ARCADIA LANGUAGE REFERENCE:

META MODEL

Arcadia Language Concepts in details

Jean-Luc Voirin

Table of Contents

1 Scope of this document

ARCADIA is a tooled method devoted to systems & architecture engineering, supported by Capella modelling tool.

It describes the detailed reasoning to

- *understand the real customer need,*
- *define and share the product architecture among all engineering stakeholders,*
- *early validate its design and justify it,*
- *ease and master Integration, Validation, Verification, Qualification (IVVQ).*

It can be applied to complex systems, equipment, software or hardware architecture definition, especially those dealing with strong constraints to be reconciled (cost, performance, safety, security, reuse, consumption, weight…).

It is intended to be used by most stakeholders in system/product/software or hardware definition and IVVQ as their common engineering reference and collaboration support.

ARCADIA stands for ARChitecture Analysis and Design Integrated Approach.

This document provides a formalised view of main concepts of engineering data elaborated and exploited by Arcadia activities, along with their relations.

Source: Voirin J.-L., Model-based System and Architecture Engineering with the Arcadia Method, ISTE Press, London & Elsevier, Oxford, 2017. ©Thales-ISTE

Note that this metamodel focuses on core Arcadia perspectives description only. A wider but less formal view of Arcadia main engineering data and assets is provided in other documents.

2 Arcadia Reference Documents

An in-depth introduction and description of Arcadia, with explanations on the method, on the language, illustrated by detailed examples of application, can be found in the Arcadia reference book:

> *Jean-Luc Voirin, 'Model-based System and Architecture Engineering with the Arcadia Method', ISTE Press, London & Elsevier, Oxford, 2017*

A presentation of Arcadia main principles and concepts can be found in the following online documents, including this one:

- Arcadia Engineering Landscape: an introduction to Engineering as supported by Arcadia
- Arcadia User Guide: a first level description of Arcadia approach and main engineering Tasks
- Arcadia Reference - Activities: an in-depth description of Arcadia tasks and activities
- Arcadia Reference - Data Model: data created and exploited by these activities
- Arcadia Reference - Capabilities: main processes supporting engineering
- Arcadia Language - MetaModel: a more formal description of Arcadia language concepts
- Arcadia Q&A: real life questions and answers on deploying Arcadia

See table 'Summary of reference Documents Contents' next page.

For easier navigation capabilities (including in diagrams, between activities and data, etc.), a web version can be browsed here.

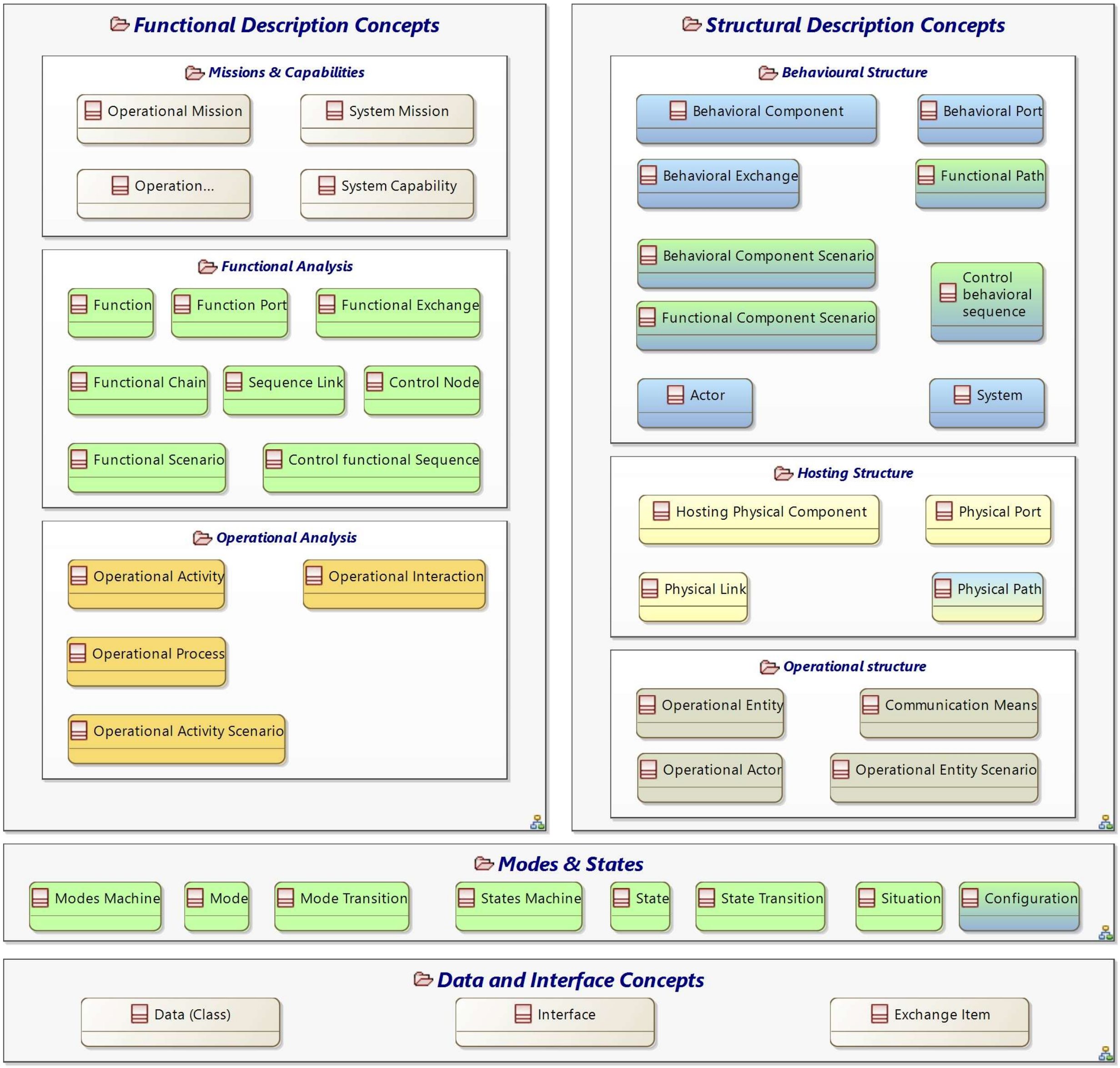
Advanced practitioners in modelling and Arcadia can also access the Arcadia-compliant Capella model of Arcadia, from which this material is automatically extracted, here.

