



ARCADIA REFERENCE: ENGINEERING DATA MODEL

Arcadia Engineering Data & Relations

Jean-Luc Voirin

©Thales 2023

Table of Contents

1	Scope of this document	4
2	Arcadia Reference Documents.....	5
3	Arcadia Data Model Contents at a glance	8
4	Arcadia Engineering Data Model.....	9
4.1	NEED Description	9
4.1.1	OPERATIONAL NEED	9
4.1.2	SYSTEM NEED	11
4.1.3	NON FORMALISED NEED.....	13
4.1.4	[non functional] Constraint.....	13
4.2	SOLUTION Description.....	13
4.2.1	LOGICAL ARCHITECTURE.....	15
4.2.2	PHYSICAL ARCHITECTURE	17
4.2.3	PRODUCT BREAKDOWN STRUCTURE	19
4.2.4	Non-functional Property	19
4.3	ARCHITECTURE JUSTIFICATION	20
4.3.1	Simulation Scenarios.....	20
4.3.2	Simulation Models.....	21
4.3.3	Simulation Model Elements.....	21
4.3.4	Simulation Results	21
4.3.5	Specialty Analysis Context	21
4.3.6	Specialty Analysis Models	21
4.3.7	Specialty Model elements	21
4.3.8	Specialty Analysis Results	21
4.3.9	Viewpoint Analysis Context.....	21
4.3.10	Architecture viewpoint Models	21
4.3.11	Viewpoint Elements	22
4.3.12	Viewpoint Analysis Results.....	22
4.4	SOLUTION ELEMENT NEED description	22
4.4.1	Solution Element Need Specification.....	23
4.4.2	Solution Element	23
4.4.3	External systems or other Elements.....	23
4.4.4	Element Capabilities.....	24

4.4.5	Functional Chains involving Element.....	24
4.4.6	Scenarios involving Element.....	24
4.4.7	Functions expected from Element	24
4.4.8	Element Functional Exchanges.....	24
4.4.9	Element Exchanged Items	24
4.4.10	Element Data	24
4.4.11	Element Modes & States.....	24
4.4.12	Element external Components Exchanges	25
4.4.13	Element external Interfaces.....	25
4.4.14	Element/neighbourhood Scenarios	25
4.4.15	Element/neighbourhood Physical Links	25
4.4.16	Element Textual Requirements	25
4.5	IVV	25
4.5.1	TESTS	26
4.5.2	IVV MASTERING	27
4.6	AGILE	28
4.6.1	Product Backlog.....	29
4.6.2	EPIC.....	29
4.6.3	User Story.....	29
4.6.4	Sprint/Increment Backlog	29
4.7	VARIABILITY	30
4.7.1	Variability Description.....	30
4.7.2	Operational Segmentation Variability	30
4.7.3	Portfolio Specification Variability.....	31
4.7.4	Solution Variability	31
4.7.5	Reference Configurations	31
4.7.6	Variability Choices.....	31
4.7.7	Features	31
4.7.8	Project Configuration	31

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 detailed view of main kinds of engineering data elaborated and exploited by Arcadia activities, along with their relations.

A more formal description of Arcadia language concepts (aka metamodel) is given in another companion document: 'Arcadia Language – MetaModel'.

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 toolled? For which purpose? With which benefits?	✓							
	Philosophy	✓	(✓)						
Principles and approach	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?	✓		(✓)	(✓)				
Details for implementation	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?	✓							
Hints for Deployment	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?	✓							
	Which major questions arise in projects applying Arcadia?								✓

