



Capella a mini User Guide

to apply the  method
ARCADIA

www.thalesgroup.com

REF xxxxxxxxxxxx rev xxx - date
Name of the company / template : 87211168-DOC-GRP-EN-005

THALES GROUP INTERNAL



Contents

Arcadia Modelling Language Quick Reference

Startup: Creating, populating & exploring a Project

- Capella User Interface
- Activity Explorer
- Editing Information & Properties on a Model Element

Creating and editing Diagrams

- Tips for Diagram Edition
- Hide, delete from diagram or model
- Diagrams only display Model Contents
- A box inside another box is meaningful!
- Visual Filters
- Visual Synthesis Capabilities

Specific Diagram Editing

- Capella Diagrams Types
- Creating a functional Chain
- Creating a Scenario
- Creating a Mode/State Machine
- Implementing Behavioral components on Implementation Component
- Building Exchange Items and Interfaces
- Allocating Exchange Items to functional Exchanges
- Allocating functional Exchanges on a Component Exchange
- Linking an Interface to a behavioral Component Port
- Allocating behavioral Component Exchanges on a physical Link
- Creating a Physical Path
- Allocating behavioral Component Exchanges on a physical Path

The Reference: Capella User Manual

This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2020 All rights reserved.

Only first essential concepts and capabilities in this *introductory guide*

Much more powerful features to be found in User Manual

- Accessible inside Capella:
‘Help/Help Contents/Capella Guide/User Manual’
 - (for the version of Capella currently running)

- Online: <https://github.com/eclipse/capella/wiki/User-manual>
 - (for the last version of Capella)
 - Further links below reference this site



Prerequisite:

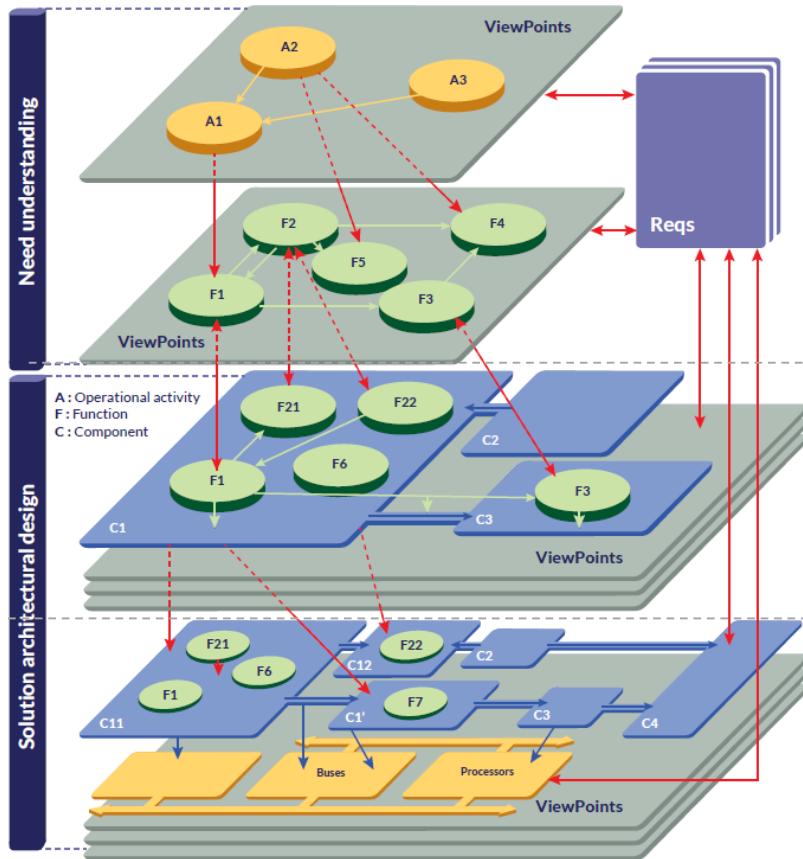
Arcadia Modelling Language Quick Reference

www.thalesgroup.com

THALES GROUP INTERNAL



The Arcadia Method main Perspectives



Operational Analysis

What the users of the system need to accomplish

System Needs Analysis

What the system has to accomplish for the users

Logical Architecture (conceptual solution)

How the system will work in order to fulfil expectations

Physical Architecture (finalized solution)

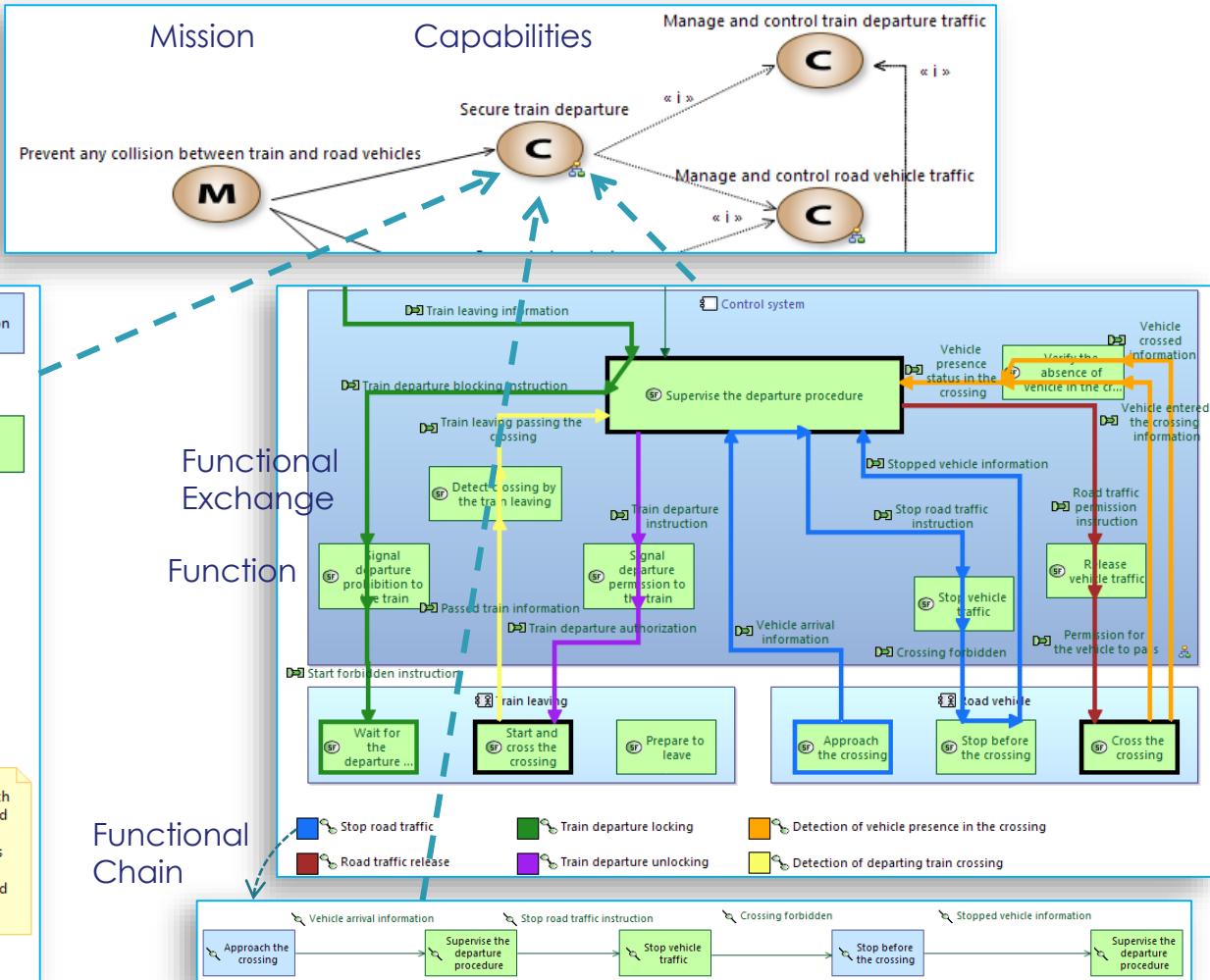
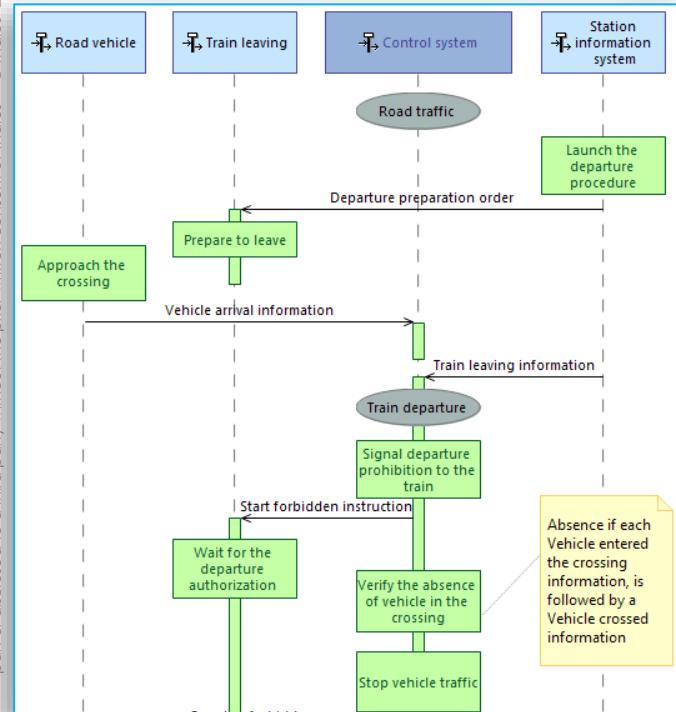
How the system be developed and built

Missions & Capabilities

(see definitions later)

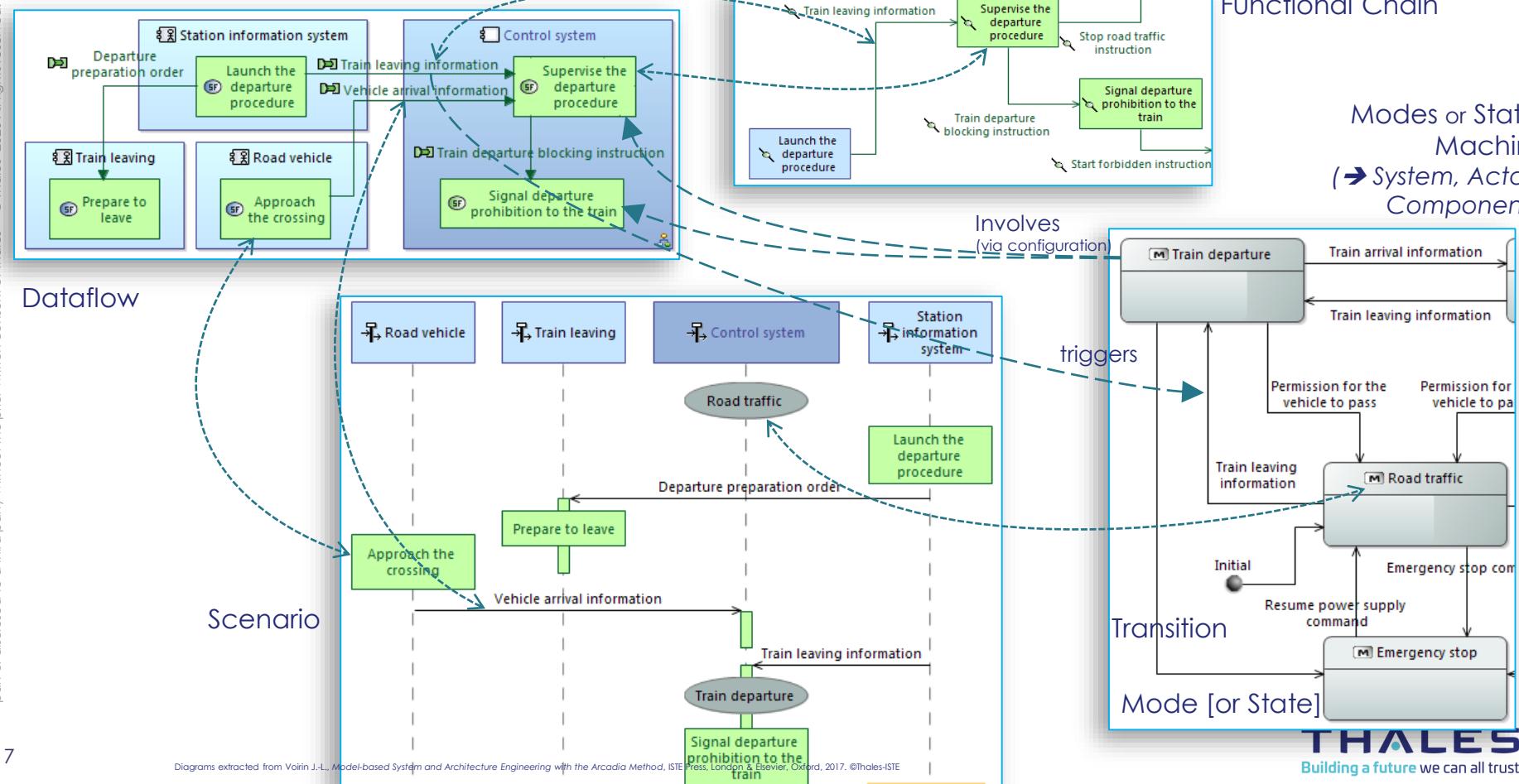
This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2020 All rights reserved.

Missions & Capabilities are described by Functional Chains & Scenarios using Dataflows



Dataflow / Scenarios & F. Chains / Modes & States Consistency

(see definitions later)

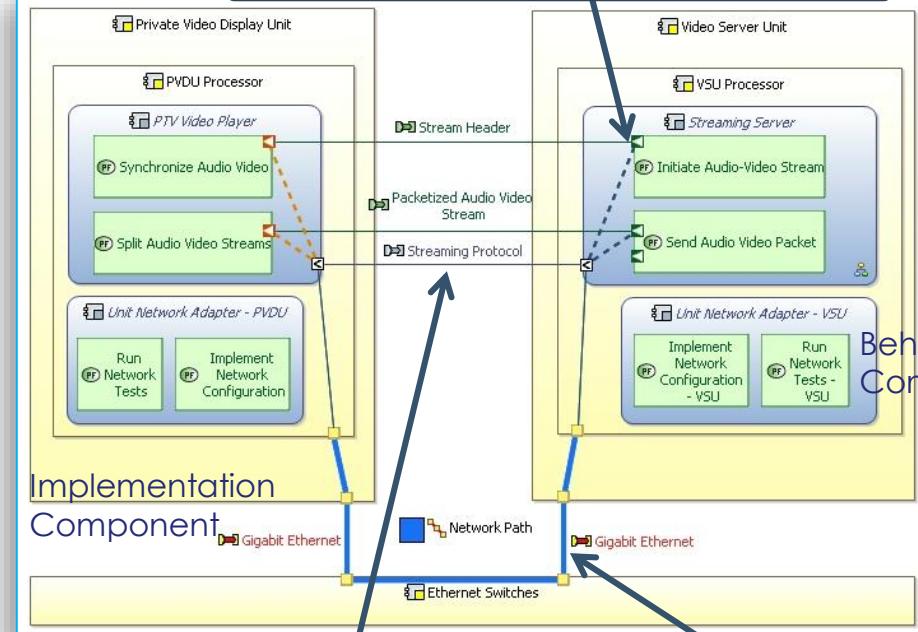


Functional analysis / Structure / Interfaces Definition Consistency

(see definitions later)

Functions are allocated to behavioral components, who use implementation components as resources

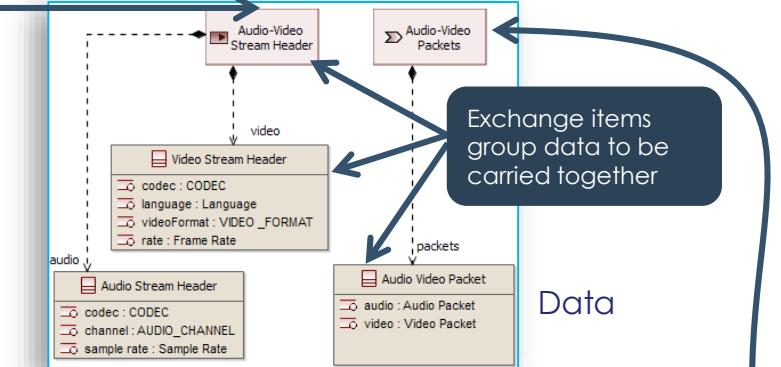
Functional exchanges & ports carry exchange items



Functional exchanges & ports are allocated to component exchanges & ports

Component exchanges & ports are allocated to Physical links/paths & ports

Exchange Item



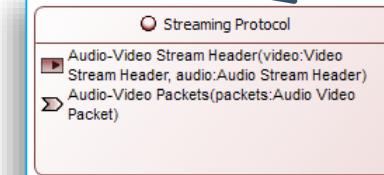
Behavioural Component

Data

Interface

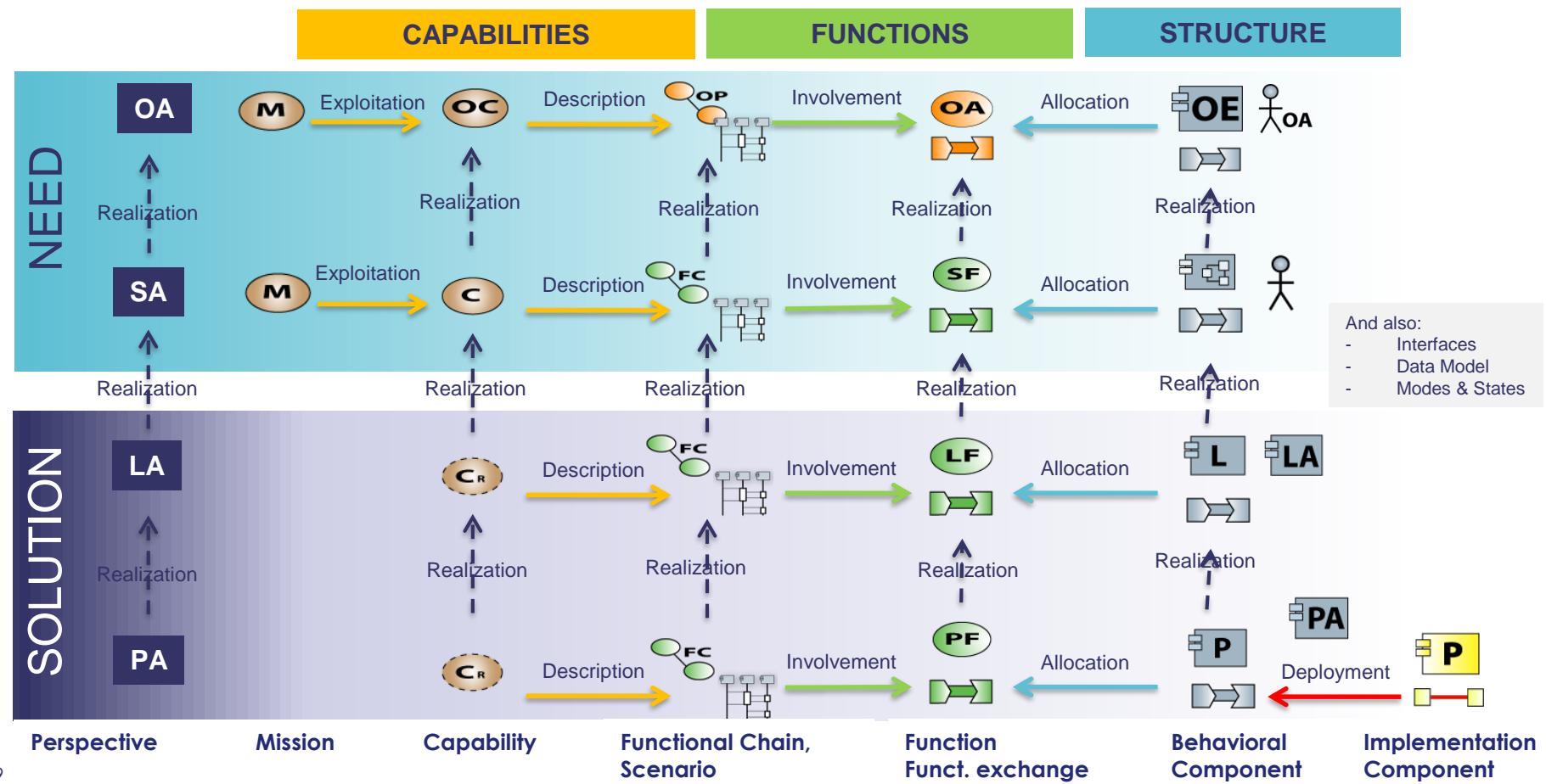
Interfaces group & reference exchange items

Interfaces are provided/required by component Ports



Relations between main concepts & Perspectives

(see definitions later)



Arcadia Language major Concepts Definition

Mission

A major goal to which the system is expected to contribute

Capability

The ability of an operational entity or of the system, to provide a service that supports the achievement of a mission

Operational Activity

An action, an operation or a service fulfilled by an operational entity, contributing to a mission

Operational Entity

A real world entity or stakeholder, involved in a mission

Operational Actor

A [usually human] non decomposable operational Entity

Operational Interaction

An interaction from an operational entity delivering exchange items to another entity (information, signal, material, torque...)