Summary of reference Documents Contents		Book	Landscape	User Guide	Reference - Activities	Reference - DataModel	Reference - Capabilities	Language - MetaModel	Q&A
History	Why was the method created and tooled? For which purpose? With which benefits?	✓							
Philosophy	What are its objectives and expected scope? What are its specificities?	✓	(✓)						
	How does it address Engineering Issues and Challenges?	✓		(✓)			(✓)		
	What kind of major levers does it use to address them?	✓		(✓)			(✓)		
Principles and approach	What are the drivers of each core perspective ...? How to build each of them?	✓		(✓)	(✓)				
	How to address Major engineering Issues using Arcadia and these perspectives?	✓					✓		
Details for implementation	What are the detailed processes to build each of the core perspectives?	(✓)			✓				
	How and where are engineering data elaborated and used to address major engineering challenges?	✓			(✓)	(✓)	✓		
	What is the formal definition of the Arcadia language & concepts?	✓				(✓)		✓	
	Examples and samples of models?	✓							
Hints for Deployment	Which major questions arise in projects applying Arcadia?								✓

✓ : fully detailled

(✓) : simplified or partial

3 Arcadia Meta Model Contents at a glance



Each asset and its detailed data are described below.

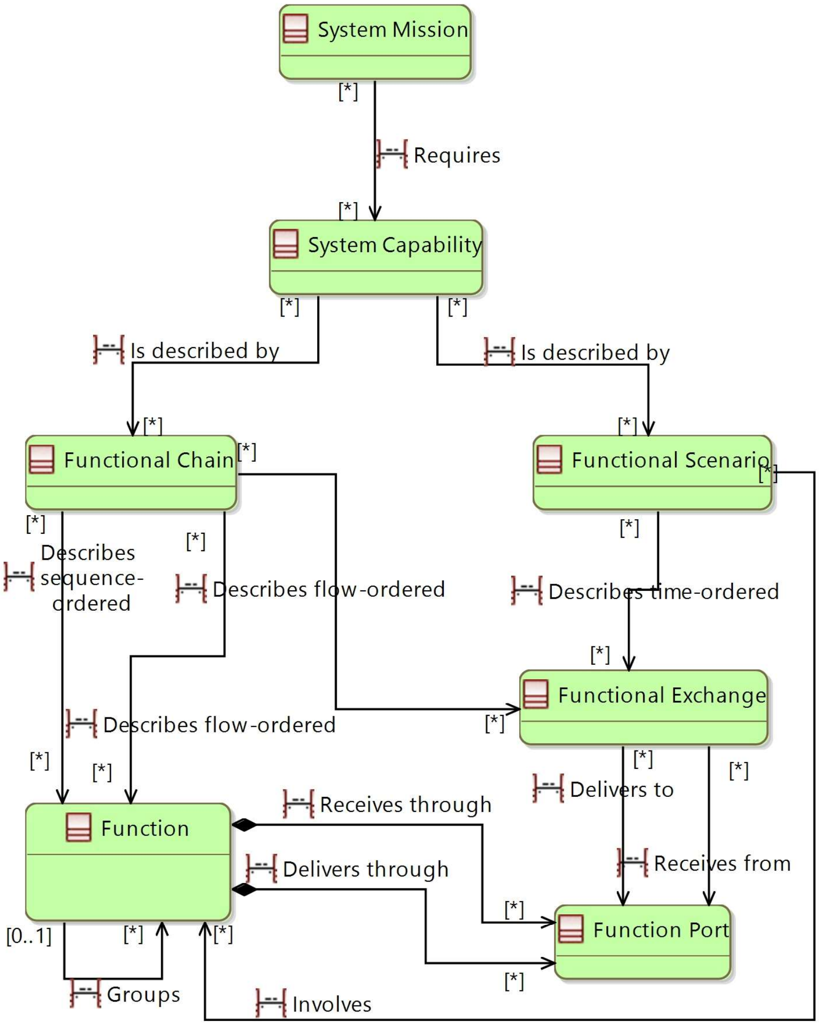
4 Concepts, Relations and Definitions

Note: the representations below are only semi-formal, favouring understandability first. For this reason, they are simplified, and their implementation in Capella modelling tool is more complex.