✓ : fully detailed

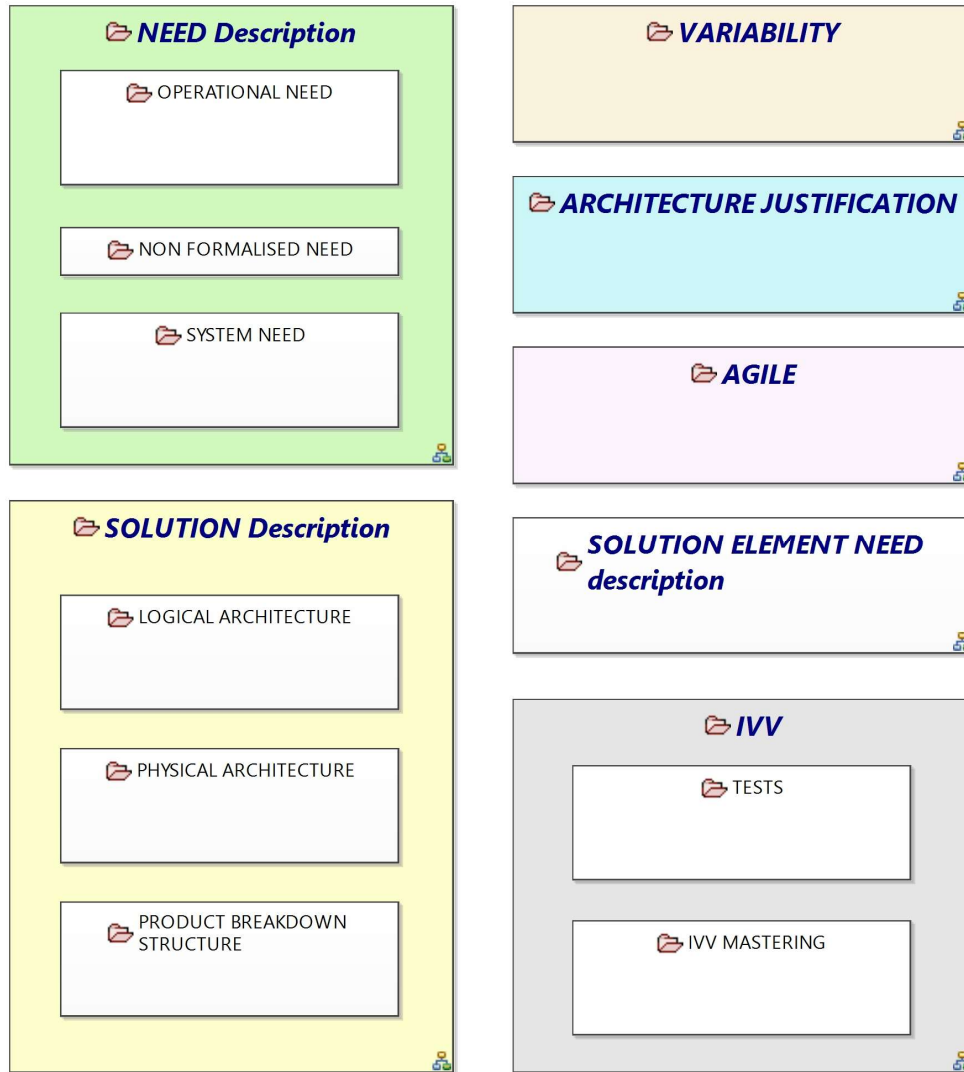
(✓) : simplified or partial

3

Arcadia Data Model Contents at a glance

This figure provides an overview of the structure of engineering data, grouped in assets.

NEED Description and SOLUTION Description assets constitute the model produced by Arcadia core perspectives activities.



Each asset and its detailed data are described below.