Operational Process

A logical organization of Interactions and Activities to fulfil an Operational Capability

Operational Scenario

A time-ordered set of interactions between operational activities performed by operational entities to fulfil an Operational Capability

System

An organized set of elements functioning as a unit to achieve a given purpose, subject of the engineering

System Actor

An external entity, organisation, system, or human, interacting with the system via its interfaces

Function

An action, an operation or a service fulfilled by the system, by a behavioural component or by an actor interacting with the system

Functional Port

An interaction point of a function with others, either for input or output

Functional Exchange

An interaction from a source function likely to deliver exchange items to another function (information, signal, material, torque...) through functional ports

Functional Chain

A logical organization of functions and functional exchanges to fulfil a Capability

Functional Scenario

A time-ordered set of functional exchanges between functions, to fulfil a Capability

Mode

A behaviour expected from the system or from a component or an actor in chosen conditions

State

A context undergone by the system or a component or an actor in specific circumstances

Transition

A move from one mode or state to another under some conditions

Component

A constituent of the system contributing to its behaviour and properties, by interacting with other components

Behavioural Component

A component in charge of realising some functions that are allocated to it

Behavioural Port

An interaction point of a component with others

Behavioural exchange

An interaction between two components likely to exchange some items (information, signal, material, torque...) through behavioural ports

Logical Component

A behavioural component in Logical Architecture

Implementation Component

A component hosting behavioural components, delivering them required resources

Physical Port

A connection point of an implementation component with others

Physical Link

A communication means between two implementation components through their physical ports

METHOD STEPS	TASKS	SAMPLE MODEL	CONCEPTS	DESCRIPTION MEANS
NEED	Customer Operational Need Analysis What the users of the system need to accomplish	<ul style="list-style-type: none"> ✓ Define operational capabilities ✓ Perform an operational need analysis <pre> graph LR BS[Broadcasting Station Elaborate Radio Programs] -- "Radio contents" --> ENV[ENVIRONMENT] ENV -- "News Radio channel/ Music Radio channel" --> User[USER Get news Listen to Music] </pre>	<ul style="list-style-type: none"> - Operational capabilities - Actors, operational entities - Actor activities - Interactions between activities & actors - Information used in activities & interactions - Operational processes chaining activities - Scenarios for dynamic behaviour 	<p>Dataflow: functions, op. activities interactions & exchanges</p>
	System/ SW/HW Need Analysis What the system has to accomplish for the Users	<ul style="list-style-type: none"> ✓ Perform a capability trade-off analysis ✓ Perform a functional and non-functional analysis ✓ Formalise and consolidate requirements <pre> graph LR Env[ENVIRONMENT] -- "EM waves" --> RS[RADIO SET] RS -- "Receive radio signal" --> SRS[Select radio station] SRS -- "Frequency" --> PS[Play radio contents] PS -- "Volume" --> AS[Audio sound] AS -- "Radio name" --> User[USER] </pre>	<ul style="list-style-type: none"> - Actors and system, capabilities - Functions of system & actors - Dataflow exchanges between functions - Functional chains traversing dataflow - Information used in functions & exchanges, data model - Scenarios for dynamic behaviour - Modes & states 	<p>Scenarios: actors, system, components interactions & exchanges</p>
SOLUTIONS	Logical Architecture Design How the system will work so as to fulfil expectations	<ul style="list-style-type: none"> ✓ Define architecture drivers and viewpoints ✓ Build candidate architectural breakdowns in components ✓ Select best compromise architecture <pre> graph LR Env[ENVIRONMENT] -- "EM wave" --> RR[RADIO RECEIVER] RR -- "Radio signal" --> RC[RADIO CONTROLS] RC -- "Extract radio RDS" --> UI[USER INTERFACE] UI -- "Display radio name" --> User[USER] User -- "Radio name" --> RR </pre>	<p>SAME CONCEPTS, PLUS :</p> <ul style="list-style-type: none"> - Components - Component ports and interfaces - Exchanges between components - Function allocation to components - Component Interface justification by functional exchanges allocation 	<p>Functional chains, operational processes through functions & op. activities</p>
	Physical Architecture Design How the system will be developed & built	<ul style="list-style-type: none"> ✓ Define architectural patterns ✓ Consider reuse of existing assets design a physical ✓ Design a physical reference architecture ✓ Validate and check it <pre> graph TD subgraph Physical_Architecture [Physical Architecture] direction TB AB[ANALOG BOARD] --- RRC[RADIO RECEIVER] AB --- DC[Discrete IO] AP[AUDIO PROCESSOR] --- ECR[Extract radio signal] KPC[KEYBOARD PROCESSOR] --- SRS[Select radio signal] MC[MICRO CONTROLLERS] --- CR[Control radio receiver] CR --- D[DISPLAY] RRC --- ECR ECR --- AP AP --- SRS SRS --- KPC KPC --- MC MC --- CR CR --- D end </pre>	<p>SAME CONCEPTS, PLUS :</p> <ul style="list-style-type: none"> - Behavioural components refining logical ones, and implementing functional behaviour - Implementation components supplying resources for behavioural components - Physical links between implementation components 	<p>Breakdown of functions & components</p>
	Development Contracts What is expected from each designer/ sub-contractor	<ul style="list-style-type: none"> ✓ Define a components IVVQ strategy ✓ Define & enforce a PBS and component integration contract <pre> graph TD RS[RADIO SET] --- JFR[JF Receiver] RS --- DEM[Demodulator] RS --- AP[Audio processing] RS --- BODY[Body] RS --- MR[Main Rack] RS --- UC[uController] RS --- AB[Analog board] RS --- AP2[Audio processor] RS --- CM[Cabling] JFR --- DEM DEM --- AP BODY --- MR MR --- UC UC --- AB AB --- AP2 AP2 --- CM </pre>	<ul style="list-style-type: none"> - Configuration items tree - Parts numbers, quantities - Development contract (expected behaviour, interfaces, scenarios, resource consumption, non-functional properties...) 	<p>Data model: dataflow & scenario contents, definition & justification of interfaces</p>
				<p>Allocation of op.activities to actors, of functions to components, of behav.components to impl.components, of dataflows to interfaces, of elements to configuration items</p>



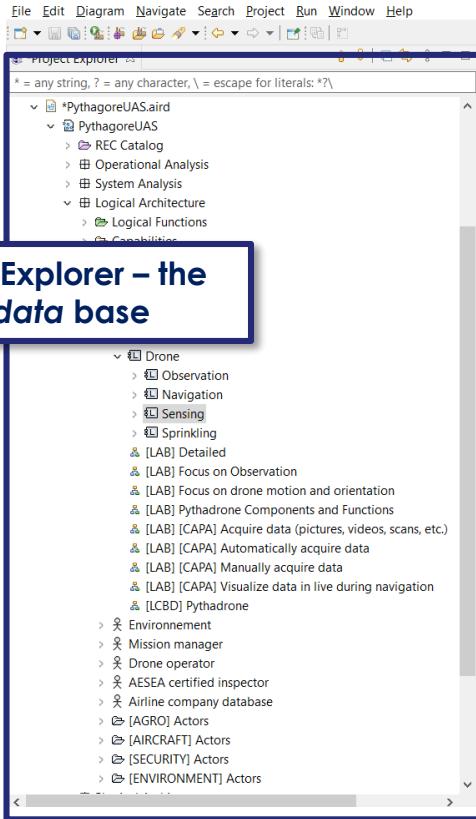
Startup: Creating, populating & exploring a Project

www.thalesgroup.com

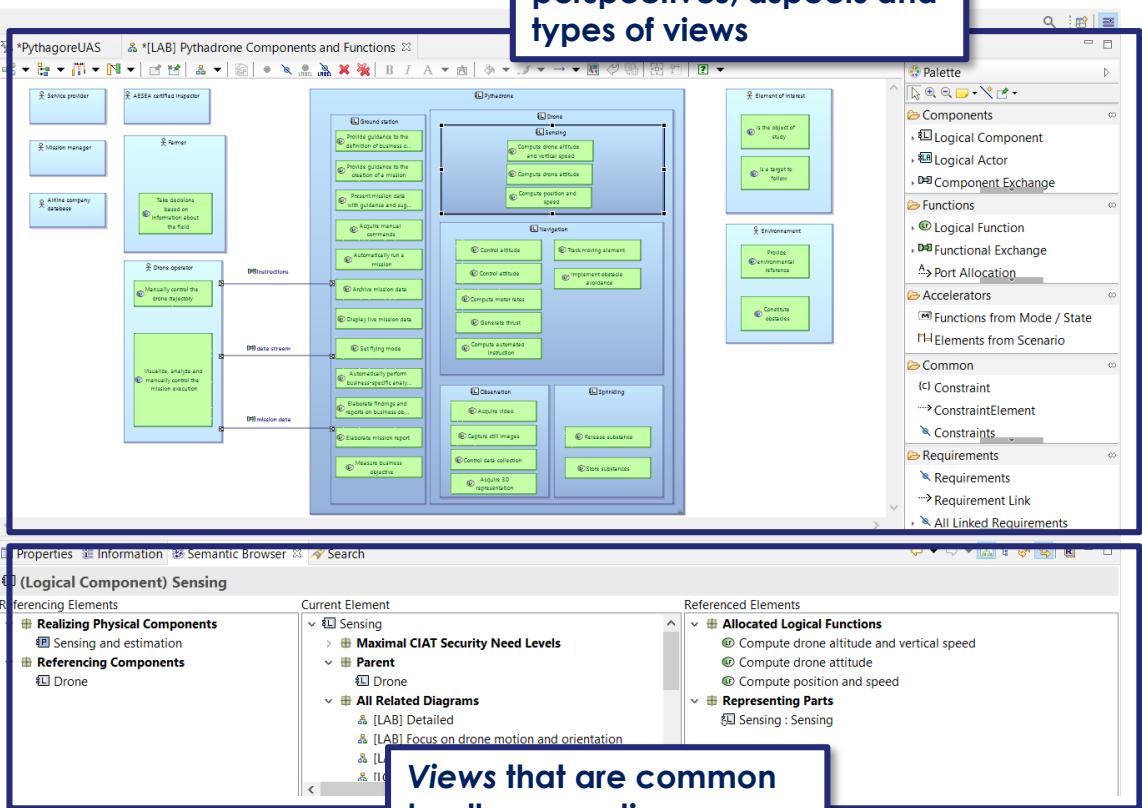
THALES GROUP INTERNAL



Capella User Interface



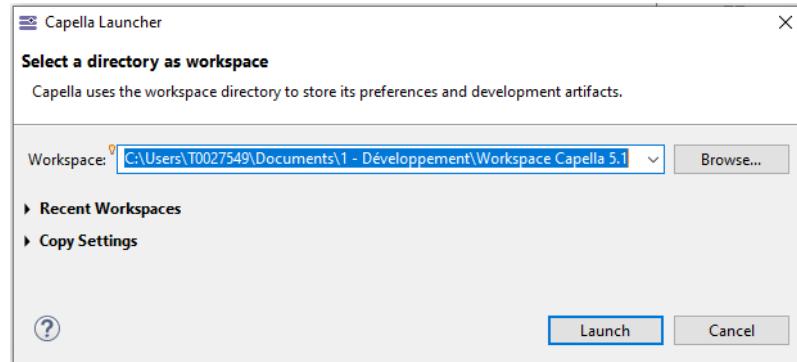
Project Explorer – the model data base



First Steps

Create a Workspace directory to store your future Projects

To change Workspace:
'File/Switch Workspace'

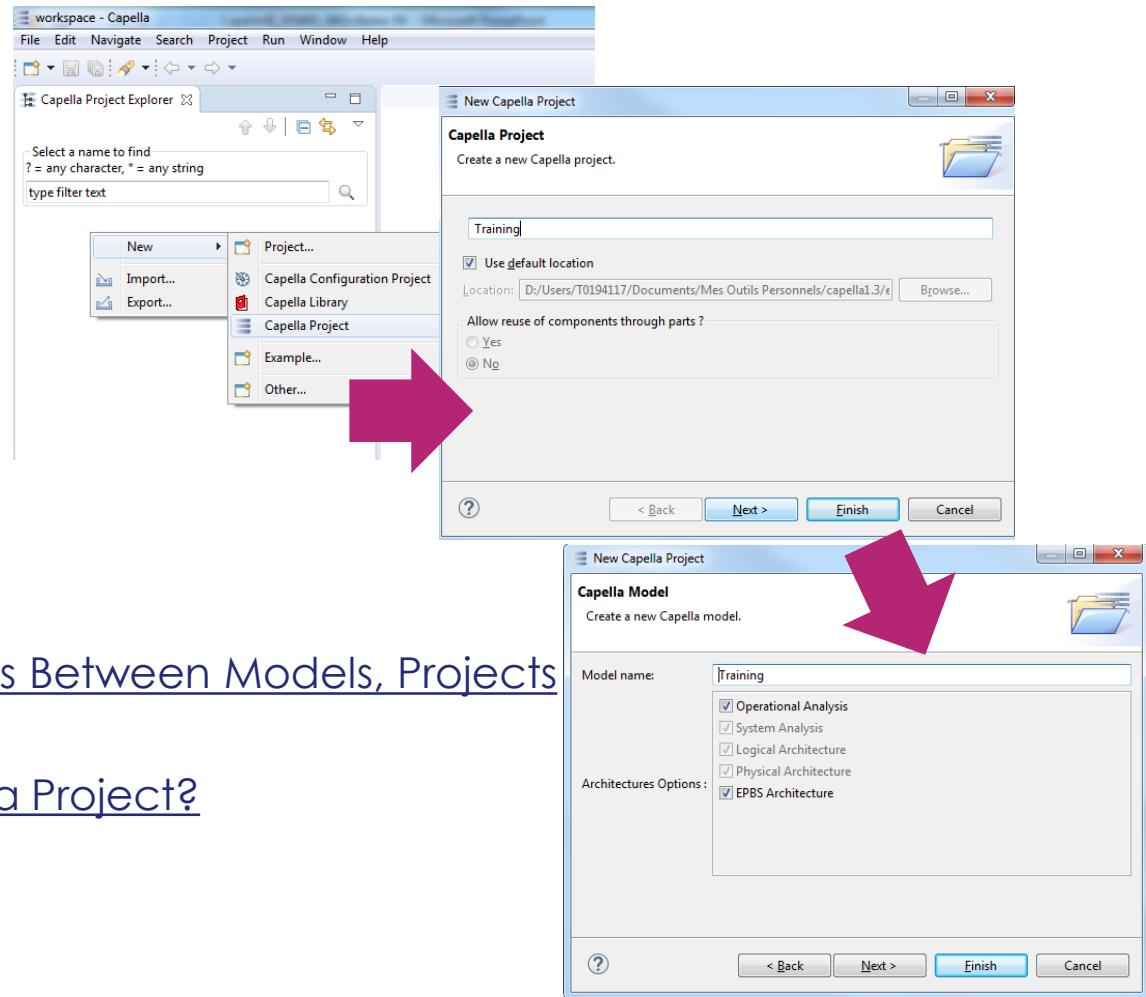


See also:

- [Understanding the Differences Between Models, Projects and Workspaces](#)
- [Starting Capella](#)

First Steps

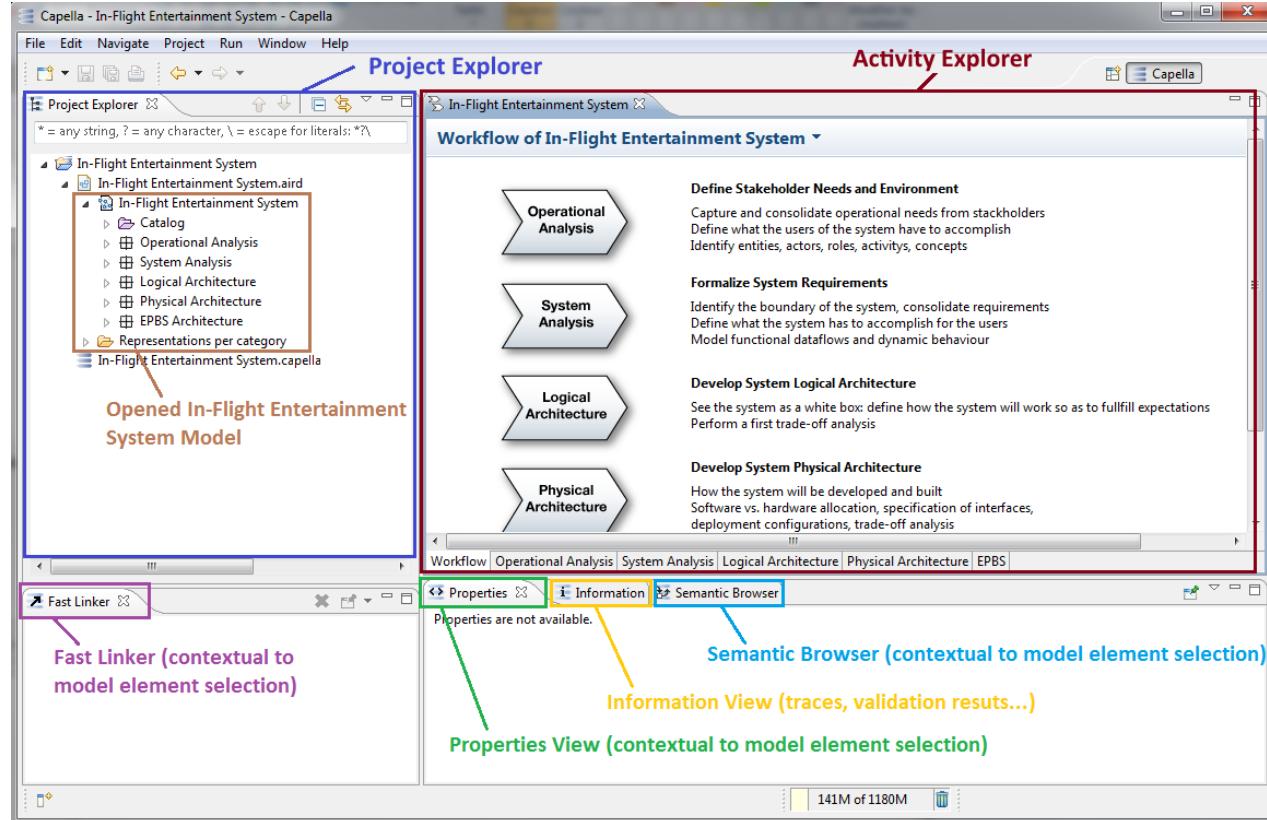
Create a Project Model



See also:

- [Understanding the Differences Between Models, Projects and Workspaces](#)
- [How to Create a New Capella Project?](#)

Capella User Interface



See also:

➤ [Capella Perspective Overview](#)

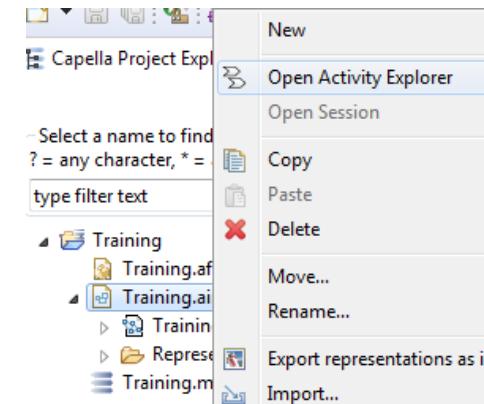
Capella User Interface

I Start from Project Explorer

- Gives access to models, model elements, diagrams

I Create and browse diagrams using Activity Explorer

- To open Activity explorer:
Right click on .aird file, 'Open Activity Explorer'



I All Windows/views can be resized, moved and closed

- To retrieve the default position of views: Window / Perspective / Reset Perspective

See also:

- Project Explorer, Workbench Manipulation - Tips and Tricks

Activity Explorer: methodological Guidance & Diagram Management

This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2020 All rights reserved.

The screenshot shows the Activity Explorer interface for the In-Flight Entertainment System. On the left, a sidebar displays the 'Workflow of In-Flight Entertainment System' with steps: Operational Analysis, System Analysis, Logical Architecture, Physical Architecture, and EPBS. The 'System Analysis' step is highlighted with a pink circle and a large pink arrow pointing to it from the top-left. Below the sidebar, the main workspace shows the 'System Analysis' activity with three parallel arrows: Operational Analysis, System Analysis (highlighted in yellow), and Logical Architecture. A blue arrow points from the 'Operational Analysis' arrow to the 'System Analysis' arrow, labeled 'Current perspective'. To the right of the workspace is a 'Diagrams Viewer' window titled 'Diagrams Viewer' with a 'type filter text' input field. It lists existing diagrams categorized by type: Common (Class Diagram Blank, Exchange Scenario, Functional Chain Description, Function Scenario, Modes and States [DEPRECATED]), System Analysis (Contextual Capability, Contextual Mission, [CM] Provide Cabin Management Solutions, [CM] Provide Entertainment Solutions, Contextual System Actors, Missions Capabilities Blank, System Architecture Blank, [SAB] High Level System Overview, [SAB] Top Level System Overview, System Data Flow Blank, System Function Breakdown, System Functions - Operational Activities). A blue arrow points from the 'Create new diagrams for different activities' text to the 'Diagrams Viewer' window.

Workflow of In-Flight Entertainment System

Operational Analysis

System Analysis

Logical Architecture

Physical Architecture

EPBS

Define Stakeholder

Capture and consolidate requirements. Define what the users want. Identify entities, actors and their interactions.

Formalize System Requirements

Formalize system requirements. Define functional requirements, system boundaries, data flows, interfaces and system behavior.

Develop System Logic

See the system as a whole. Perform a first trade-off analysis.

Develop System Physics

How the system will be deployed. Software vs. hardware deployment configuration.

Formalize Components

Manage industrial criticality expected from each diagram. Specify requirements.

Transition From Operational Activities

Define Actors, Missions and Capabilities

Refine System Functions, describe Functional Exchanges

[SFBD] Create a new Functional Breakdown diagram

[SDFB] Create a new Functional Dataflow Blank diagram

[FS] Create a new Functional Scenario

Allocate System Functions to System and Actors

[SAB] Create a new System Architecture diagram

[ES] Create a new Exchange Scenario

System Analysis

Operational Analysis

Logical Architecture

Diagrams Viewer

Common

- Class Diagram Blank
- Exchange Scenario
- Functional Chain Description
- Function Scenario
- Modes and States [DEPRECATED]

System Analysis

- Contextual Capability
- Contextual Mission
 - [CM] Provide Cabin Management Solutions
 - [CM] Provide Entertainment Solutions
- Contextual System Actors
- Missions Capabilities Blank
- System Architecture Blank
 - [SAB] High Level System Overview
 - [SAB] Top Level System Overview
- System Data Flow Blank
- System Function Breakdown
- System Functions - Operational Activities

Create new diagrams for different activities

Existing diagrams sorted by type

See also:

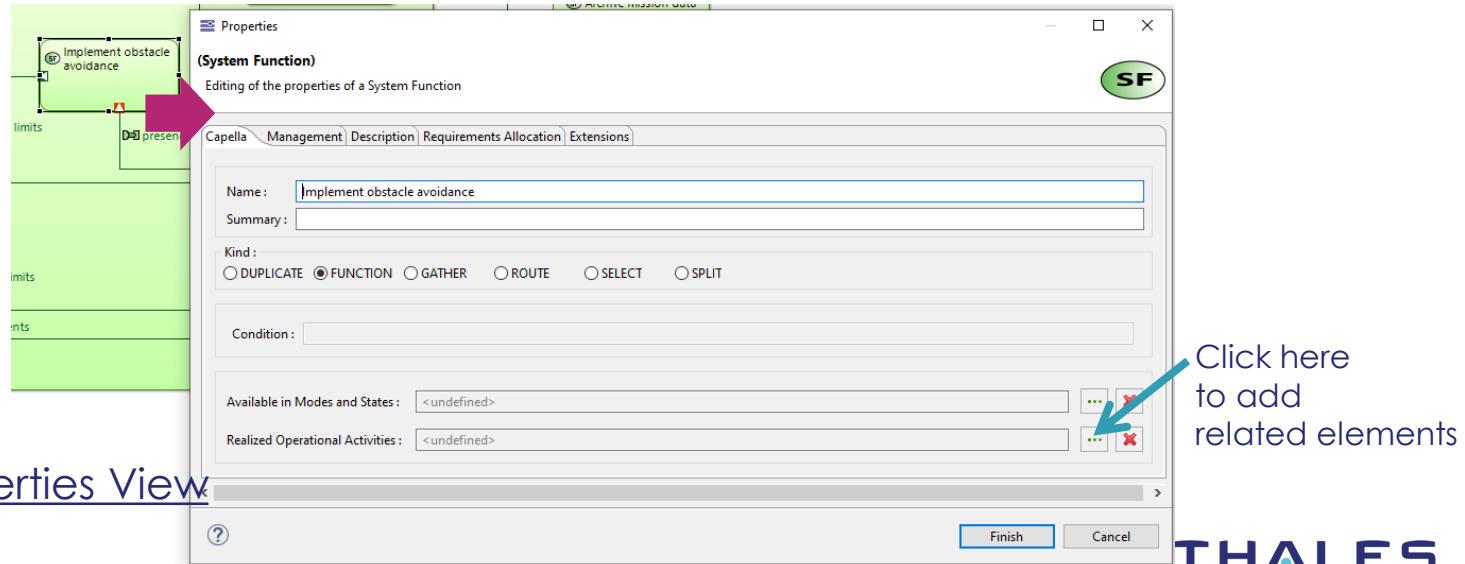
➤ [Activity Explorer](#)

Editing Information & Properties on a Model Element

This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2020 All rights reserved.

Applicable to any element shown in Project Explorer, Diagrams, Semantic Browser...

| Double-left-Click on an element to view and modify its properties



See also:

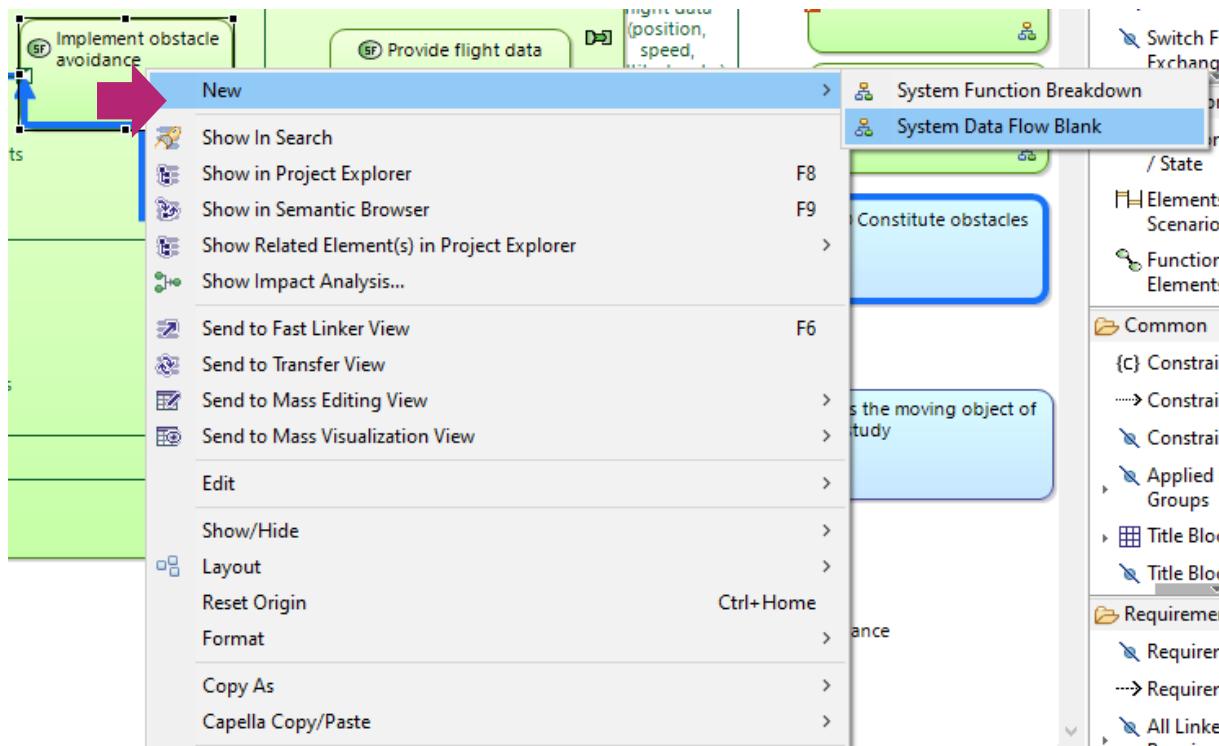
➤ [Capella Properties View](#)

Editing Information & Properties on a Model Element

This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2020 All rights reserved.

Right-Clic on an element to manipulate and exploit it

- 'New' allows creating other model elements or diagrams related to this element

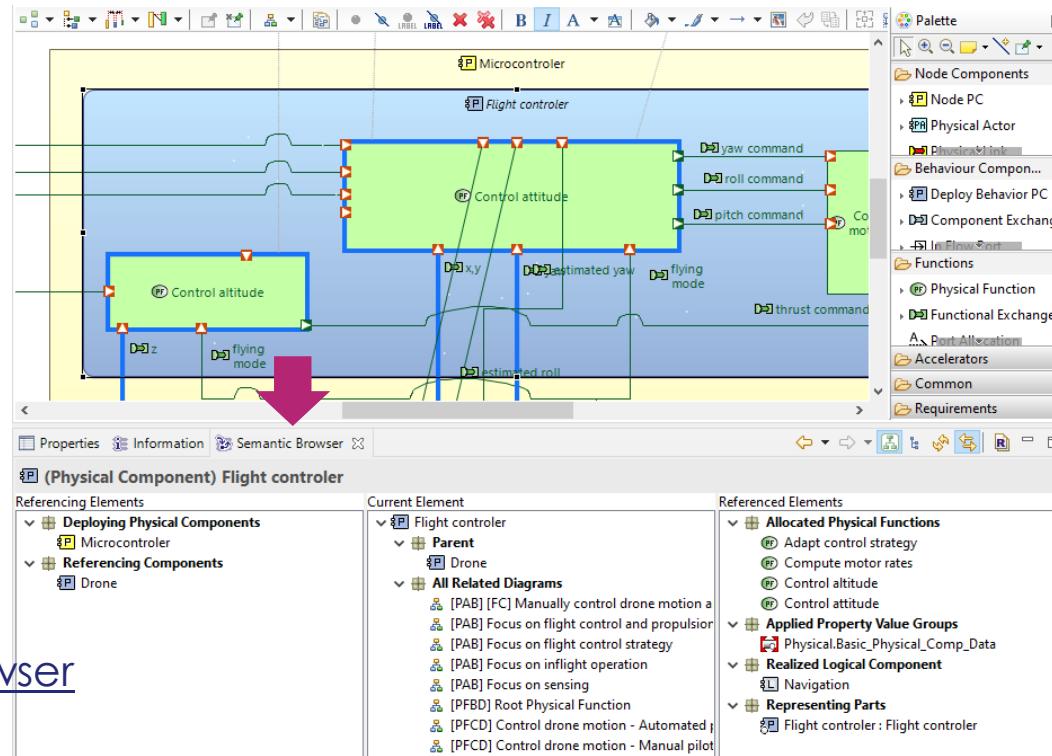


Editing Information & Properties on a Model Element

This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2020 All rights reserved.

| Clic on an element and press 'F9' to explore related elements in Semantic Browser

> navigate through them inside Semantic Browser, using 'F9' & right-click Menu



See also:

> [Semantic Browser](#)



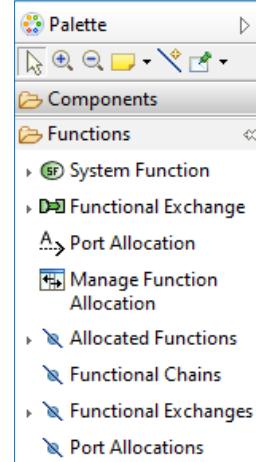
Creating and editing Diagrams



Tips for Diagram Edition

Create elements from the right side palette tools

- They are created both in the diagram and in the model



Insert in the diagram existing model elements previously created

- Using palette tools prefixed by ✎ symbol

Customise or filter Display using toolbar



See also:

Capella Diagram Editor



Create a new model element and display it in current diagram



Insert/ remove an existing model element in the current diagram

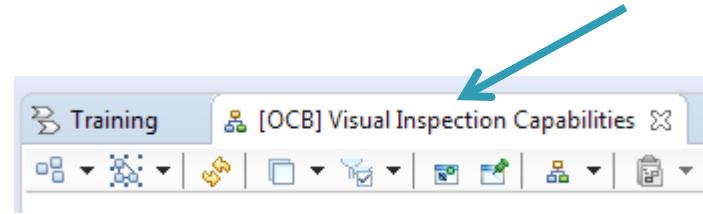
Tips for Diagram Edition

| Double click on diagram tab:
diagram in full screen

- Do it again to recover other views

| Hold « ctrl » key : use the same palette tool several times

| F2: rename an element



Hide, delete from diagram or model (in diagram toolbar)

Hide an element



- Element is still in the diagram, but not visible
- Can be displayed later on with same position, color...

