process Business Function Business Interaction Business Object Contract Representation Data Object Application Component Application Collaboration Application Interface Application Event Application Service Application Function Application Process Application Interaction Junction
Capability Map	Capability Outcome Resource Concept Relationship	Capability Outcome Resource

ViewPoint	Defined MetaClass	Available ArchiMate Elements
Goal Realization	Goal Outcome Principle Requirement Pattern Concept Relationship	Goal Outcome Principle Requirement Constraint
Implementation & Deployment	Application Layer Element Technology Layer Element Junction Concept Relationship	Data Object Application Component Application Collaboration Application Interface Application Event Application Service Application Function Application Process Application Interaction Junction Artifact Technology Interface Node System Software Device Technology Collaboration Path Communication Network Technology Event Technology Service Technology Function Technology Process Technology Interaction Equipment Facility

ViewPoint	Defined MetaClass	Available ArchiMate Elements
Implementation & Migration	Business Actor Business Role Core Layer Element Deliverable Gap Goal Implementation Event Location Plateau Requirement Pattern Work Package Concept Relationship	Business Role Business Actor Business Collaboration Goal Requirement Constraint Business Service Business Interface Business Event Business Process Business Function Business Interaction Business Object Contract Representation Location Data Object Application Component Application Collaboration Application Interface Application Event Application Service Application Function Application Process Application Interaction Junction Artifact Technology Interface Node System Software Device Technology Collaboration Path Communication Network Technology Event Technology Service Technology Function Technology Process Technology Interaction Material Equipment Facility Distribution Network Implementation Event Plateau Gap Deliverable Work Package

ViewPoint	Defined MetaClass	Available ArchiMate Elements
Information Structure	Artifact Business Object Data Object Meaning Representation Association Specialization Structure Relationship	Meaning Business Object Representation Data Object Artifact
Migration	Gap Plateau	Gap Plateau
Motivation	Assessment Driver Goal Stakeholder Meaning Outcome Principle Specialization Value Structure Relationship Dependency Relationship Association	Stakeholder Meaning Value Driver Assessment Goal Outcome Principle
Organization	Business Actor Business Collaboration Business Interface Business Role Junction Location Specialization Structure Relationship Association	Business Role Business Actor Business Collaboration Business Interface Location Junction
Physical	Communication Network Device Junction Node Path Concept Relationship	Junction Node Device Path Communication Network

ViewPoint	Defined MetaClass	Available ArchiMate Elements
Product	Application Layer Element Business Layer Element Product Technology Service Value Structure Relationship	Business Role Business Actor Business Collaboration Value Business Service Business Interface Business Event Business Process Business Function Business Interaction Business Object Contract Representation Product Data Object Application Component Application Collaboration Application Interface Application Event Application Service Application Function Application Process Application Interaction Technology Service
Project	Business Actor Business Role Deliverable Goal Implementation Event Work Package Concept Relationship	Business Role Business Actor Goal Implementation Event Deliverable Work Package
Resource Map	Capability Resource Work Package Concept Relationship	Capability Resource Work Package

ViewPoint	Defined MetaClass	Available ArchiMate Elements
Service Realization	Application Layer Element Business Layer Element Specialization Structure Relationship Association	Business Role Business Actor Business Collaboration Business Service Business Interface Business Event Business Process Business Function Business Interaction Business Object Contract Representation Data Object Application Component Application Collaboration Application Interface Application Event Application Service Application Function Application Process Application Interaction

ViewPoint	Defined MetaClass	Available ArchiMate Elements
Strategy	Course of Action Capability Resource Outcome Concept Relationship	Capability Course of Action Resource Outcome
Technology	Composite Element Junction Technology Layer Element Concept Relationship	Location Grouping Junction Artifact Technology Interface Node System Software Device Technology Collaboration Path Communication Network Technology Event Technology Service Technology Function Technology Process Technology Interaction Equipment Facility Plateau
Technology Usage	Application Layer Element Junction Technology Layer Element Concept Relationship	Data Object Application Component Application Collaboration Application Interface Application Event Application Service Application Function Application Process Application Interaction Junction Artifact Technology Interface Node System Software Device Technology Collaboration Path Communication Network Technology Event Technology Service Technology Function Technology Process Technology Interaction Equipment