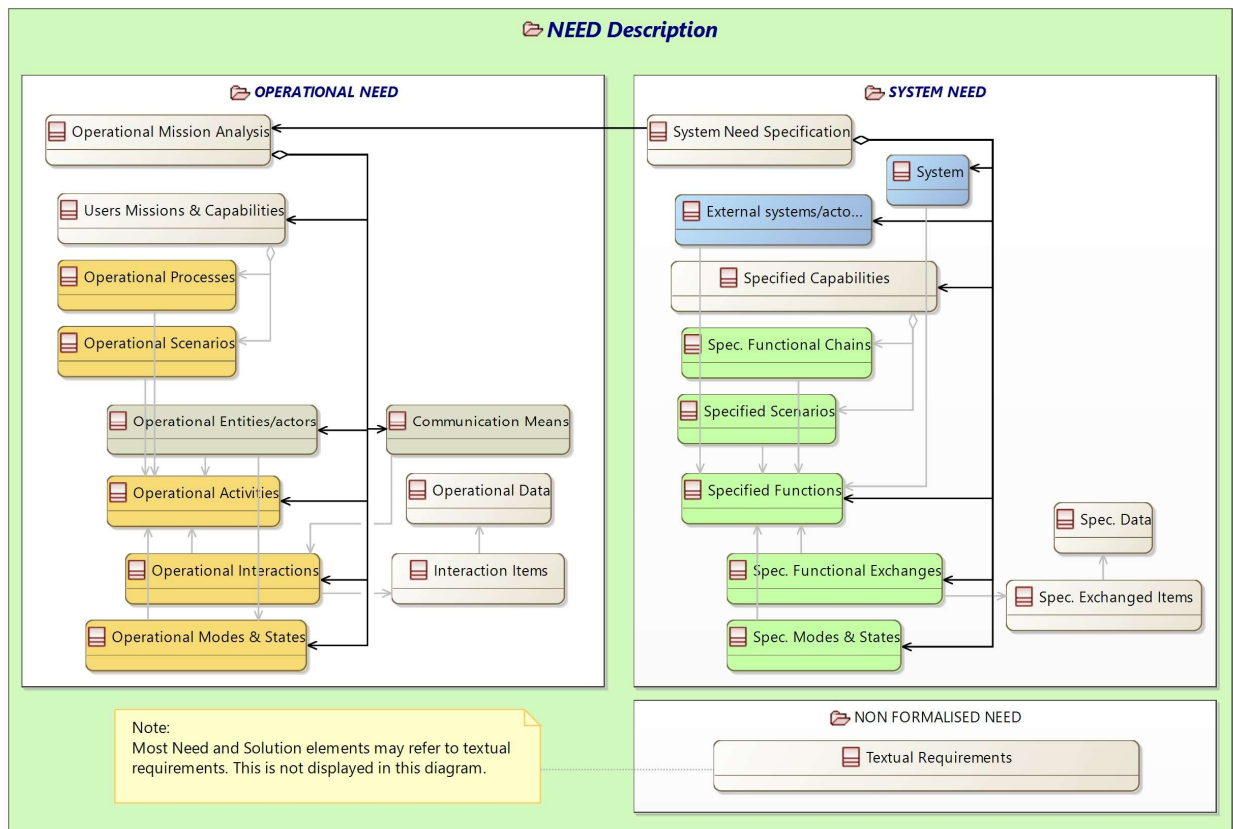
4 Arcadia Engineering Data Model

The engineering Data used in Arcadia perspectives and activities are described here, grouped in data sets by semantic proximity, either because they are created together, or in the same activities, or exploited for a common purpose.

4.1 NEED Description

This data set groups data and documents related to the definition of customer and stakeholders need.

This figure describes major concepts involved in Need Description, and their relations.



4.1.1 OPERATIONAL NEED

This data set groups data and documents related to the description of end users, customer and stakeholders goals, activities and environment.

4.1.1.1 **Operational Mission Analysis**

The operational analysis focuses on analyzing the stakeholders needs and goals, expected missions and activities, far beyond (and often before) customer requirements, and before specifying expectations on the system and its boundaries.

4.1.1.2 **Operational Entities/actors**

An operational entity is a real world entity or stakeholder, involved in a mission. An actor is a [usually human] non decomposable operational Entity.

4.1.1.3 **Users Missions & Capabilities**

A mission is a major goal to which the system is expected to contribute.

A capability is the ability of an operational entity, to provide a service that supports the achievement of a mission.

4.1.1.4 **Operational Processes**

An operational process is a logical organization of Interactions and Activities to fulfil an Operational Capability.

4.1.1.5 **Operational Scenarios**

An operational scenario is a time-ordered set of interactions between operational activities performed by operational entities to fulfil an Operational Capability.

4.1.1.6 **Operational Activities**

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

4.1.1.7 **Operational Interactions**

Operational interactions are conveyed through (allocated to) communication links between operational entities or actors

4.1.1.8 **Interaction Items**

A specified interaction contents describes what is expected to be exchanged by the entities or actors, and activities allocated to each of them.