To be avoided
=> can have impact on performances

Delete from diagram

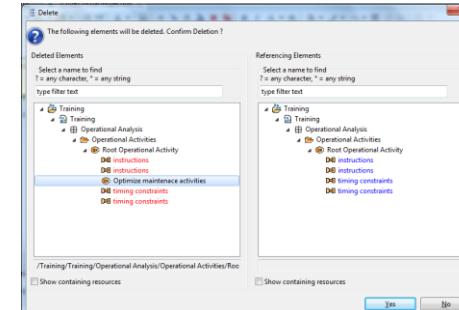


- Element is removed from diagram (but still in the model, may be displayed elsewhere)
- Can be displayed later on but position and style has to be redefined

Delete from model

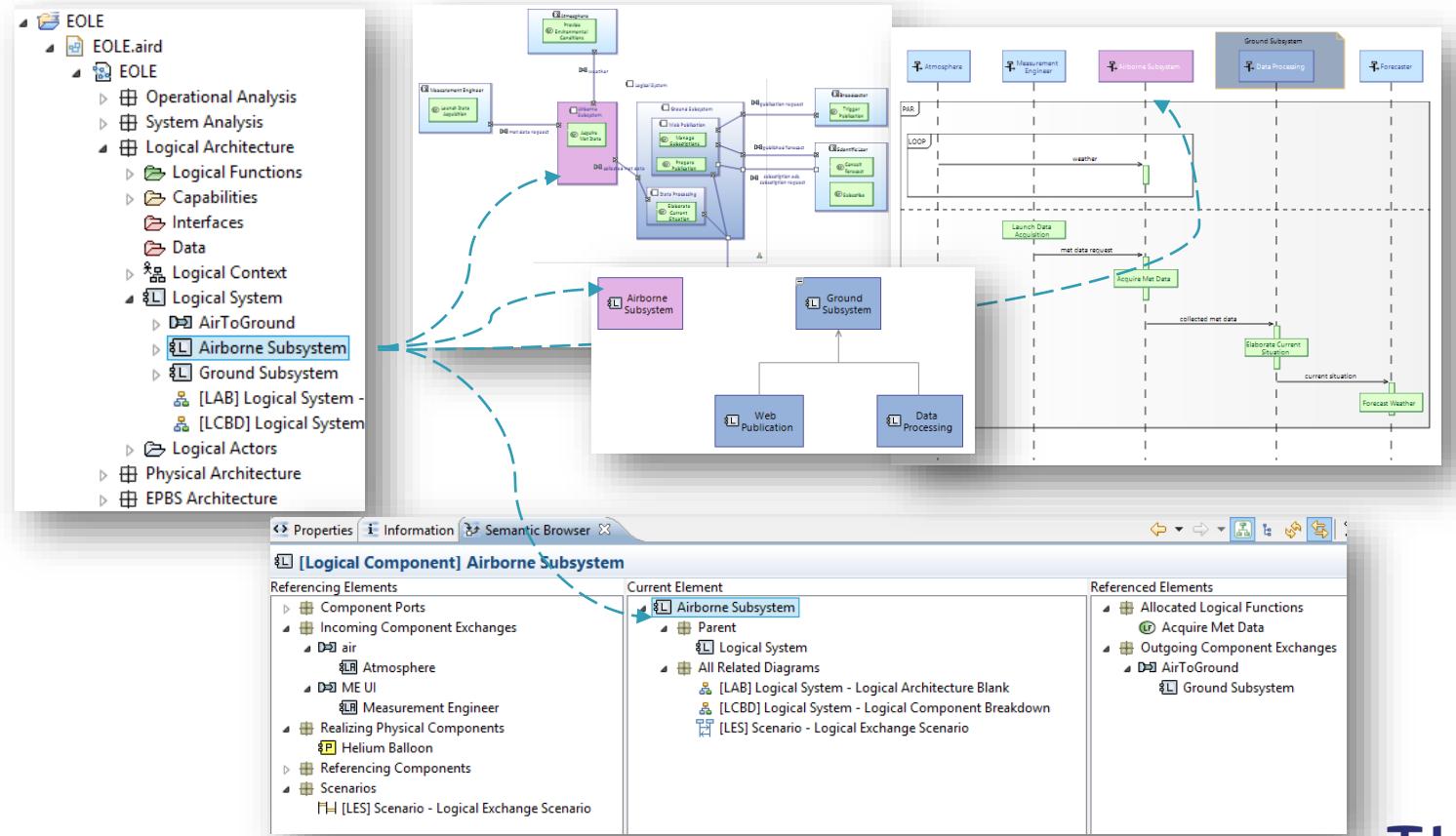


- Element is deleted => confirmation window with impacts
- Its representation from all diagrams is removed



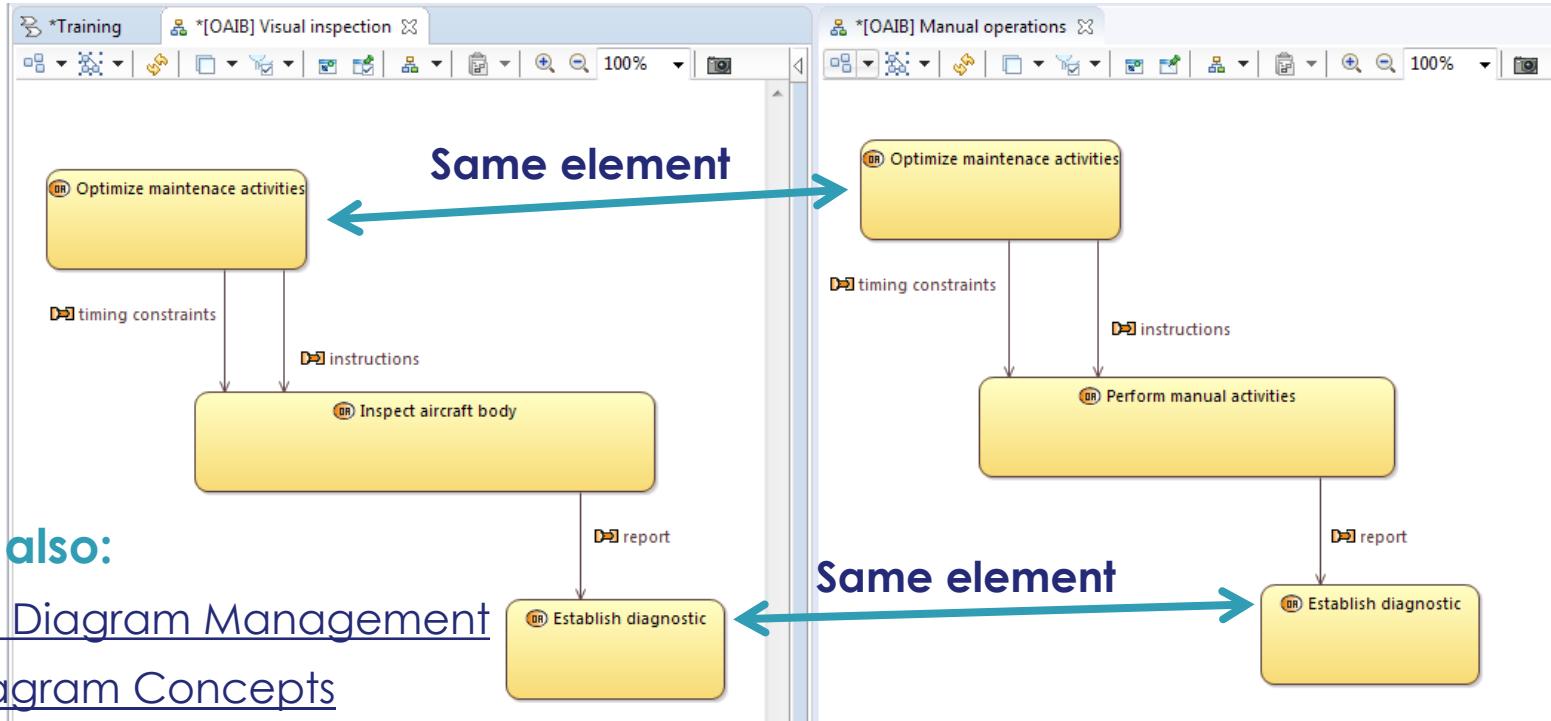
Diagrams only display Model Contents (Model Elements)

This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2020 All rights reserved.



Diagrams only display Model Contents (Model Elements)

| Renaming or modifying in one diagram renames or modifies everywhere (in the Model and in other Diagrams)



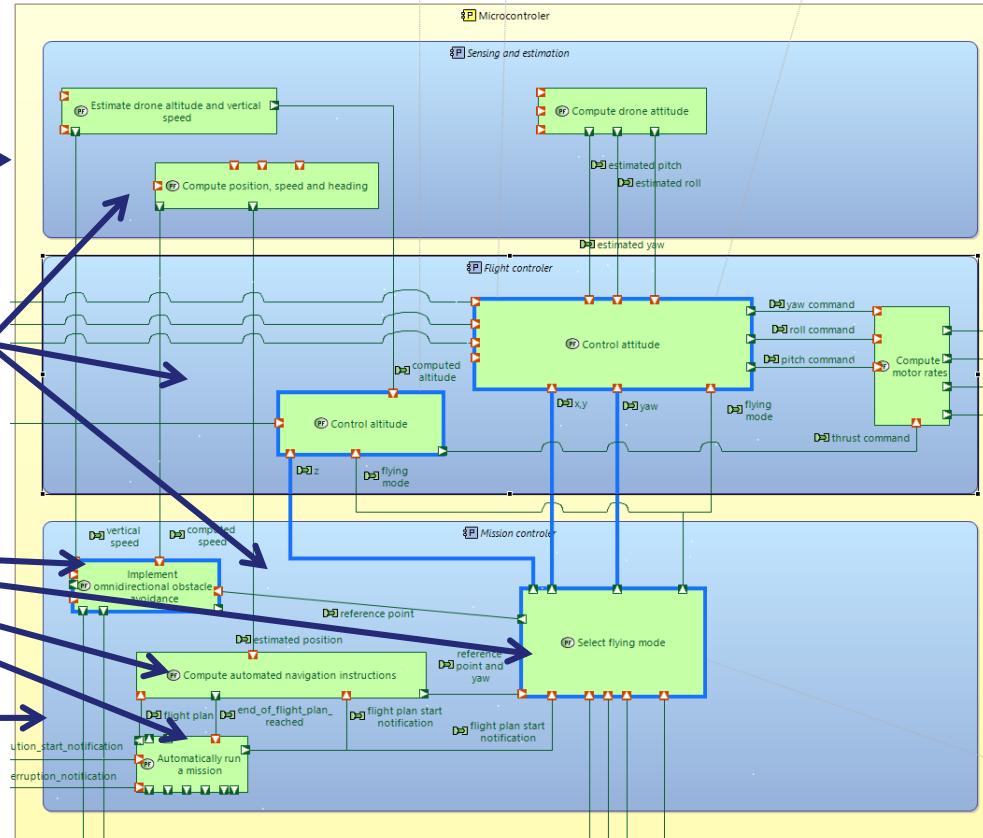
| See also:

- > [05. Diagram Management](#)
- > [Diagram Concepts](#)

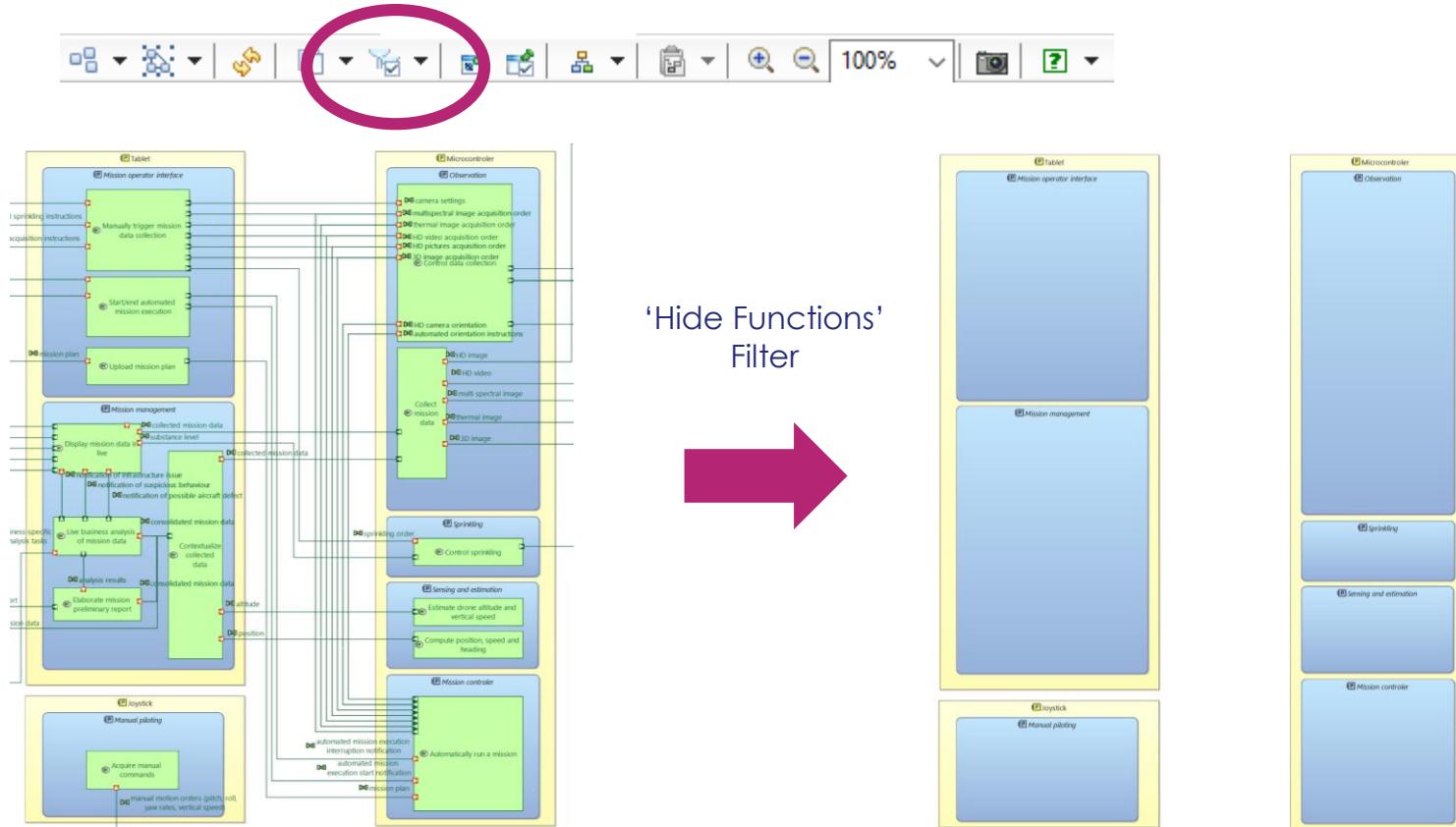
A box inside another box is meaningful!

In most diagrams, allocation or composition relations are depicted by visual containment

This Resource Component →
implements
These behavioral Components
These Functions
are allocated to
This behavioral Component



Capella Features sample: visual Filters

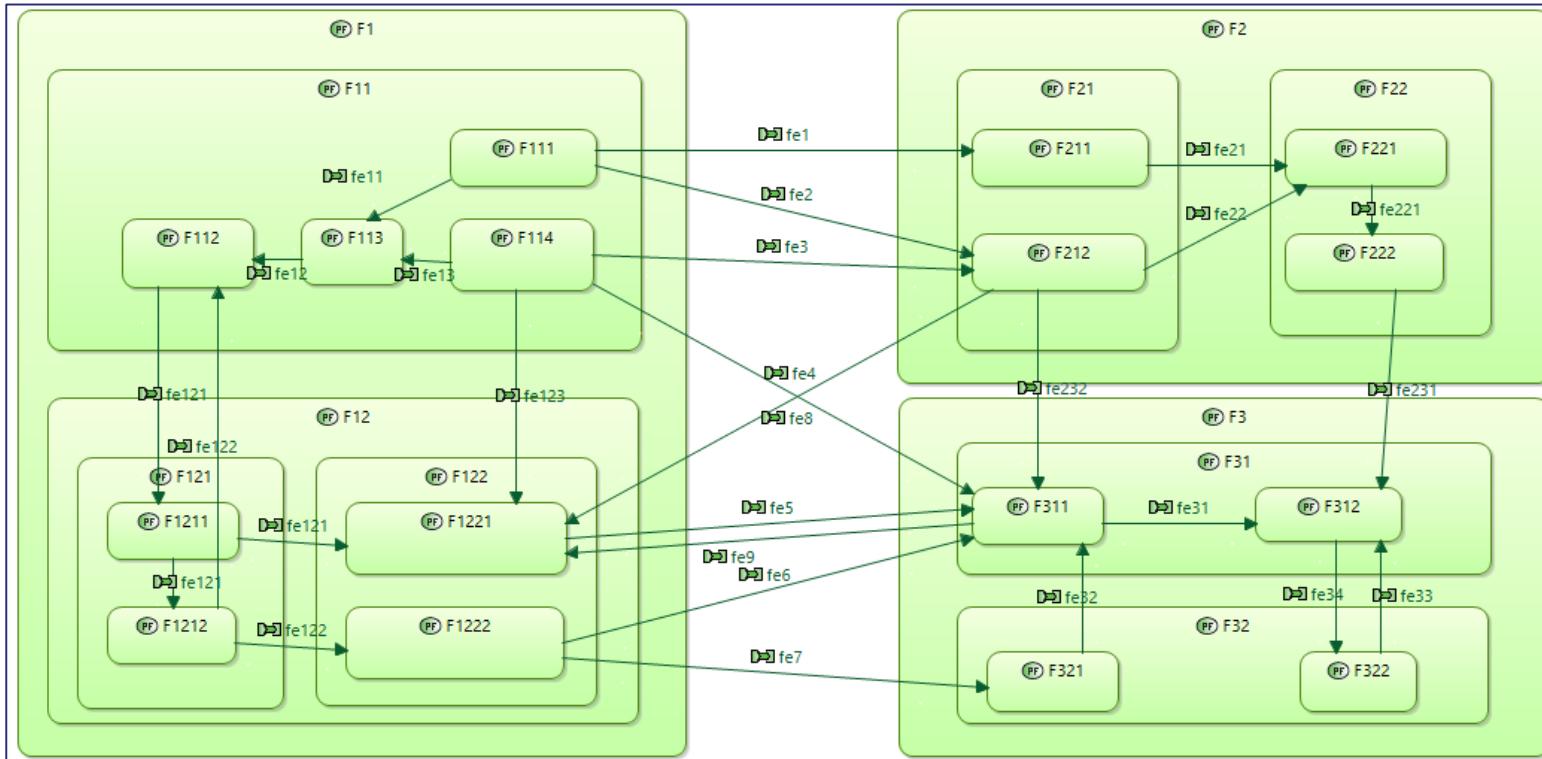


'Hide Functions'
Filter

Capella Features sample: visual Synthesis Capabilities

This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2020 All rights reserved.