4.1 Functional Description Concepts

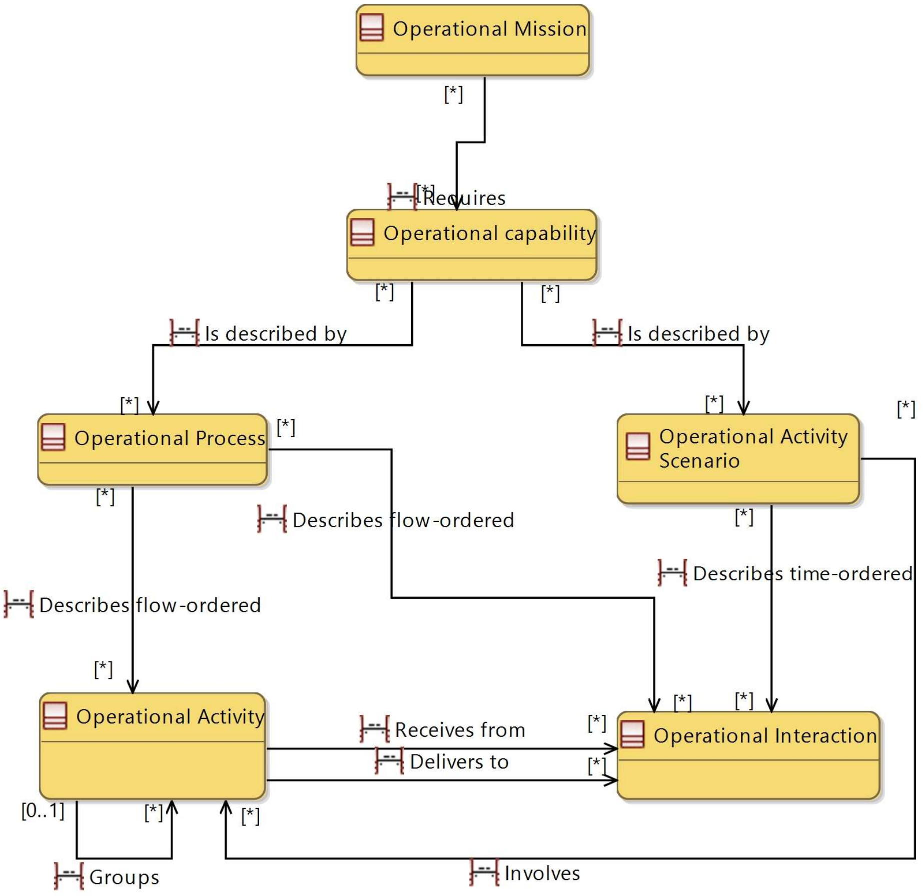
These concepts allow describing the expected behaviour of the solution as required and as designed, and also the behaviour of main users, operators, and stakeholders in operations.

Functional Description Concepts & MetaModel



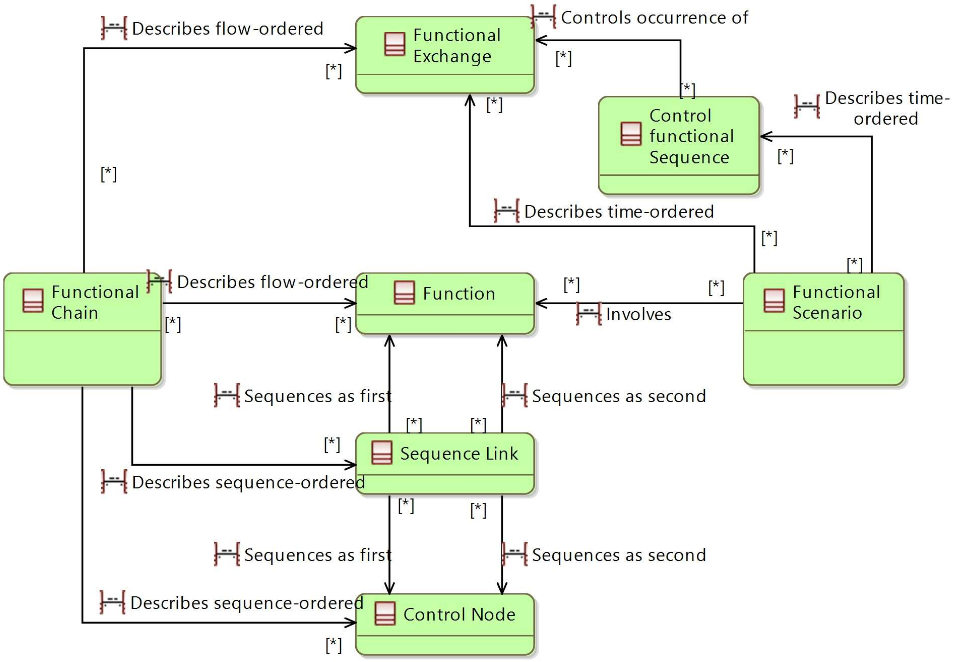
Source: Voirin J.-L., Model-based System and Architecture Engineering with the Arcadia Method, ISTE Press, London & Elsevier, Oxford, 2017. ©Thales-ISTE

Operational Description Concepts & MetaModel



Source: Voirin J.-L., Model-based System and Architecture Engineering with the Arcadia Method, ISTE Press, London & Elsevier, Oxford, 2017. ©Thales-ISTE

Functional Chain & Scenario detailled Concepts & MetaModel



Source: Voirin J.-L., Model-based System and Architecture Engineering with the Arcadia Method, ISTE Press, London & Elsevier, Oxford, 2017. ©Thales-ISTE

4.1.1 Functional Analysis

These concepts allow describing the expected behaviour of the solution as required and as designed.

4.1.1.1 Function

Function

An action, an operation, or a service, performed by the system or one of its components, or also by an actor interacting with the system

©Thales-ISTE

4.1.1.2 Functional Exchange

Functional Exchange

A possible interaction between a source function and a destination function, likely to transmit exchange items through their output and input ports, respectively

©Thales-ISTE

4.1.1.3 Function Port

Functional Port

A place where the function interacts with other functions of its environment. It can be either an input port or an output port, exclusively

©Thales-ISTE

4.1.1.4 Functional Scenario

Functional Scenario

A time-ordered dynamic flow, on a temporal axis (conventionally vertical from top to bottom), of exchanges between different functions in the context of implementing a capability

©Thales-ISTE

4.1.1.5 Functional Chain

Functional Chain

An ordered set of references to functions and the functional exchanges that link them, describing one possible path among all the paths forming the dataflow, in the context of implementing a capability

©Thales-ISTE

4.1.1.6 Control functional Sequence

Control functional Sequence

Control sequences can be defined as time-bounded zones (therefore vertical) to express the parallelism or alternative between several sequences of interactions, or also the iteration or condition of a sequence of interactions.