4.1.1.9 **Operational Data**

elements of interactions or communications between entities or actors.

Data can be referenced and grouped into exchange items, the latter being conveyed by interactions or exchanges

4.1.1.10 **Communication Means**

a medium enabling interactions between entities and actors

4.1.1.11 **Operational Modes & States**

An automaton describes how modes & states are linked by transitions; it is linked to the appropriate object (actor, entity, system, component...);

functions or operational activities being available in this mode or state can be described.

4.1.2 **SYSTEM NEED**

This data set groups data and documents related to the definition of expectations on the system contents and boundaries, as required by customer and other stakeholders.

4.1.2.1 **System Need Specification**

The system need analysis and specification focuses on the system itself, in order to define how it can contribute to satisfy the former operational needs, along with its expected behavior and qualities.

4.1.2.2 **External systems/actors in SA**

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

4.1.2.3 **Specified Capabilities**

A specified capability is the ability required from the system, to provide a service that supports the achievement of a mission.

4.1.2.4 **Spec. Functional Chains**

A functional chain describes a usage of the system in a given context. It is a logical organization of functions and functional exchanges to fulfil a Capability.

4.1.2.5 **Specified Scenarios**

A specified scenario describes a usage of the system in a given context. It is a time-ordered set of functional exchanges between functions, or between the system and external actors, to fulfil a Capability.

4.1.2.6 **System**

The system of interest or solution to be delivered to customers (to be considered as a generic name: could be a software, a service, etc.).

4.1.2.7 **Specified Functions**

A specified function is an action, an operation or a service expected from the system, or fulfilled by an actor interacting with the system.

4.1.2.8 **Spec. Functional Exchanges**

A specified functional exchange is an interaction from a source function likely to deliver exchange items to another function (information, signal, material, torque...).

4.1.2.9 **Spec. Exchanged Items**

A specified exchange contents describes what is expected to be exchanged by the system, actors and functions allocated to each of them.

4.1.2.10 **Spec. Data**

elements of interactions or communications between system and actors (or inside the system).

Data can be referenced and grouped into exchange items, the latter being conveyed by interactions or exchanges

4.1.2.11 **Spec. Modes & States**

An automaton describes how modes & states are linked by transitions; it is linked to the appropriate object (actor, entity, system, component...);

functions or operational activities being available in this mode or state can be described.

4.1.3 **NON FORMALISED NEED**

This data set groups data and documents related to non formalised description of the need, such as textual requirements, plain documents and text, etc.

4.1.3.1 **Textual Requirements**

A textual requirement is an unformalised description of an expectation on the system or solution to be delivered.