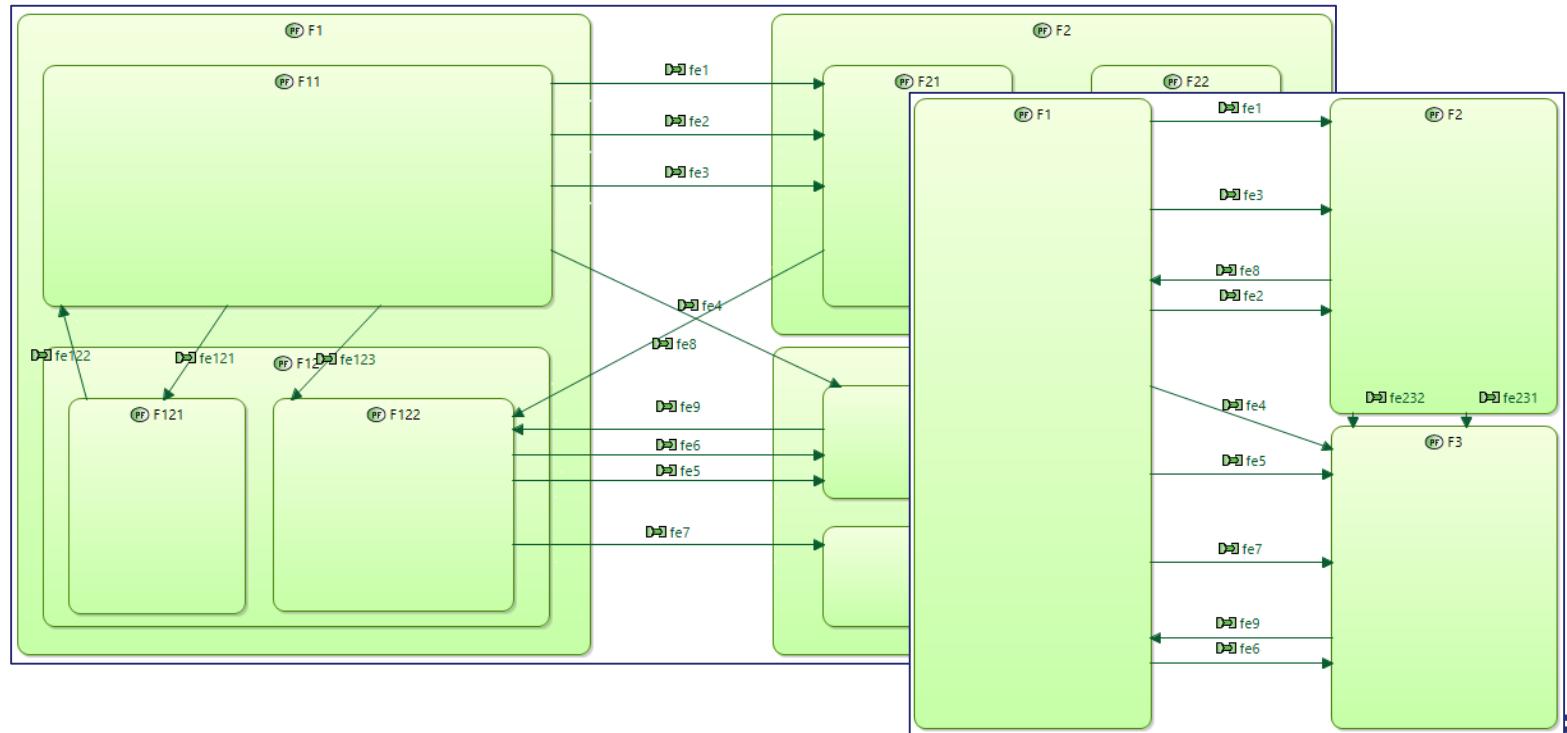
■ Exchanges shall be allocated to leaf elements (at least for functions)



Capella Features sample: visual Synthesis Capabilities

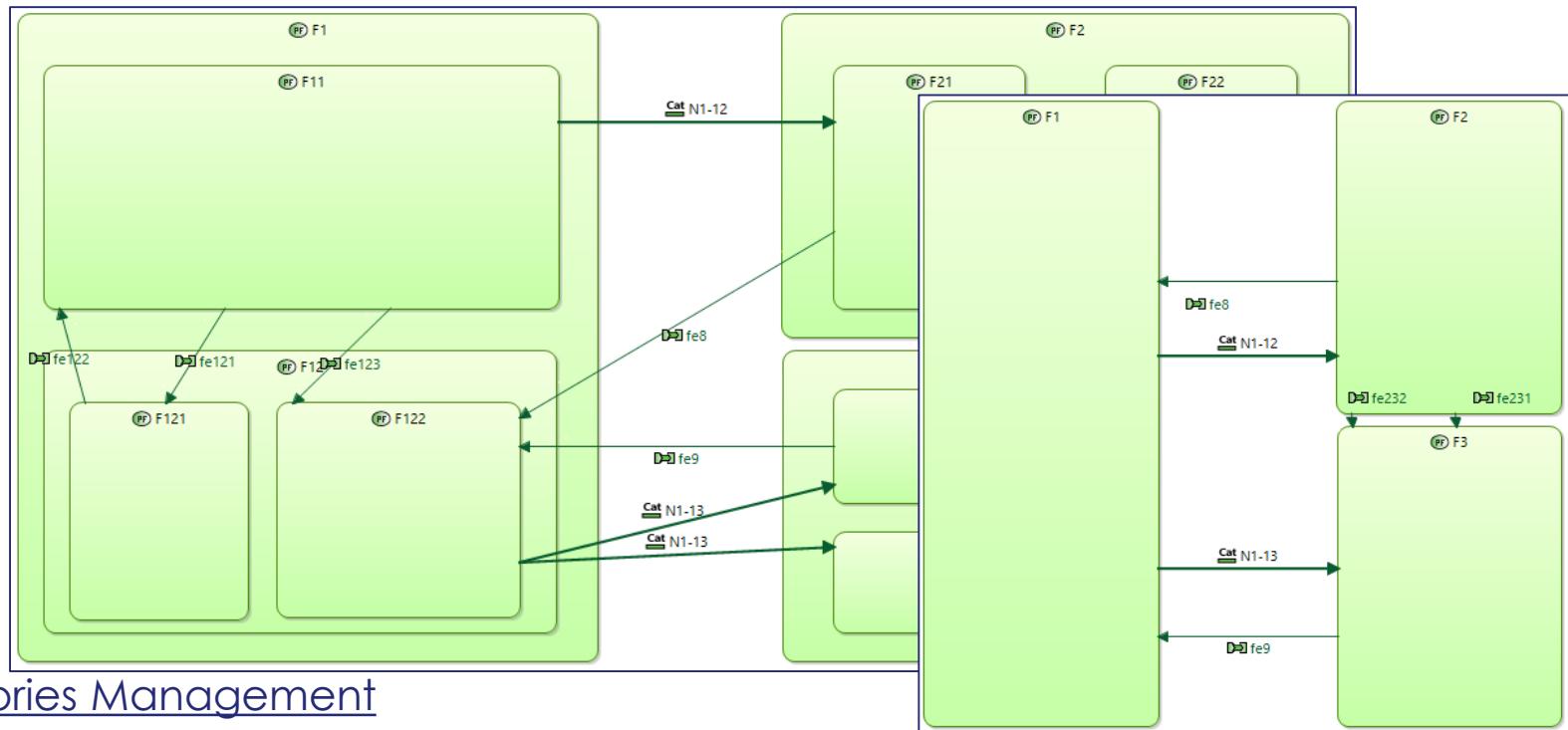
This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2020 All rights reserved.

| Select the level of detail, the tool will synthesise lower level exchanges allocation



Capella Features sample: visual Synthesis Capabilities

| Grouping exchanges in 'Categories' allows automatic display synthesis of several exchanges into one single category





Specific Diagram Editing

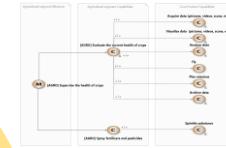


Capella Diagrams Types – all views are kept consistent by the tool

This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2020 All rights reserved.

[OCB] Operational Capabilities
[MCB] Missions & Capabilities
[FCD] Functional Chain
[xS] Scenarios

Purpose
the system's why



[xAB] Architecture
[CDB] Class Diagram



Structure
the system is

See also:

- [Typology of Capella Diagrams, Diagrams](#)

[xDFA] Data flow
[MSM] Modes & States



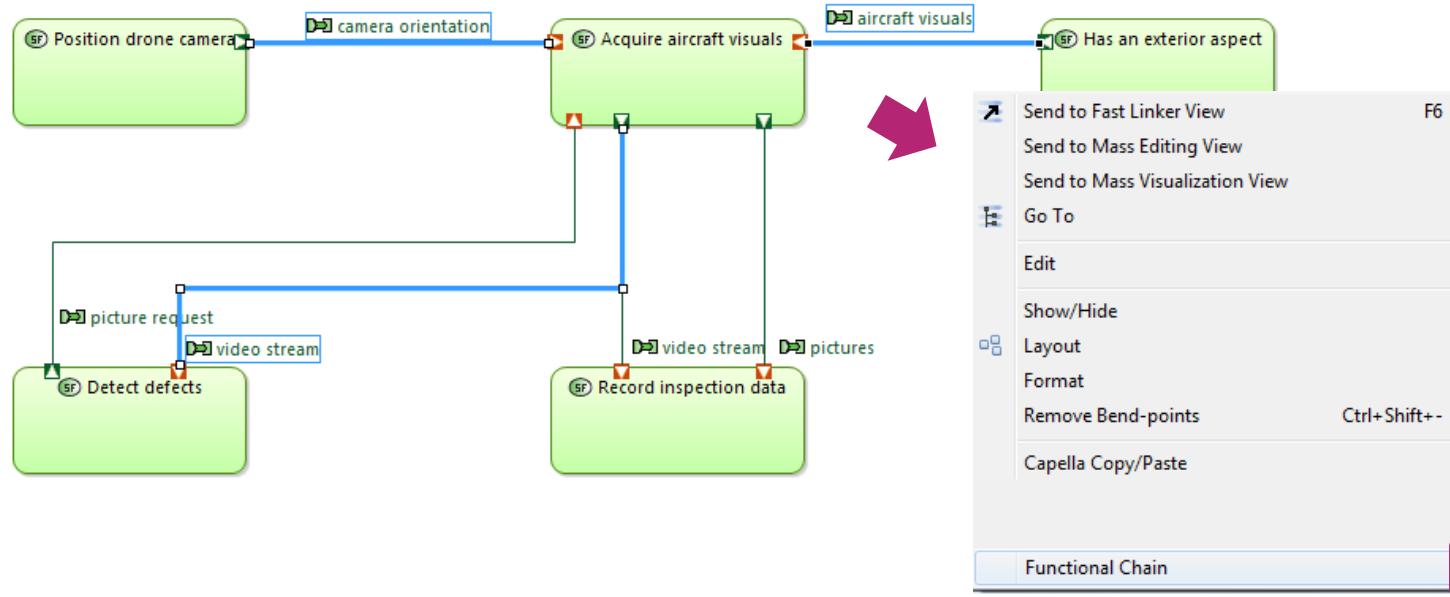
Behavior
the system does

Capella Diagrams by Perspectives & Aspects [Capella acronyms]

	ASPECTS				
	Purpose	Function	Behavior	Structure	Interfaces
Operational Analysis What the stakeholders need to accomplish	[OCB] Operational Capabilities & Missions [OPD] Operational Process description [OES/OAS] Operational Entities/Activities Scenario	[OAIB] Operational Activities Interactions [OABD] Operational Activities Breakdown	[MSM] Modes or States Machines	[OEBD] Operational Entities Breakdown [OAB] Operational Architecture [ORB] Operational Roles	[CDB] Data/Classes
System Needs Analysis What the system has to accomplish for the stakeholders	[MCB] Missions & Capabilities [SFCD] System Functional Chain description [ES/FS] (system) Exchanges/Functions Scenario	[SDFB] System (Functions) dataflow [SFBD] System Functions Breakdown	[MSM] Modes or States Machines	[CSA] Context System & Actors [SAB] System (Need) Architecture	[CDB] Data/Classes [CEI/CDI] Contextual External/Detailed Interfaces [IS] Interface Scenario
Conceptual Architecture How the system will work to fulfill expectations	[CRB] Capability Realizations [LFCD] Logical Functional Chain description [ES/FS] (components) Exchanges/Functions Scenario	[LDFB] Logical (Functions) dataflow [LFBD] Logical Functions Breakdown	[MSM] Modes or States Machines	[LAB] Logical Architecture	[CDB] Data/Classes [CLI] Contextual Internal Interfaces [IS] Interface Scenario
Finalized Architecture How the system will be developed and built	[CRB] Capability Realizations [PFCD] Physical Functional Chain description [ES/FS] (components) Exchanges/Functions Scenario	[PDFB] Physical (Functions) dataflow [PFBD] Physical Functions Breakdown	[MSM] Modes or States Machines	[PAB] Physical Architecture	[CDB] Data/Classes [CLI] Contextual Internal Interfaces [IS] Interface Scenario

Creating a functional Chain

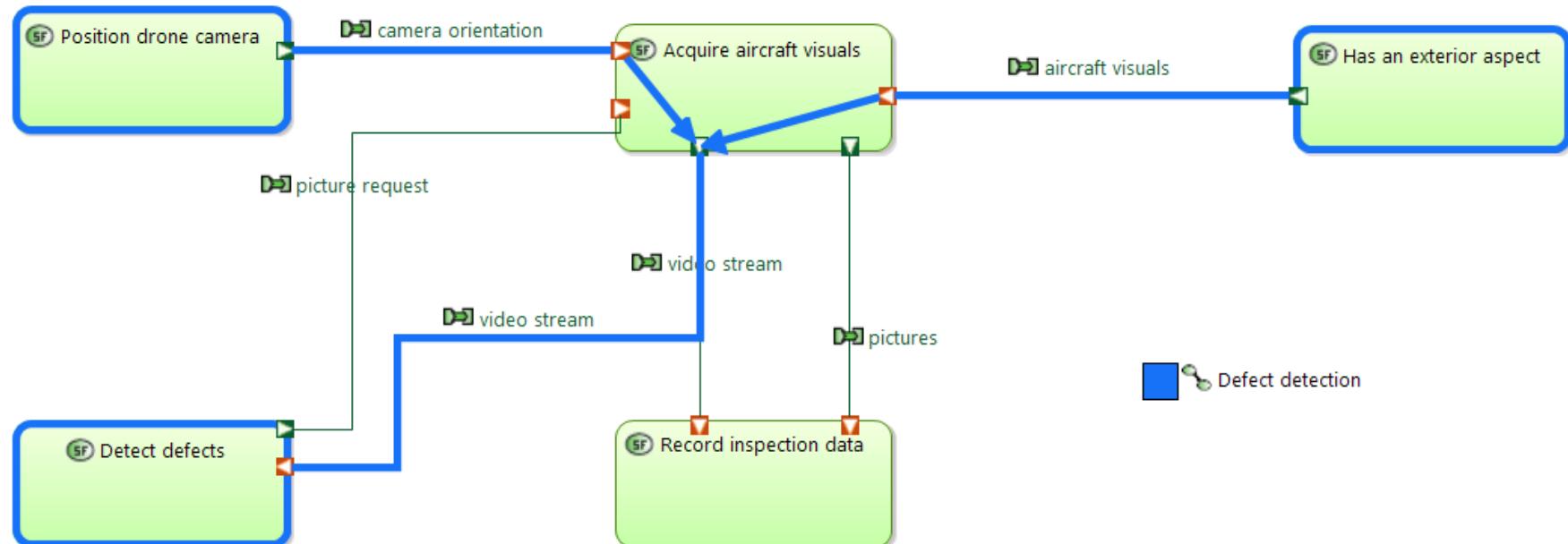
In a dataflow ([xDFB] or [xAB]), select an ordered set of functional exchanges, then initialise the functional Chain



Creating a functional Chain

| The « footprint » of the functional chain is displayed in the diagram

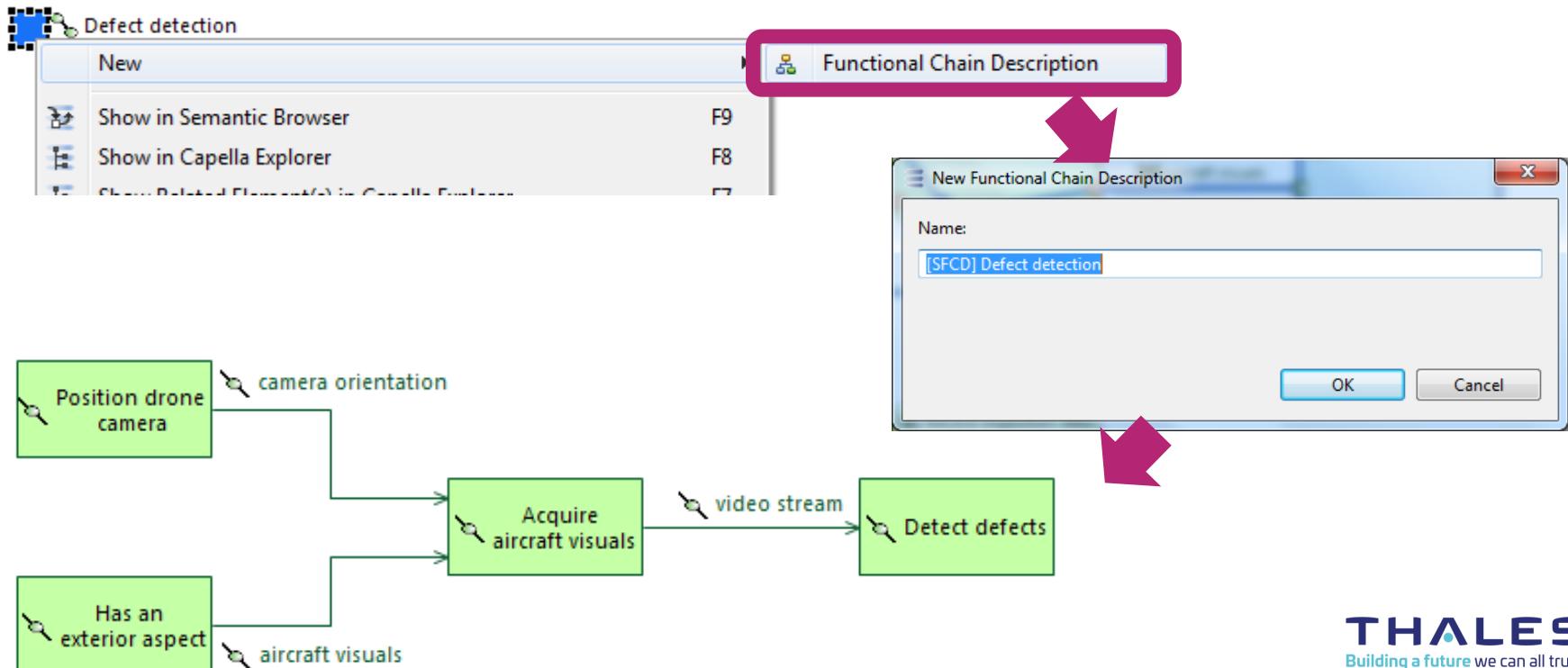
| The functional chain element in the model is also displayed:



Creating a functional Chain

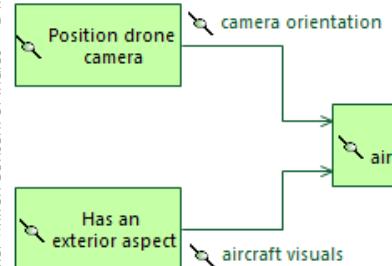
This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2020 All rights reserved.

| For further modifications of the functional Chain,
create a dedicated diagram



Creating a functional Chain

The tool will suggest possible functions or exchanges to complement the functional chain



Screenshot of the Functional Chain tool interface. The palette on the right shows options: Function (highlighted), Exchange, Exchange and Function, Functional Chain, and Connect Functions. A Selection Dialog window is open, showing a tree view of system components under 'PythagoreUAS' and a list of notifications under 'Root System Function'.