©Thales-ISTE

4.1.1.7 Control Node

Control Node

Control nodes can be defined between the sequence links, to express the parallelism or alternative between several sequences of functions, or, also the iteration or condition of a sequence to be realized

©Thales-ISTE

4.1.1.8 Sequence Link

Sequence Link

a sequence link between two functions (between function references, in fact) indicates that the source function should operate before the destination function, at least in the context of this functional chain.

©Thales-ISTE

4.1.2 Operational Analysis

These concepts allow describing the expected behaviour of main users, operators, and stakeholders in operations.

4.1.2.1 Operational Interaction

Operational Interaction

A possible dependency between two operational activities – the interaction source and destination, in the form of transmitting elements conveyed by the interaction

©Thales-ISTE

4.1.2.2 Operational Activity

Operational Activity

An action, an operation or a service, realized by an operational entity likely to influence the system definition or usage

©Thales-ISTE

4.1.2.3 Operational Activity Scenario

Operational (Activity) Scenario

A time-ordered dynamic flow, on a temporal axis (conventionally vertical from top to bottom), of interactions between different operational activities in the context of implementing a capability

4.1.2.4 Operational Process

Operational Process

An ordered set of references to operational activities and the interactions that link them, describing one possible path among all the paths forming the operational analysis dataflow

4.1.3 Missions & Capabilities

These concepts allow describing the main goals of main users, operators, and stakeholders in operations, and those allocated to the solution itself.

4.1.3.1 System Mission

System Mission

A high-level goal to which the system should contribute

4.1.3.2 Operational Capability

Operational Capability

An ability, expected of one or more operational entities, to provide a service contributing to fulfilling one or more operational missions

4.1.3.3 System Capability

System Capability

The system’s expected ability to supply a service contributing to fulfilling one or more missions

4.1.3.4 Operational Mission

Operational Mission

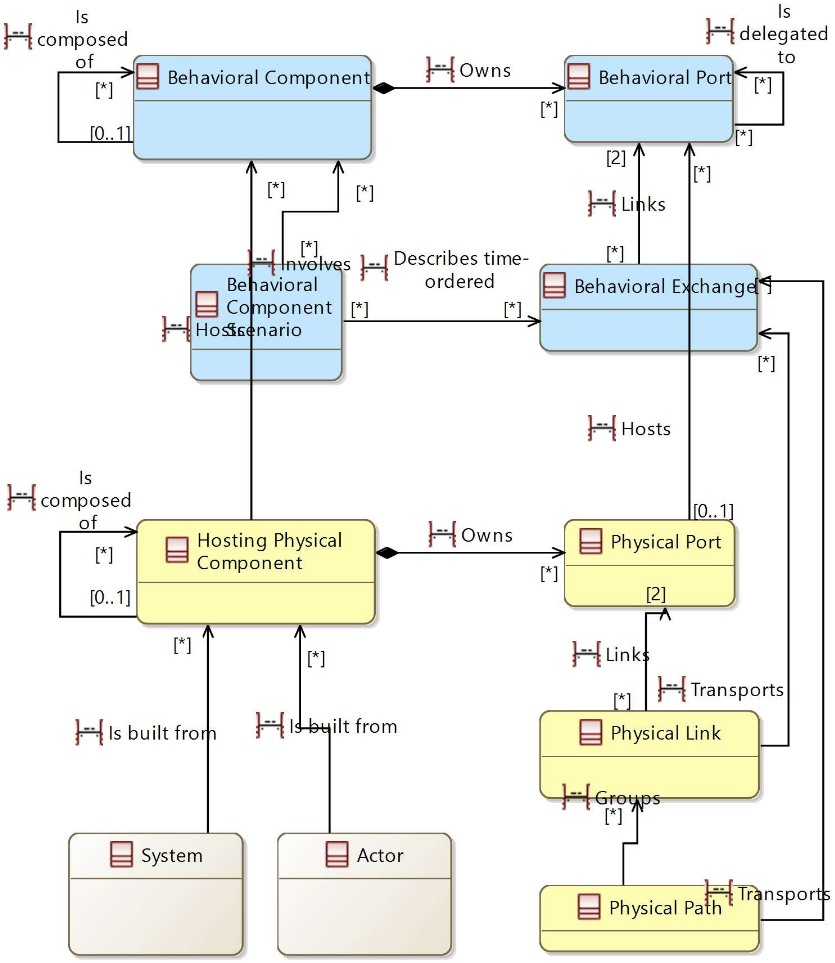
A high-level goal to which one or more operational entities should contribute, and which is likely to influence system definition or usage

©Thales-ISTE

4.2 Structural Description Concepts

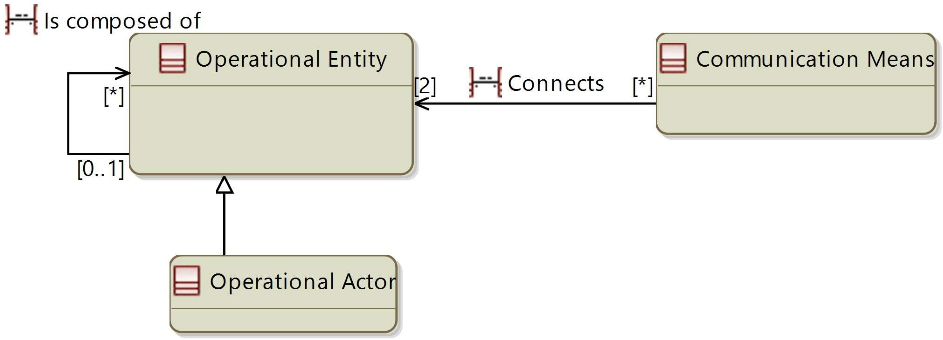
These concepts allow describing the structure of the solution as designed, and also the organisation of main users, operators, and stakeholders in operations.