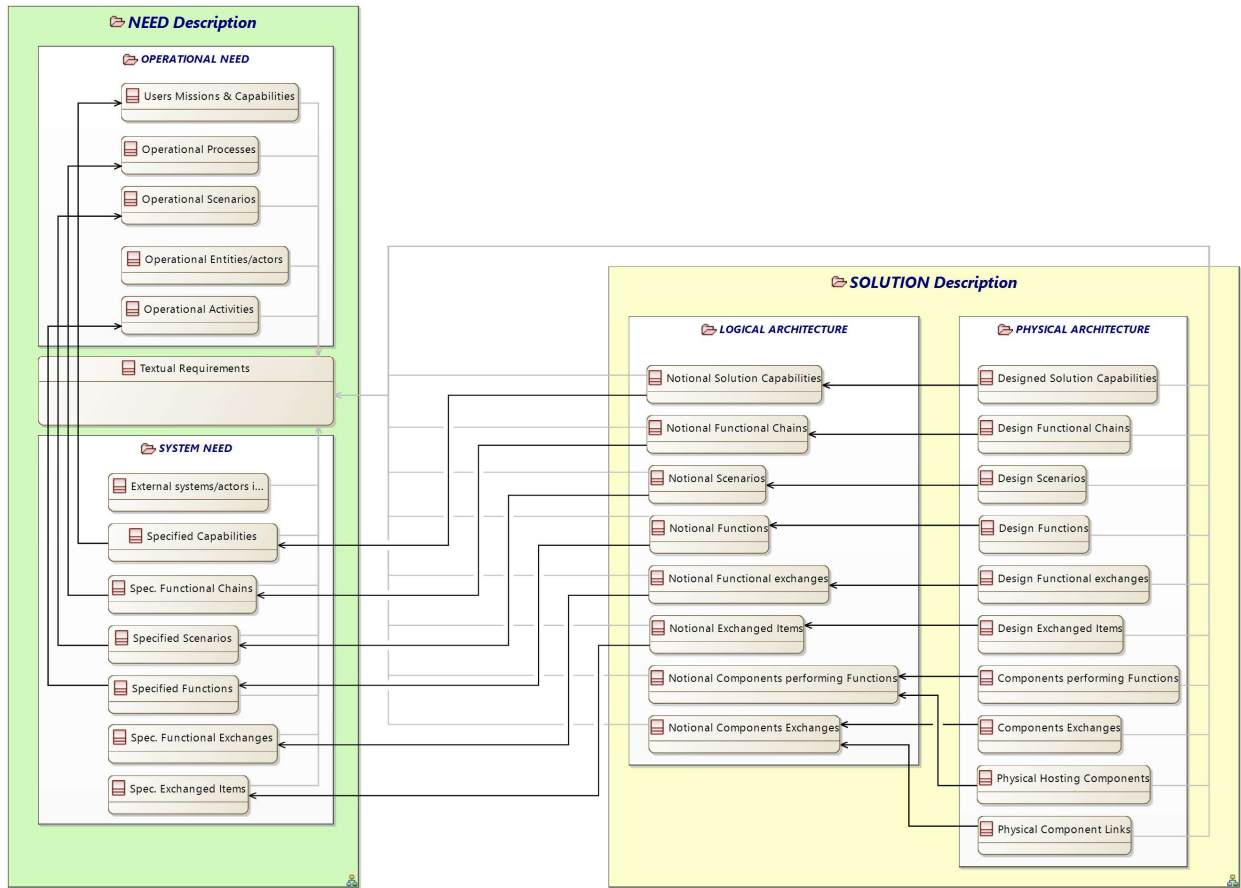
4.1.4 **[non functional] Constraint**

An expectation to be fulfilled by one or several model elements.