Selection Dialog

Selection Wizard

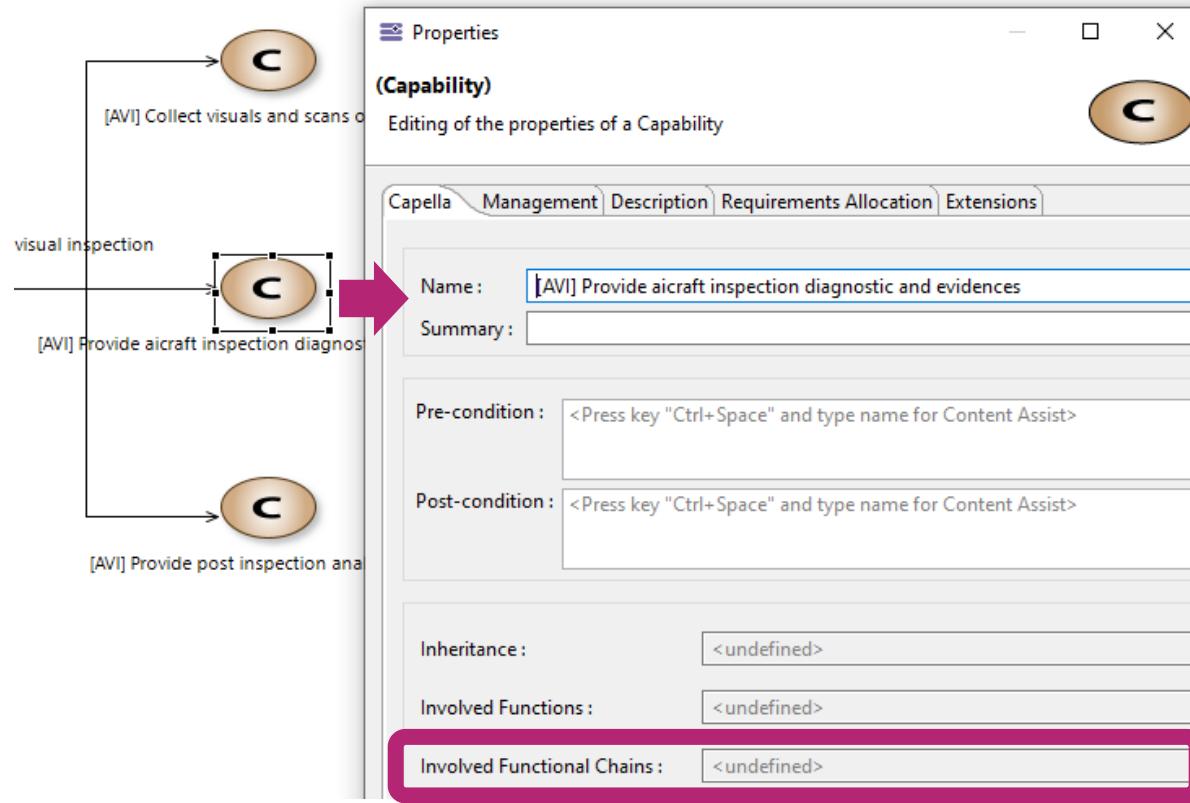
Select a value to involve.

- PythagoreUAS
 - PythagoreUAS
 - System Analysis
 - System Functions
 - Root System Function
 - notification of possible aircraft defect [Automatically perform business-specific analysis of collected data -> Display mission data in live]
 - notification of infrastructure issue [Automatically perform business-specific analysis of collected data -> Display mission data in live]
 - notification of suspicious behaviour [Automatically perform business-specific analysis of collected data -> Display mission data in live]

OK Cancel

Creating a functional Chain

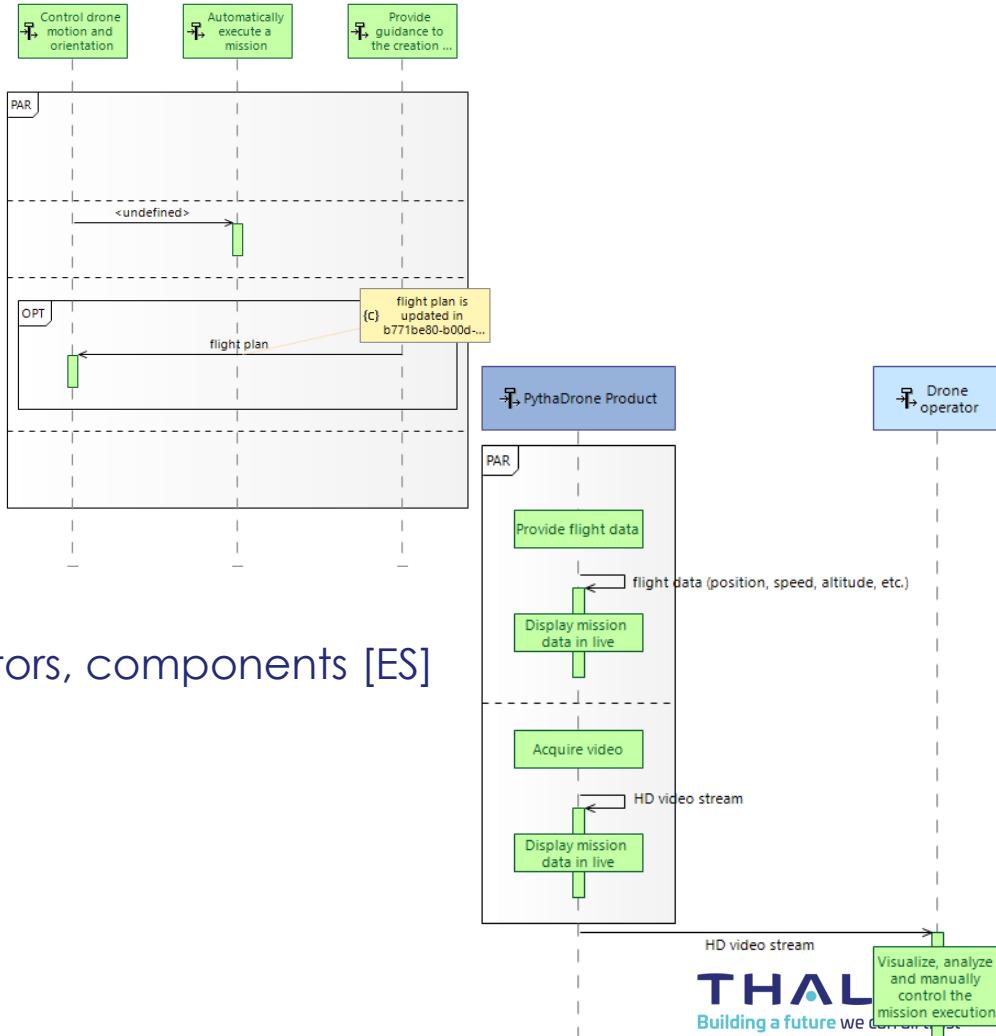
| Capabilities described by the functional Chain have to reference it



Creating a Scenario

Different kinds of Scenarios:

- Interactions between Functions [FS]

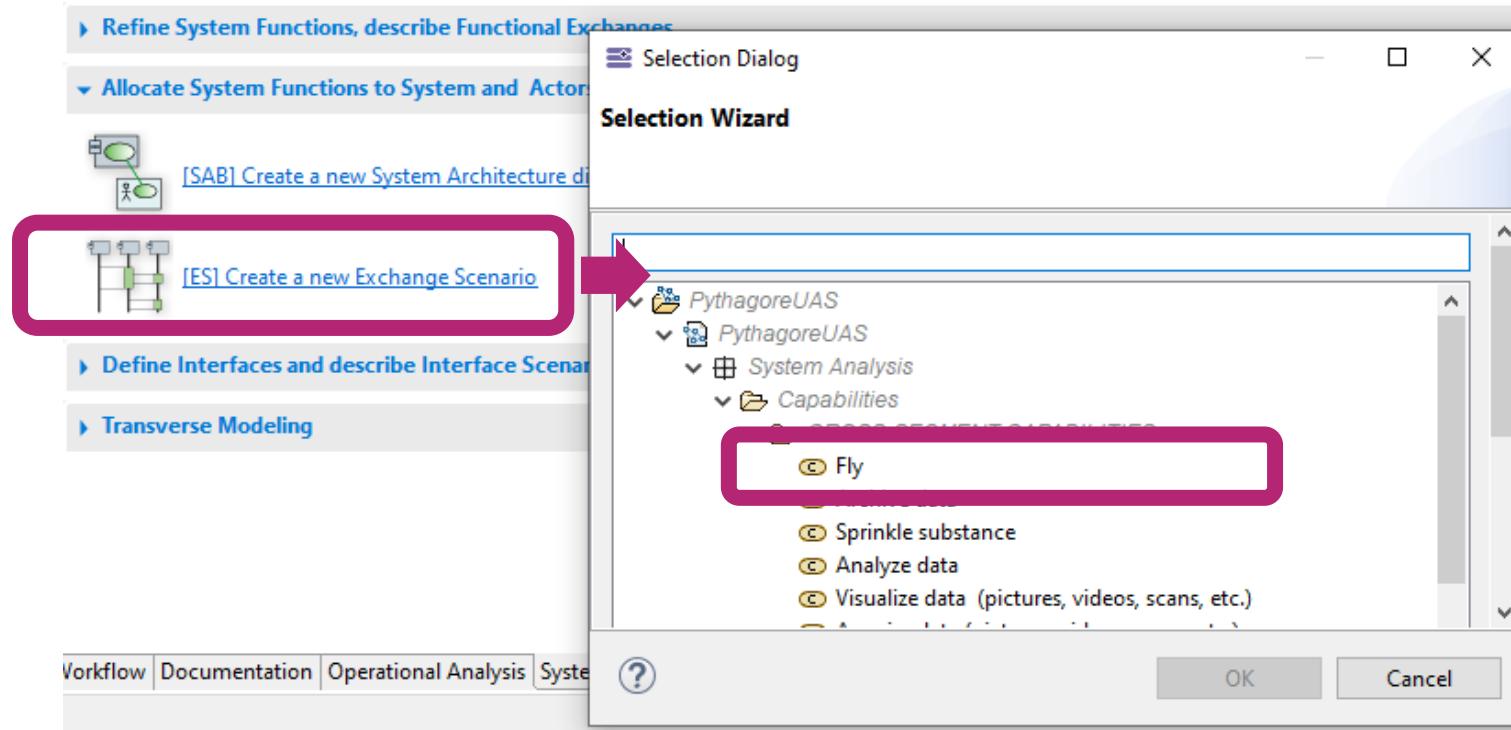


- Interactions between the system, actors, components [ES]

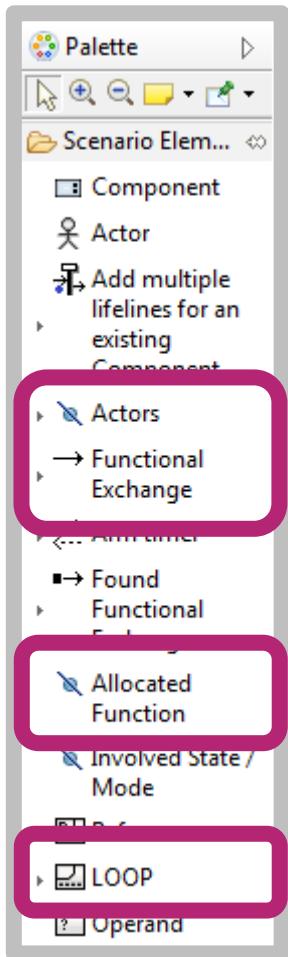
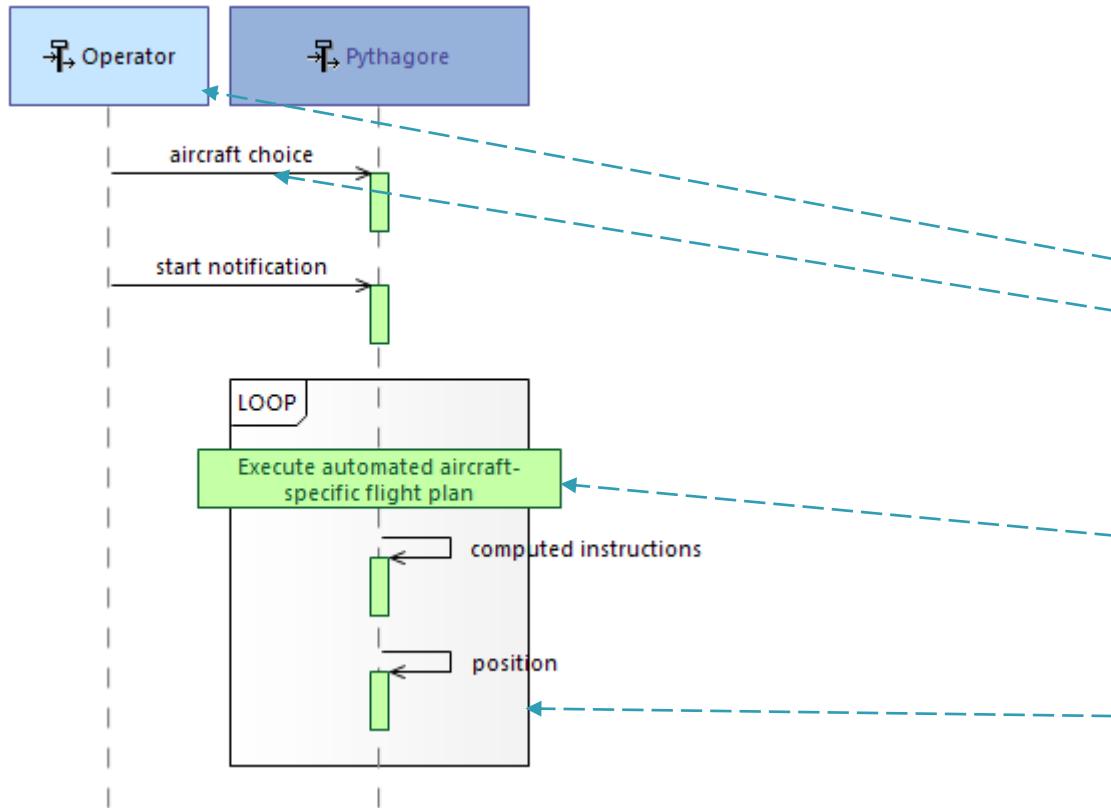
Creating a Scenario

This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2020 All rights reserved.

| Select the kind of scenario in Activity Explorer, then select the main Capability that the scenario will describe



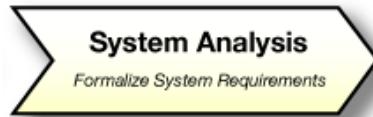
Creating a Scenario



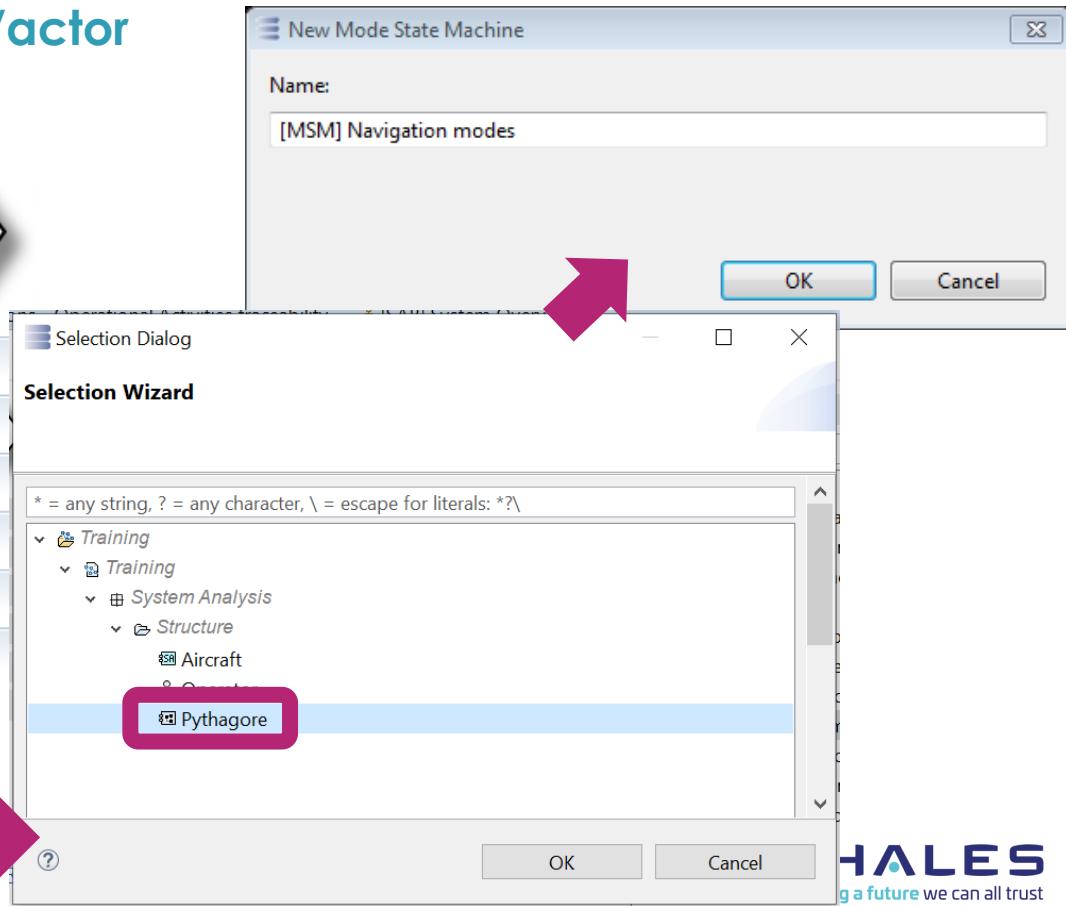
Creating a Mode/State Machine

This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2020 All rights reserved.