Structural Description Concepts & MetaModel



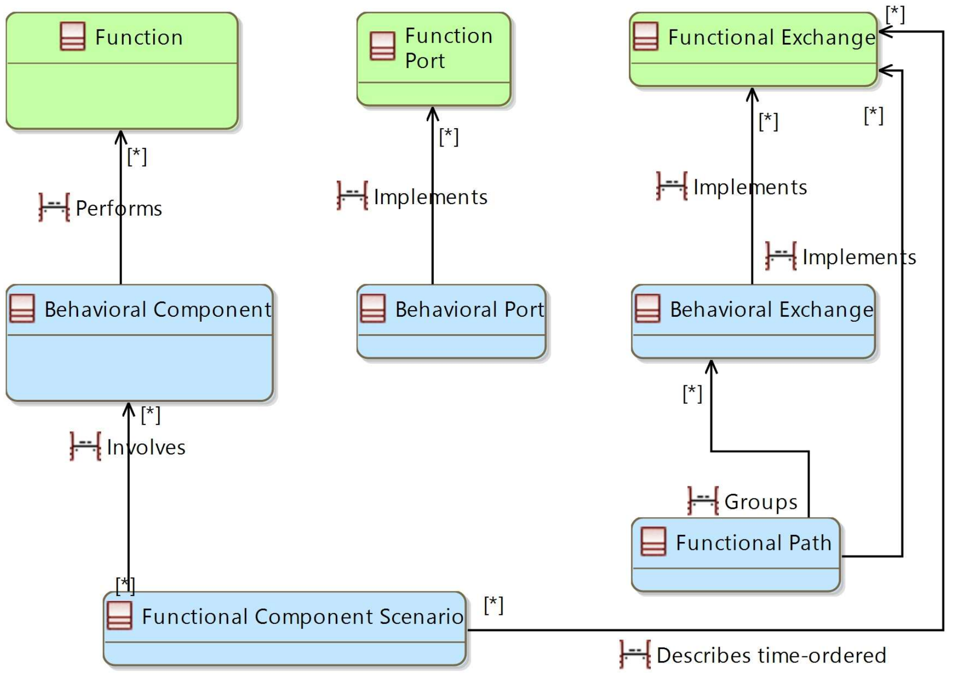
Source: Voirin J.-L., Model-based System and Architecture Engineering with the Arcadia Method, ISTE Press, London & Elsevier, Oxford, 2017. ©Thales-ISTE

Structural operational Description Concepts & MetaModel



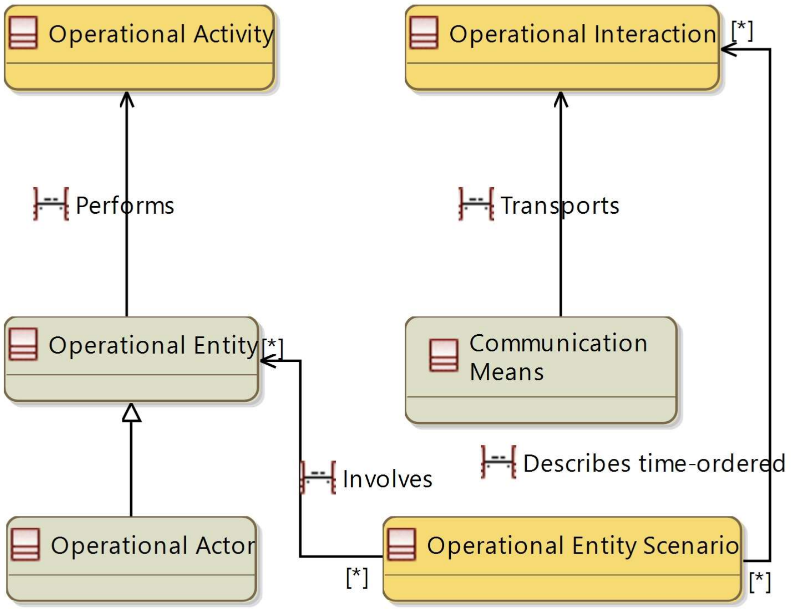
Source: Voirin J.-L., Model-based System and Architecture Engineering with the Arcadia Method, ISTE Press, London & Elsevier, Oxford, 2017. ©Thales-ISTE