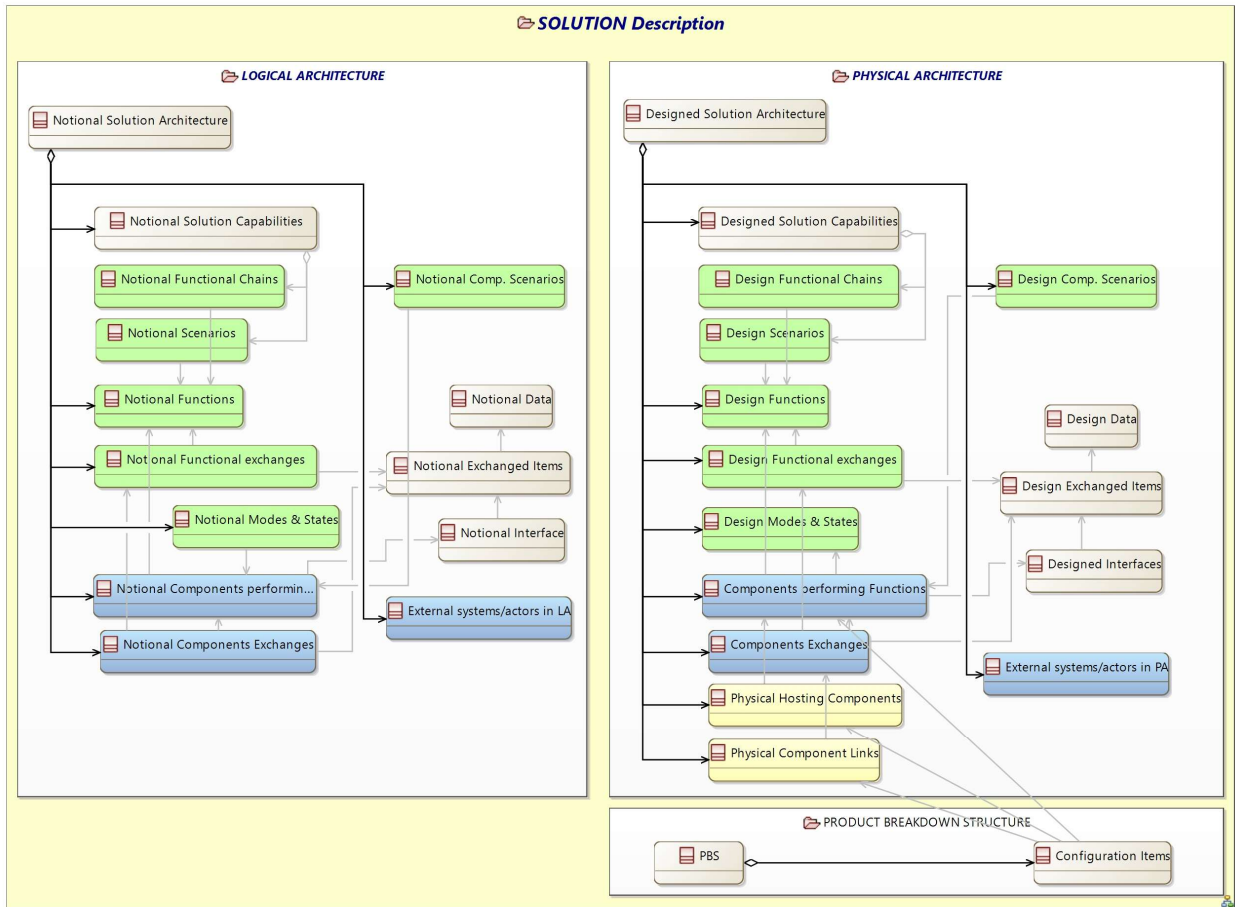
4.2 **SOLUTION Description**

This data set groups data and documents related to the description of major elements defining the solution as designed.

This figure describes major concepts involved in Need / Solution Traceability, and their relations.



This figure describes major concepts involved in Solution Description, and their relations.



4.2.1 LOGICAL ARCHITECTURE

This data set groups data and documents related to the description of major elements defining the notional solution, in a synthetic manner.

4.2.1.1 Notional Solution Architecture

The designed solution architecture describes the component breakdown of the system, the designed behavior of each component and related interfaces.

4.2.1.2 External systems/actors in LA

Users & actors can be decomposed in a tree structure. The system is a specific kind of actor.

4.2.1.3 Notional Solution Capabilities

A designed solution capability is the way the system provides a service that supports the achievement of a mission, notably specified capabilities.

4.2.1.4 **Notional Functional Chains**

A design functional chain describes the behavior of the system in a given context. It is a logical organization of functions and functional exchanges to fulfil a Capability.

4.2.1.5 **Notional Scenarios**

A design scenario describes the behavior of the system in a given context. It is a time-ordered set of functional exchanges between functions, or between the system and external actors, to fulfil a Capability.

4.2.1.6 **Notional Functions**

A design function is an action, an operation or a service fulfilled by the system, or by an actor interacting with the system.

4.2.1.7 **Notional Functional exchanges**

A design functional exchange is an interaction from a source function likely to deliver exchange items to another function (information, signal, material, torque...).

4.2.1.8 **Notional Exchanged Items**

A design exchange contents describes what is exchanged by the system, system components, actors and functions allocated to each of them.

4.2.1.9 **Notional Data**

elements of interactions or communications between system, components and/or actors.

Data can be referenced and grouped into exchange items, the latter being conveyed by interactions or exchanges

4.2.1.10 **Notional Modes & States**

An automaton describes how modes & states are linked by transitions; it is linked to the appropriate object (actor, entity, system, component...);

functions or operational activities being available in this mode or state can be described.