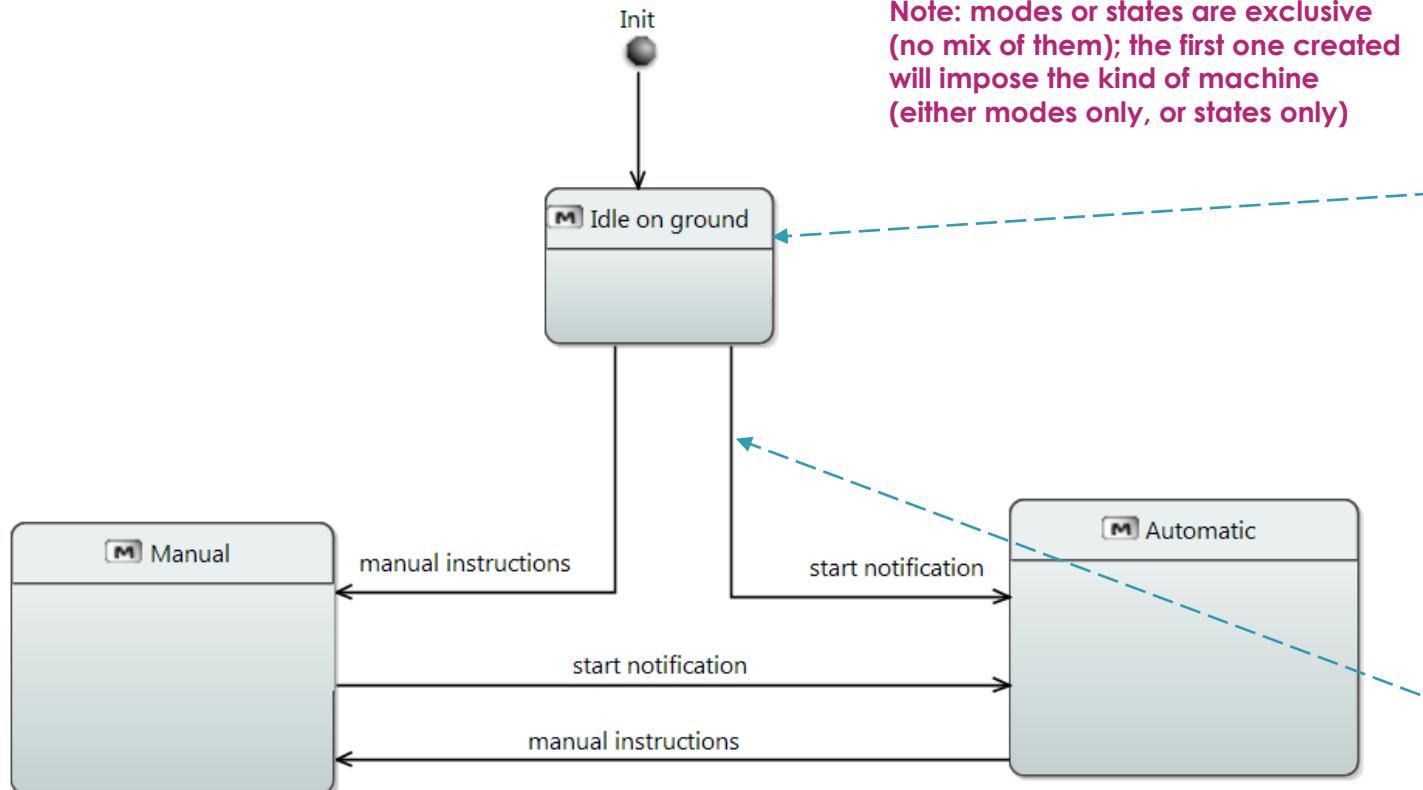
| Select the system/component/actor to which it applies



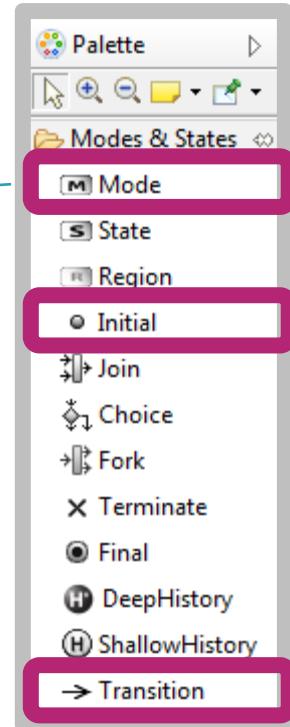
- ▶ Transition From Operational Activities
- ▶ Define Actors, Missions and Capabilities
- ▶ Refine System Functions, describe Functional Exchanges
- ▶ Allocate System Functions to System and Actors
- ▶ Define Interfaces and describe Interface Scenarios
- ▼ Transverse Modeling
 - [CDB] Create a new Class diagram
 - [MSM] Create a new Mode State Machine diagram



Creating a Mode/State Machine

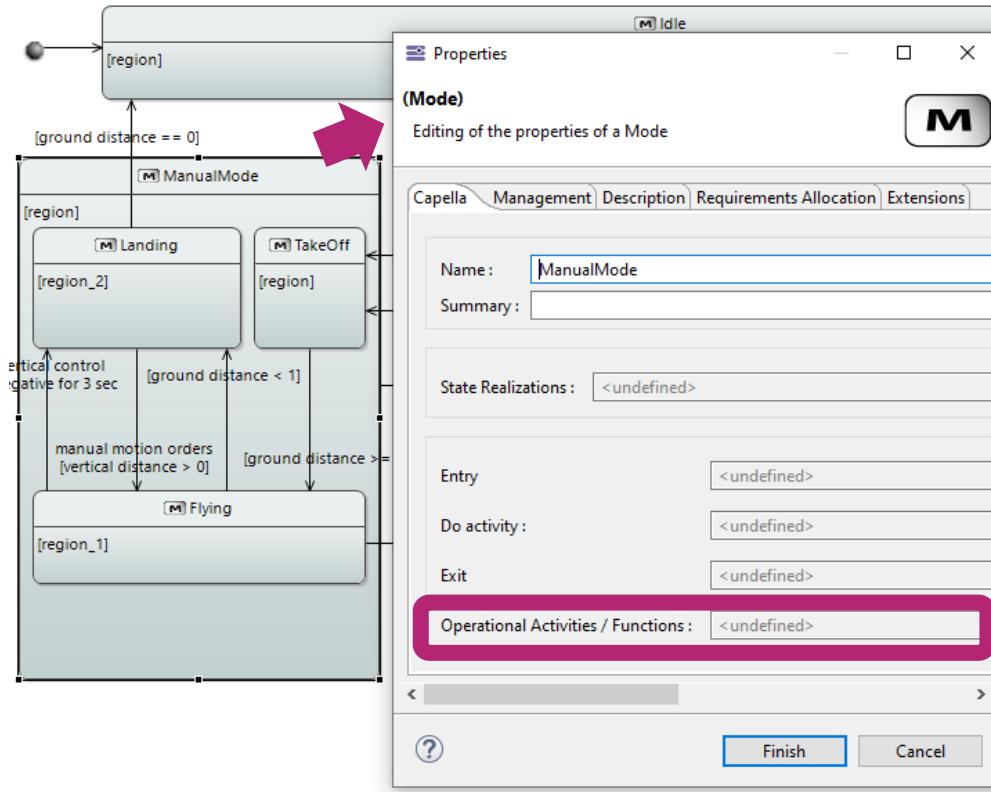


Note: modes or states are exclusive
(no mix of them); the first one created
will impose the kind of machine
(either modes only, or states only)



Creating a Mode/State Machine

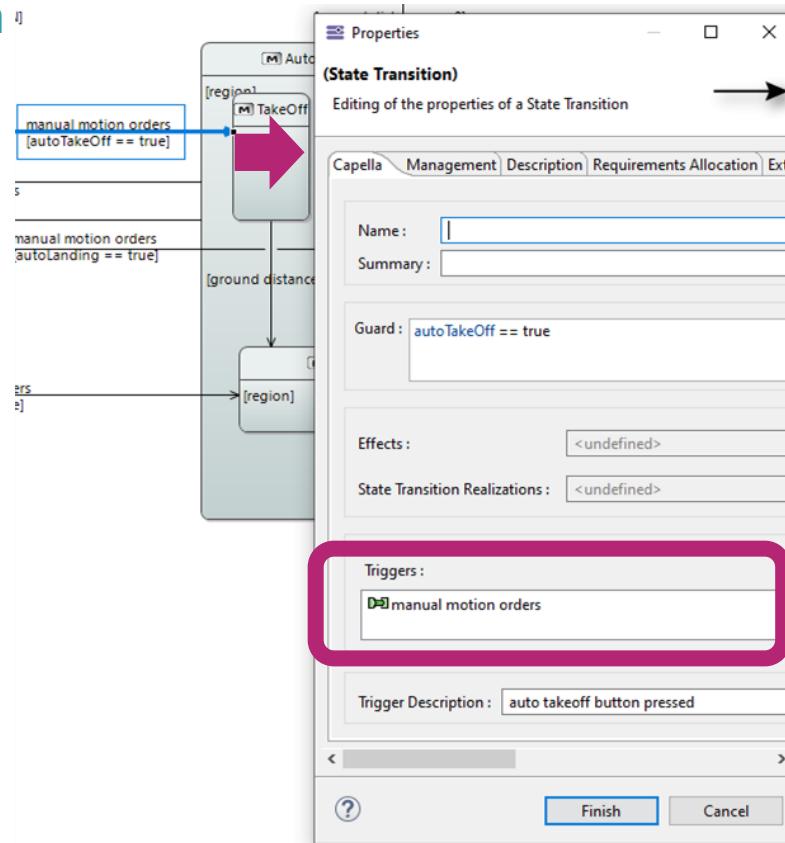
Functions that must be available in a mode or a state are described in the properties of the mode or state



Note: this procedure uses the simplified, basic version of modes & states management, not the more powerful one available as an addon, and providing advanced concepts such as configurations and situations

Creating a Mode/State Machine

Functional exchanges that trigger a transition are described in the properties of the transition

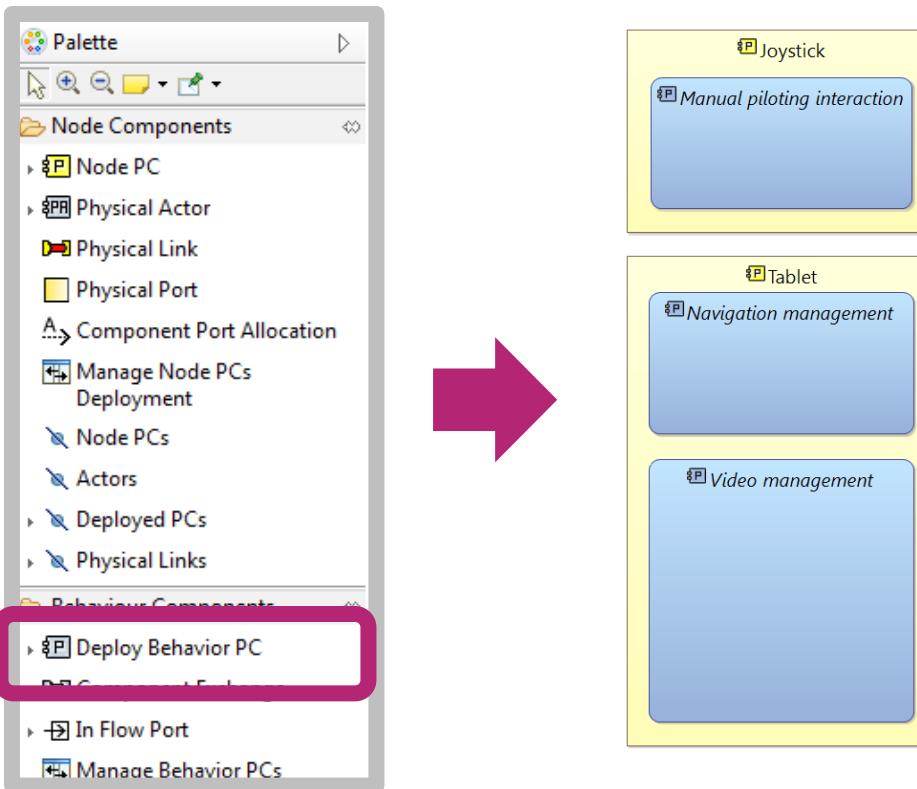


Note: this procedure uses the simplified, basic version of modes & states management, not the more powerful one available as an addon, and providing advanced concepts such as configurations and situations

Implementing Behavioral components on Implementation Component

This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2020 All rights reserved.

| Select 'Deploy Behavior PC', then clic on the Implementation Component to create a new behavioral component and implement it there

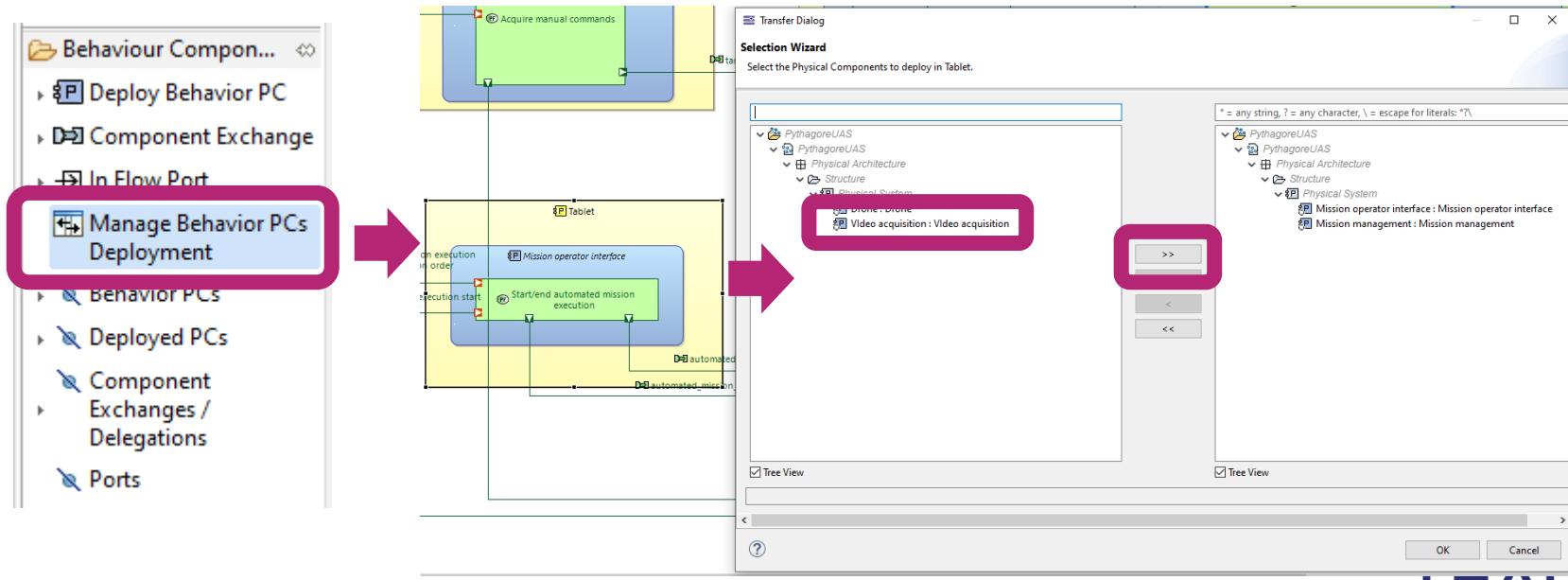


Rounded corners:
deployment relationships

Implementing Behavioral components on Implementation Component

This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2020 All rights reserved.

To implement an existing behavioral component in an implementation component, select 'Manage Behavior PC Deployment', click on the implementation component, then move existing components from left to right list using '>>'



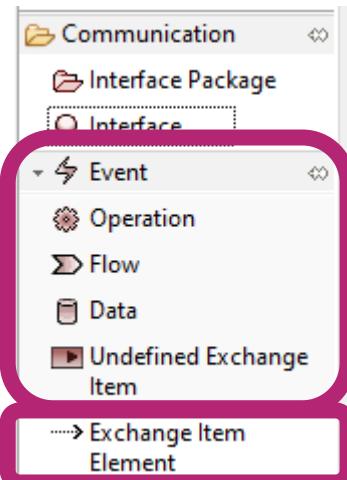
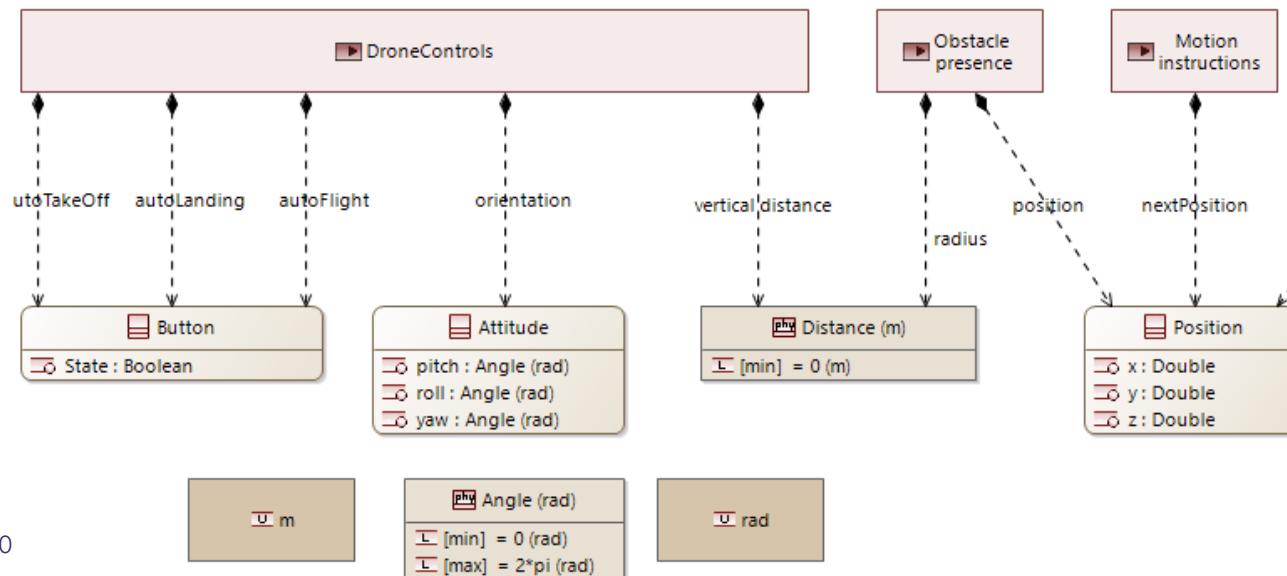
Building Exchange Items and Interfaces

This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2020 All rights reserved.

Exchange Items creation: from the right-side palette

- Click on down arrow to list them

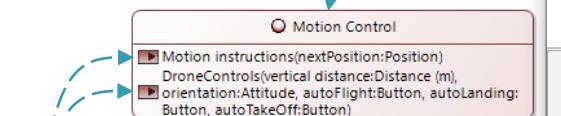
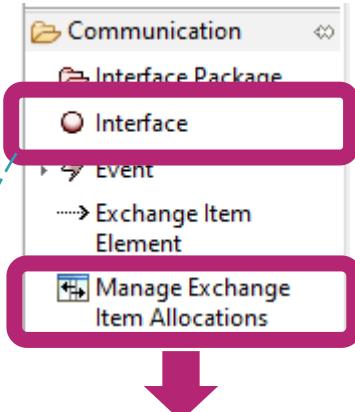
Once an Exchange Item is created, it can be linked to any class using 'Exchange Item Element' tool



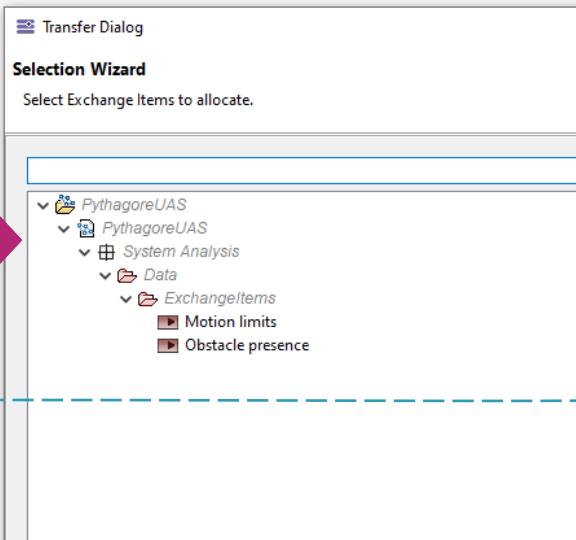
Building Exchange Items and Interfaces

Interface creation: from the right-side palette

Once an Interface is created, any Exchange Item can be added using 'Manage Exchange Item Allocations' tool and clicking in the Interface



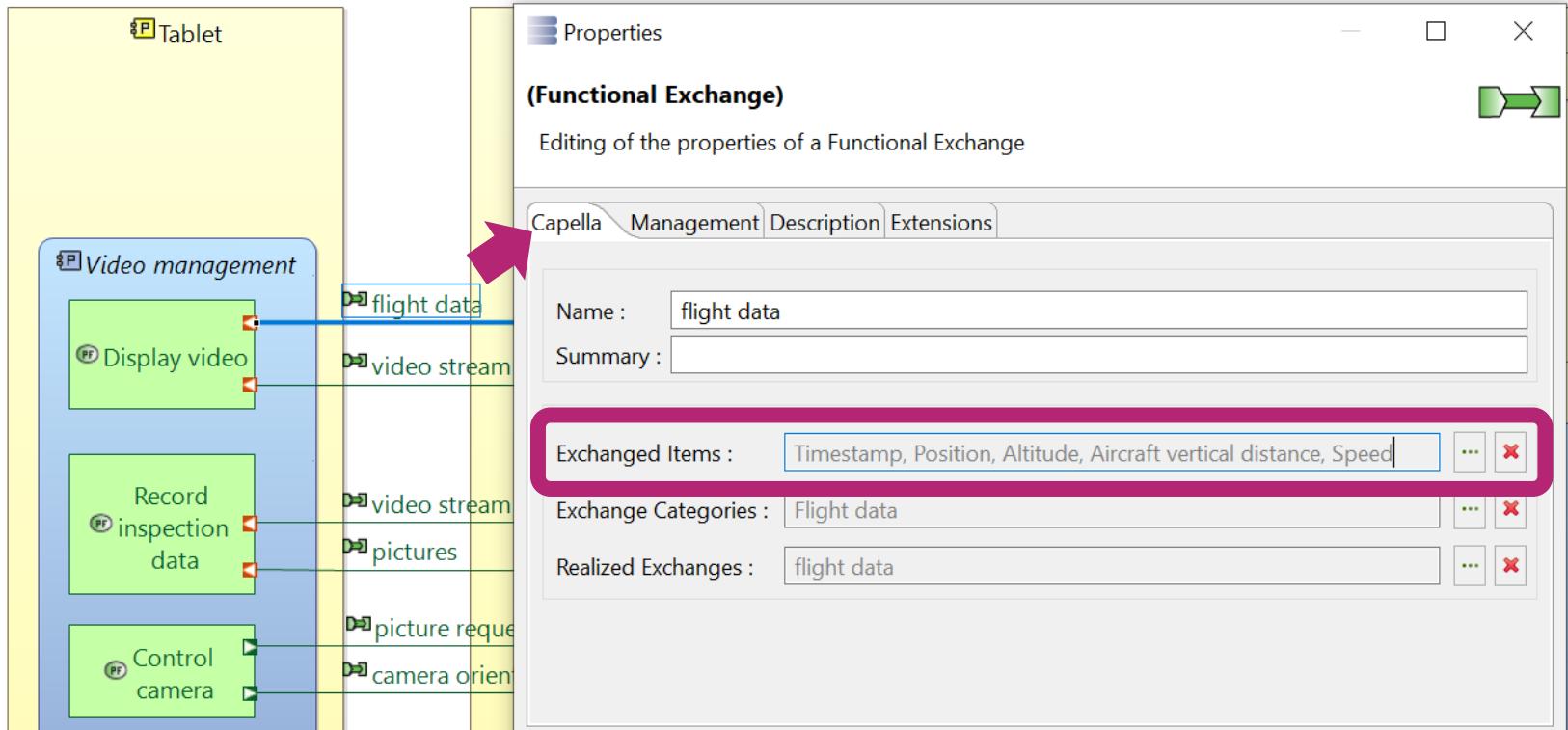
Exchange Items
provided by the
Interface



Allocating Exchange Items to functional Exchanges

This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2020 All rights reserved.