Functional Vs Structural Description Concepts & MetaModel



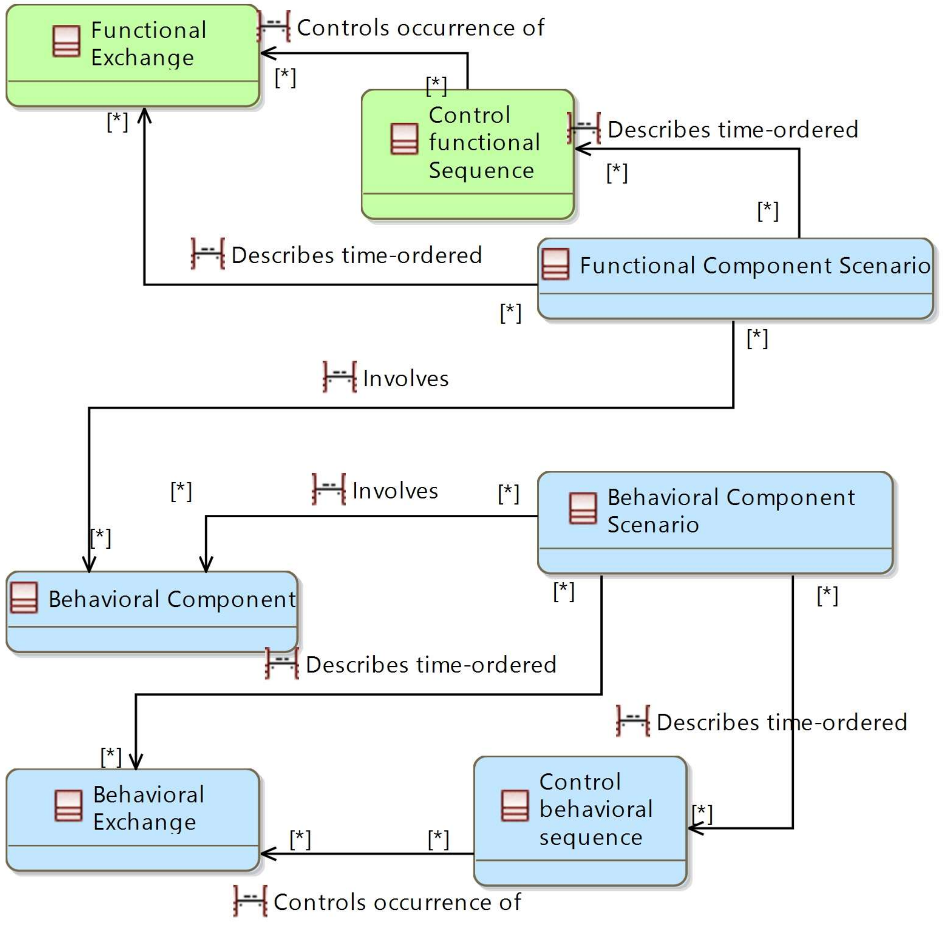
Source: Voirin J.-L., Model-based System and Architecture Engineering with the Arcadia Method, ISTE Press, London & Elsevier, Oxford, 2017. ©Thales-ISTE

Functional Vs Structural operational Description Concepts & MetaModel



Source: Voirin J.-L., Model-based System and Architecture Engineering with the Arcadia Method, ISTE Press, London & Elsevier, Oxford, 2017. ©Thales-ISTE

Component Scenario detailled Concepts & MetaModel



Source: Voirin J.-L., Model-based System and Architecture Engineering with the Arcadia Method, ISTE Press, London & Elsevier, Oxford, 2017. ©Thales-ISTE

4.2.1 Behavioural Structure

These concepts allow describing the behavioural structure of the solution as designed, in terms of components that realise functional behaviour.

4.2.1.1 System

System

An ordered set of elements functioning as a whole, responding to customer and user demand and need, and subject of engineering

©Thales-ISTE

4.2.1.2 Behavioral Port

Behavioural Port

A place of interaction for the component to which it is attached to other components or actors in its environment. It can be of three types: input port, output port and bidirectional port

©Thales-ISTE

4.2.1.3 Behavioral Exchange

Behavioural exchange

A possible interaction between a source behavioural component and a destination behavioural component, likely to transmit exchange items via their ports

©Thales-ISTE

4.2.1.4 Actor

System Actor

An entity that is external to the system (human or not), interacting with it, especially via its interfaces

©Thales-ISTE

4.2.1.5 Behavioral Component

Component

A constituent part of the system, contributing to its behaviour and/or properties, along with other components and actors external to the system

Behavioural Component

A system component, responsible for carrying out some of the functions devolved to the system, by interacting with its other behavioural components and external actors

Logical Component

A system component described at a conceptual level (in principle, abstract) in logical architecture

©Thales-ISTE

4.2.1.6 Behavioral Component Scenario

Behavioural Component Scenario

A time-ordered dynamic flow, on a temporal axis (conventionally vertical from top to bottom), of *behavioural* exchanges between different behavioural components in the context of implementing a capability

©Thales-ISTE

4.2.1.7 Functional Path

Functional Path

An ordered set of references to behavioural exchanges and delegation links between behavioural ports, defining a continuous path likely to implement one or more functional exchanges between two functions allocated to the path source and destination components

©Thales-ISTE

4.2.1.8 Functional Component Scenario

Functional Component Scenario

A time-ordered dynamic flow, on a temporal axis (conventionally vertical from top to bottom) of *functional* exchanges between different behavioural components, in the context of implementing a capability

©Thales-ISTE

4.2.1.9 Control behavioral sequence

Control behavioural Sequence

Control sequences can be defined as time-bounded zones (therefore vertical) to express the parallelism or alternative between several sequences of interactions, or also the iteration or condition of a sequence of interactions.

©Thales-ISTE

4.2.2 Hosting Structure

These concepts allow describing the hosting structure of the solution as designed, in terms of components providing resources to (or hosting, or implementing) behavioural components.

4.2.2.1 Physical Path

Physical Path

An ordered set of references to physical links, defining a continuous path likely to route one or more behavioural exchanges between components not linked by a single physical link

©Thales-ISTE

4.2.2.2 Physical Port

Physical Port

A hosting physical component’s point of connection with its environment. A physical port is not oriented

©Thales-ISTE

4.2.2.3 Physical Link

Physical Link

A means of communication, transport or routing between two hosting physical components, used as a support for behavioural exchanges

©Thales-ISTE

4.2.2.4 Hosting Physical Component

Component

A constituent part of the system, contributing to its behaviour and/or properties, along with other components and actors external to the system

Hosting physical Component

A component hosting a number of behavioural components, providing them with the resources they require to function and to interact with their environment

©Thales-ISTE

4.2.3 Operational structure

These concepts allow describing the structure of the organisation of main users, operators, and stakeholders in operations.

4.2.3.1 Operational Actor

Operational Actor

A [usually human] non decomposable operational Entity

©Thales-ISTE

4.2.3.2 Operational Entity

Operational Entity

A real-world entity (a physical element, a group or organization, another system), carrying out operational activities to which the system is likely to contribute, or which can influence the system

©Thales-ISTE

4.2.3.3 Communication Means

Communication Means

A support linking two operational entities, and is followed by interactions between these two entities

©Thales-ISTE

4.2.3.4 Operational Entity Scenario

Operational Entity Scenario

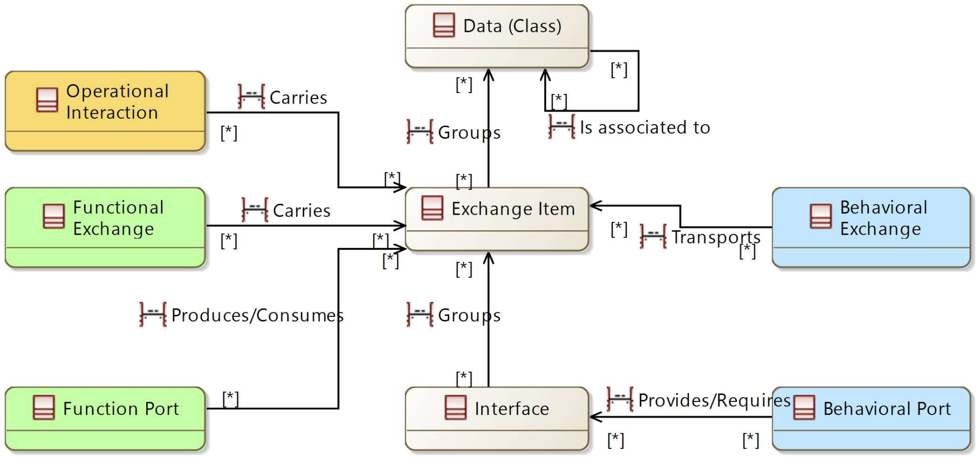
An operational Entity scenario is a time-ordered dynamic flow, on a temporal axis (conventionally vertical from top to bottom), of interactions between different operational entities or actors, in the context of an operational capability.

©Thales-ISTE

4.3 Data and Interface Concepts

These concepts allow describing the contents of exchanges and interactions involved in the behaviour of system, components, actors and stakeholders of the solution in operation.

Data and Data Use Description Concepts & MetaModel



Source: Voirin J.-L., Model-based System and Architecture Engineering with the Arcadia Method, ISTE Press, London & Elsevier, Oxford, 2017. ©Thales-ISTE

4.3.1.1 Data (Class)

Data (or Class)

An element produced or used by functions or components, and routed by one or more exchanges between them

©Thales-ISTE

4.3.1.2 Interface

Interface

A set of semantically coherent exchange items, allowing two components (and the system and actors), to communicate, according to a communication "contract" shared between them

©Thales-ISTE

4.3.1.3 Exchange Item

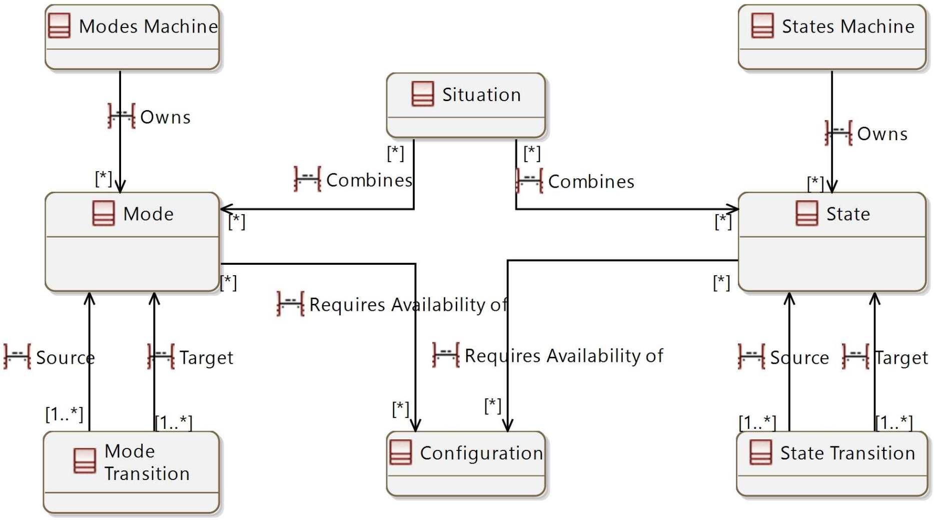
Exchange Item

An ordered set of references to elements routed together, during an interaction or exchange between functions, components and actors