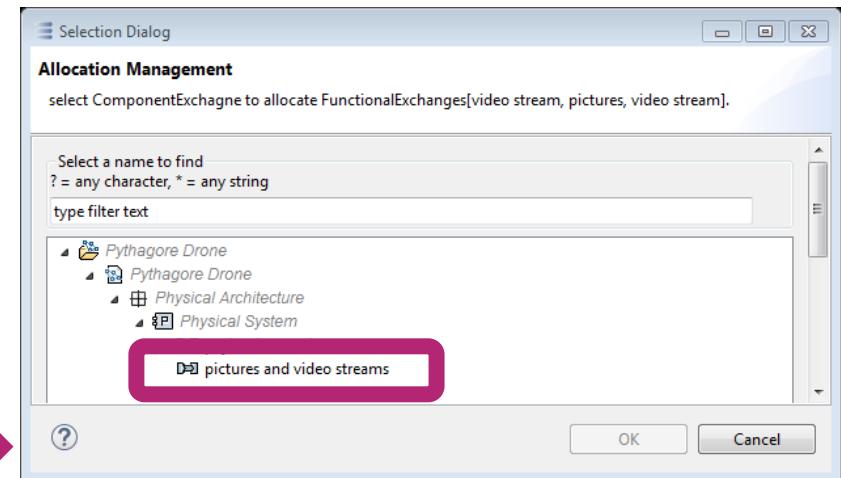
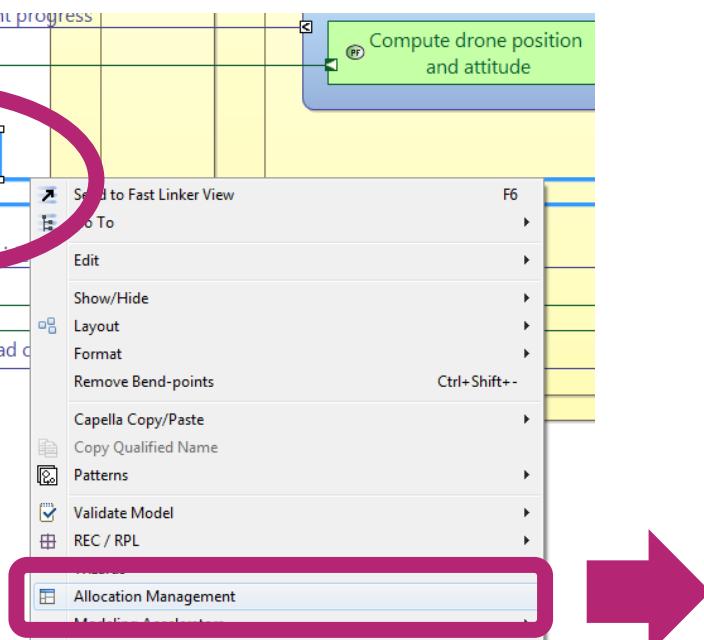
Through the Properties dialog box (double click on the functional Exchange)



Allocating functional Exchanges on a Component Exchange

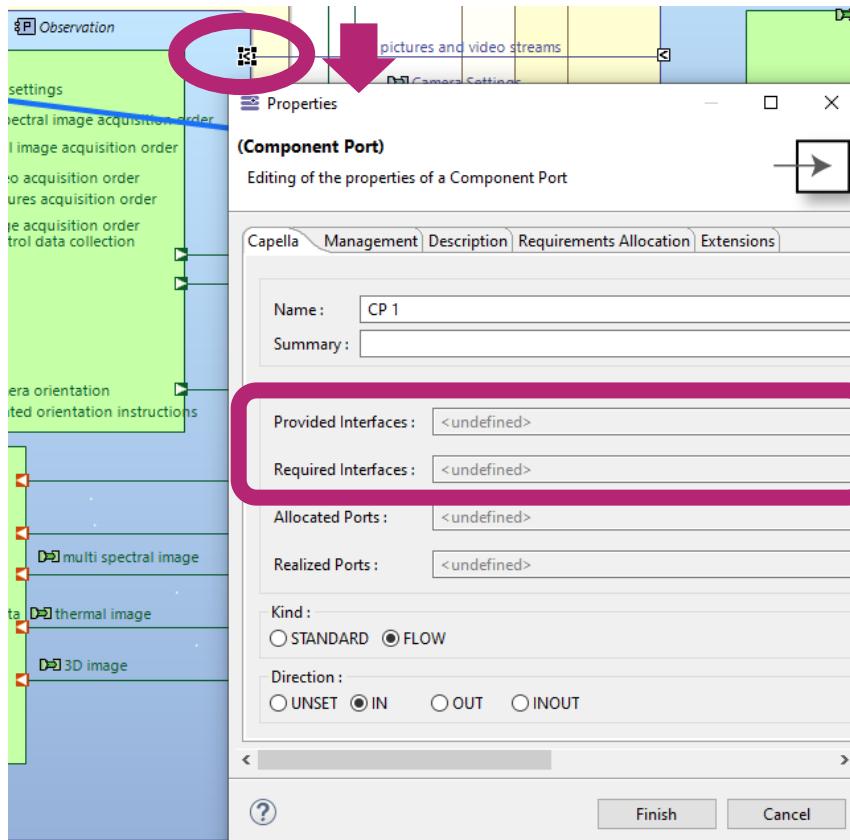
This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2020 All rights reserved.

| Select a set of functional exchanges to be allocated to the same component exchange, then right-click menu, and choose the component exchange



Linking an Interface to a behavioral Component Port

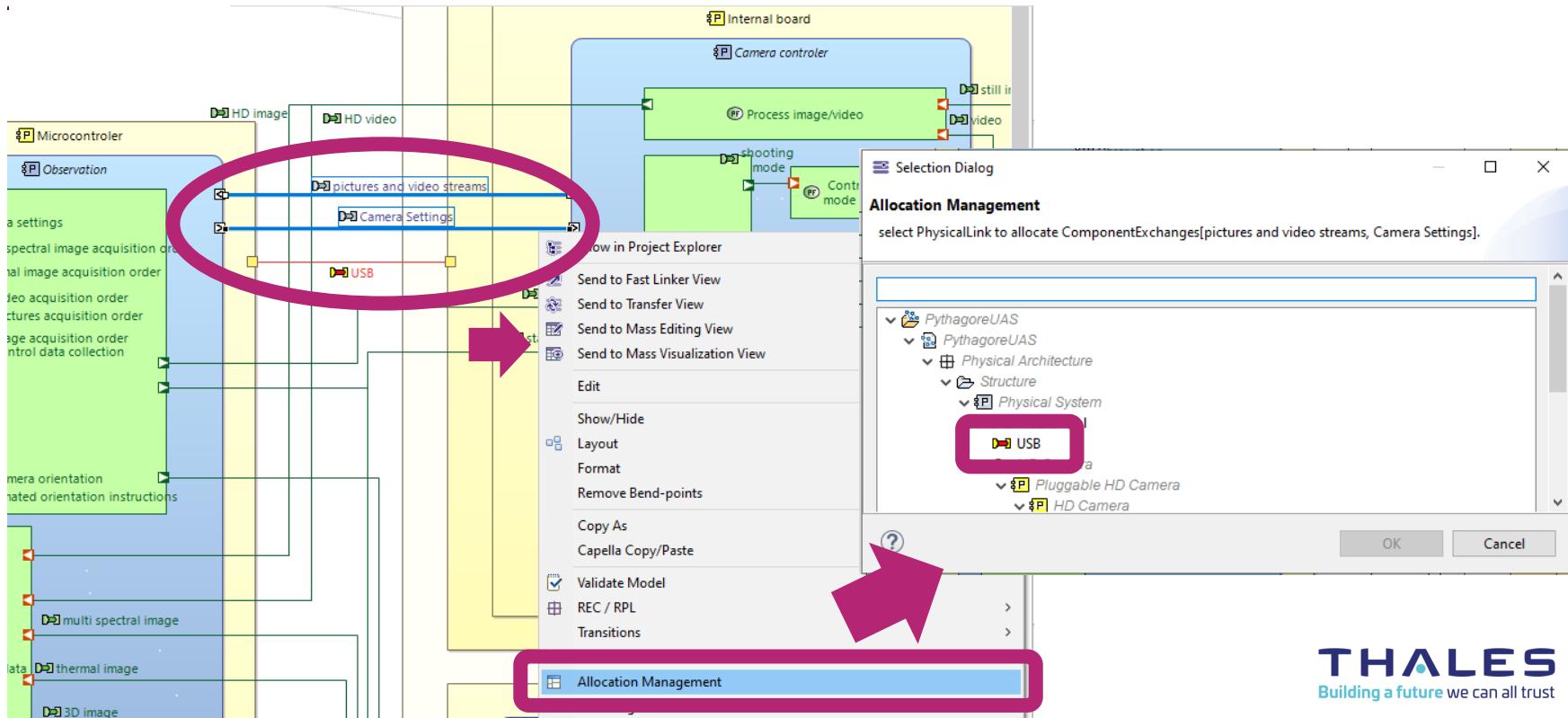
Through the Component Port Properties dialog box (double click on the Component Port)



Allocating behavioral Component Exchanges on a physical Link

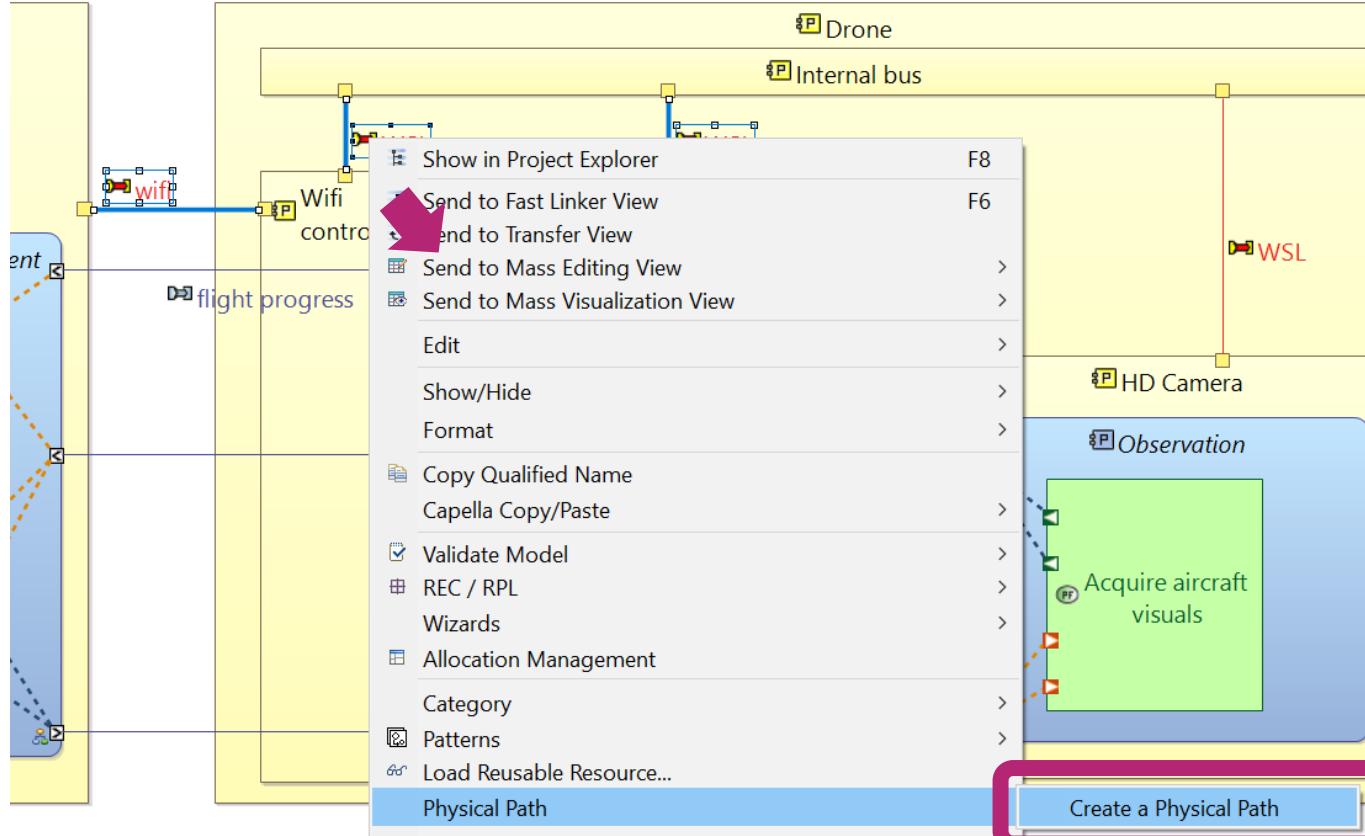
This document may not be reproduced, modified, adapted, published, translated, in any way, in whole or in part or disclosed to a third party without the prior written consent of Thales - © Thales 2020 All rights reserved.

| Select a set of component exchanges to be allocated to the same physical Link, then right-click menu, and choose the physical Link



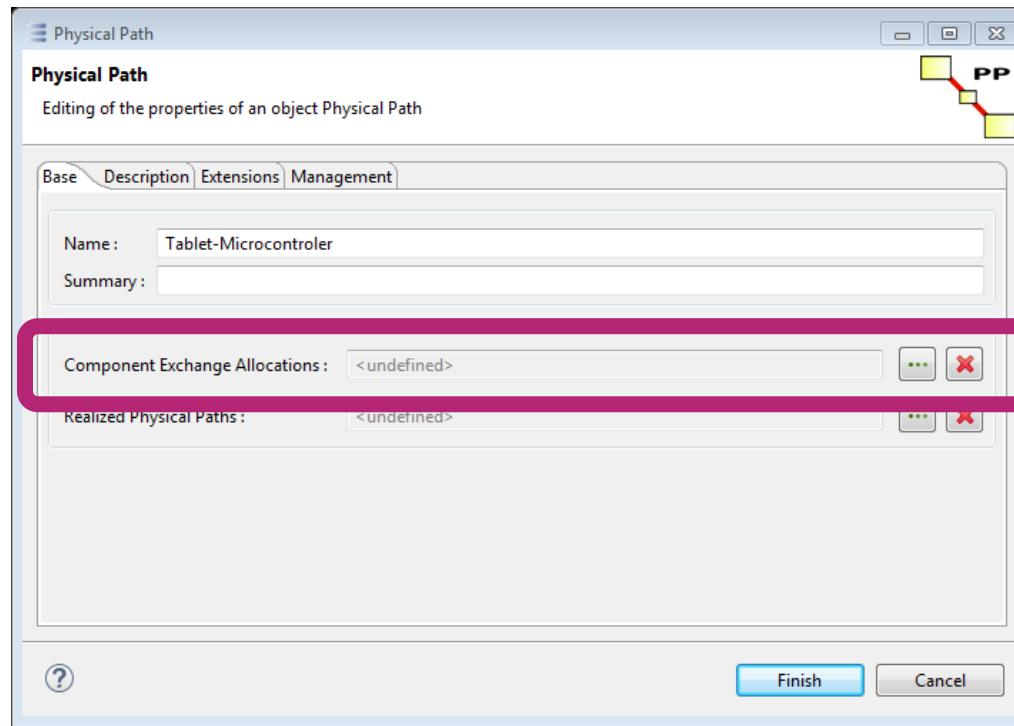
Creating a Physical Path

| Similar to functional chain creation, selecting a set of physical Links



Allocating behavioral Component Exchanges on a physical Path

| Through the physical Path Properties dialog box (double click on the physical Path)



Need more?

Arcadia and Capella are free and Open Source

Capella & Arcadia website:

<https://www.eclipse.org/capella/>

LinkedIn 

<http://www.linkedin.com/company/capella-modelling-workbench>

Capella forum:

<https://polarsys.org/forums/index.php/f/13/>

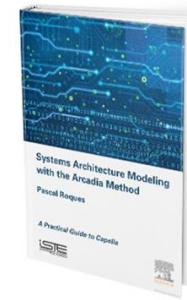
Clarity consortium for the Capella ecosystem

<http://www.clarity-se.org/>



JEAN-LUC VOIRIN

Model-based System and Architecture
Engineering with the Arcadia Method



PASCAL ROQUES

Systems Architecture Modeling with the
Arcadia Method - A Practical Guide to
Capella Modeling Tool