4.4 Modes & States

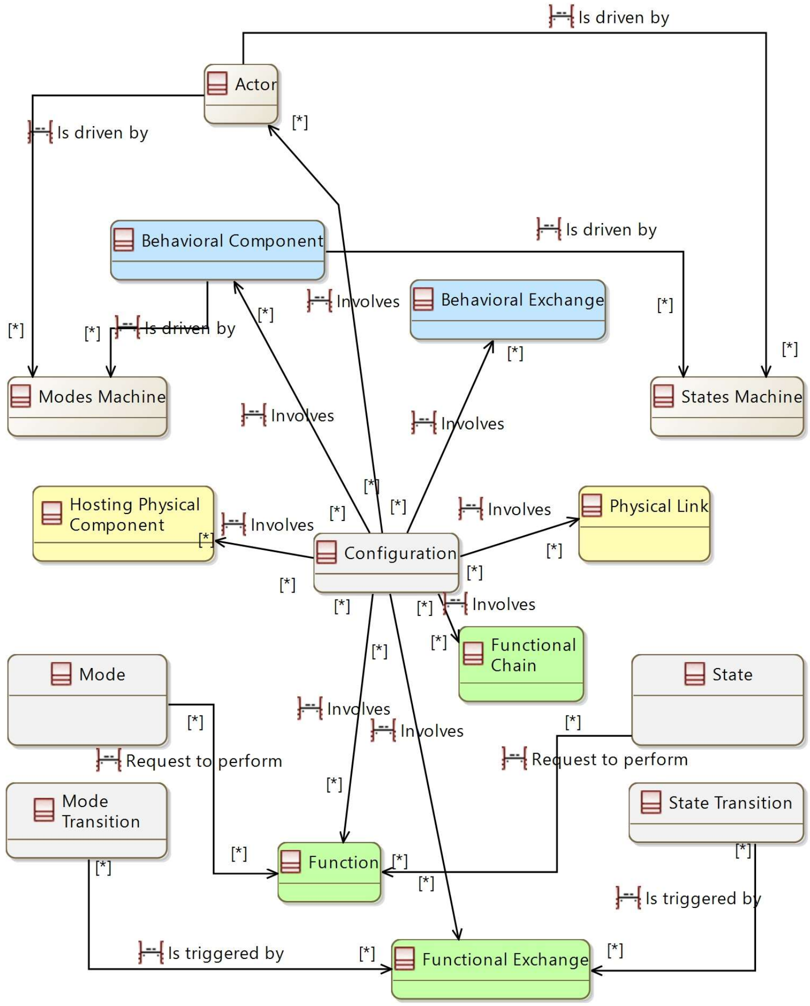
These concepts allow complementing the former behaviour of the solution as required and as designed, and also the behaviour of main users, operators, and stakeholders in operations, in terms of contexts or conditions evolving along time.

Modes & States Description Concepts & MetaModel



Source: Voirin J.-L., Model-based System and Architecture Engineering with the Arcadia Method, ISTE Press, London & Elsevier, Oxford, 2017. ©Thales-ISTE

Modes & States Vs Functional and Structural Descriptions Concepts & MetaModel



Source: Voirin J.-L., Model-based System and Architecture Engineering with the Arcadia Method, ISTE Press, London & Elsevier, Oxford, 2017. ©Thales-ISTE

4.4.1.1 States Machine

Mode or State Machine

a mode(s) or state(s) machine describes the occurrence of a single model element, which may be the system, a component, an actor or an operational entity. Its transitions are commanded by elements of the functional dataflow

©Thales-ISTE

4.4.1.2 State Transition

Transition

A change from one mode to another mode or from one state to another state

©Thales-ISTE

4.4.1.3 Modes Machine

Mode or State Machine

a mode(s) or state(s) machine describes the occurrence of a single model element, which may be the system, a component, an actor or an operational entity. Its transitions are commanded by elements of the functional dataflow

©Thales-ISTE

4.4.1.4 Mode Transition

Transition

A change from one mode to another mode or from one state to another state

©Thales-ISTE

4.4.1.5 Situation

Situation

A combination of states and modes linked by Boolean operators (of the type AND, OR, NOT), and representing the conditions of superposition of these states and modes simultaneously at a given instant

©Thales-ISTE

4.4.1.6 State

State

A behaviour undergone by the system, a component, an actor or an operational entity, in some conditions imposed by the environment

©Thales-ISTE

4.4.1.7 Mode

Mode

A behaviour expected of the system, a component or also an actor or operational entity, in some chosen conditions

©Thales-ISTE

4.4.1.8 Configuration

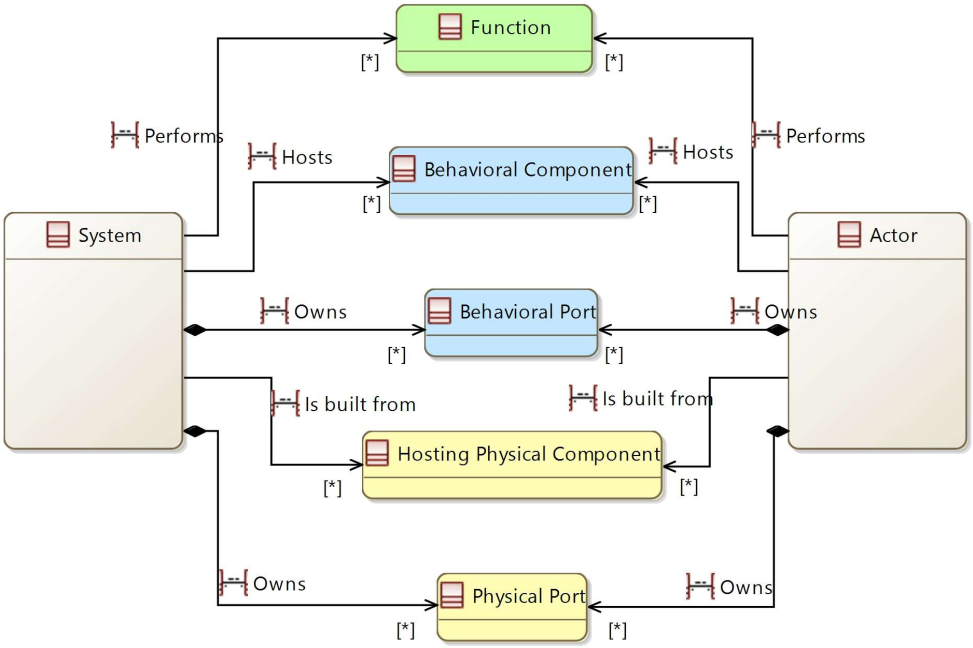
Configuration

A set of model items that are globally available or unavailable in a given context. A context can here be an active mode or state

©Thales-ISTE

4.5 Simplification rules for representation

For sake of simplicity, some simplification rules are allowed in representation, as depicted in the figure below.



Source: Voirin J.-L., Model-based System and Architecture Engineering with the Arcadia Method, ISTE Press, London & Elsevier, Oxford, 2017. ©Thales-ISTE

5