4.2.1.11 **Notional Components performing Functions**

A component performing functions (behavioral component) is a constituent of the system contributing to its behaviour and properties, by implementing functions interacting with other components.

4.2.1.12 **Notional Components Exchanges**

A (behavioral) component exchange is an interaction between two components likely to exchange some items (information, signal, material, torque...). It realises functional exchanges between functions allocated to source and target components.

4.2.1.13 **Notional Interface**

Interfaces group various exchange items, and are allocated to behavioural component exchanges

4.2.1.14 **Notional Comp. Scenarios**

A component scenario describes the behavior of the system in a given context. It is a time-ordered set of exchanges between components, or between the system and external actors, to fulfil a Capability.

4.2.2 **PHYSICAL ARCHITECTURE**

This data set groups data and documents related to the description of elements defining the finalised solution, at a level of detail sufficient to develop, integrate and verify it.

4.2.2.1 **Designed Solution Architecture**

The designed solution architecture describes the component breakdown of the system, the designed behavior of each component and related interfaces.

4.2.2.2 **External systems/actors in PA**

Users & actors can be decomposed in a tree structure. The system is a specific kind of actor.

4.2.2.3 **Designed Solution Capabilities**

A designed solution capability is the way the system provides a service that supports the achievement of a mission, notably specified capabilities.

4.2.2.4 **Design Functional Chains**

A design functional chain describes the behavior of the system in a given context. It is a logical organization of functions and functional exchanges to fulfil a Capability.

4.2.2.5 **Design Scenarios**

A design scenario describes the behavior of the system in a given context. It is a time-ordered set of functional exchanges between functions, or between the system and external actors, to fulfil a Capability.

4.2.2.6 **Design Functions**

A design function is an action, an operation or a service fulfilled by the system, or by an actor interacting with the system.

4.2.2.7 **Design Functional exchanges**

A design functional exchange is an interaction from a source function likely to deliver exchange items to another function (information, signal, material, torque...).

4.2.2.8 **Design Exchanged Items**

A design exchange contents describes what is exchanged by the system, system components, actors and functions allocated to each of them.

4.2.2.9 **Design Data**

elements of interactions or communications between system, components and/or actors.

Data can be referenced and grouped into exchange items, the latter being conveyed by interactions or exchanges

4.2.2.10 **Design Modes & States**

An automaton describes how modes & states are linked by transitions; it is linked to the appropriate object (actor, entity, system, component...);

functions or operational activities being available in this mode or state can be described.

4.2.2.11 **Components performing Functions**

A component performing functions (behavioral component) is a constituent of the system contributing to its behaviour and properties, by implementing functions interacting with other components.

4.2.2.12 **Components Exchanges**

A (behavioral) component exchange is an interaction between two components likely to exchange some items (information, signal, material, torque...). It realises functional exchanges between functions allocated to source and target components.

4.2.2.13 **Designed Interfaces**

Interfaces group various exchange items, and are allocated to behavioural component exchanges

4.2.2.14 **Design Comp. Scenarios**

A component scenario describes the behavior of the system in a given context. It is a time-ordered set of exchanges between components, or between the system and external actors, to fulfil a Capability.

4.2.2.15 **Physical Hosting Components**

A physical hosting component is a component hosting behavioural components, delivering them required resources and support for interactions or communications between them.

4.2.2.16 **Physical Component Links**

A physical link is a communication means between two physical hosting components.

4.2.3 **PRODUCT BREAKDOWN STRUCTURE**

This data set groups data and documents related to the structure of configuration items constituting the final solution.

4.2.3.1 **PBS**

The Organisation of the System-of-Interest (end products components) in a tree frame is the End Product Breakdown Structure (EPBS). The EPBS is the part of the PBS related to the System under consideration in the development process.

4.2.3.2 **Configuration Items**

A Configuration Item is any hardware, software, or combination of both that satisfies an end use function and is designated for separate configuration management. Configuration items are typically referred to by an alphanumeric identifier which also serves as the unchanging base for the assignment of serial numbers to uniquely identify individual units of the CI.

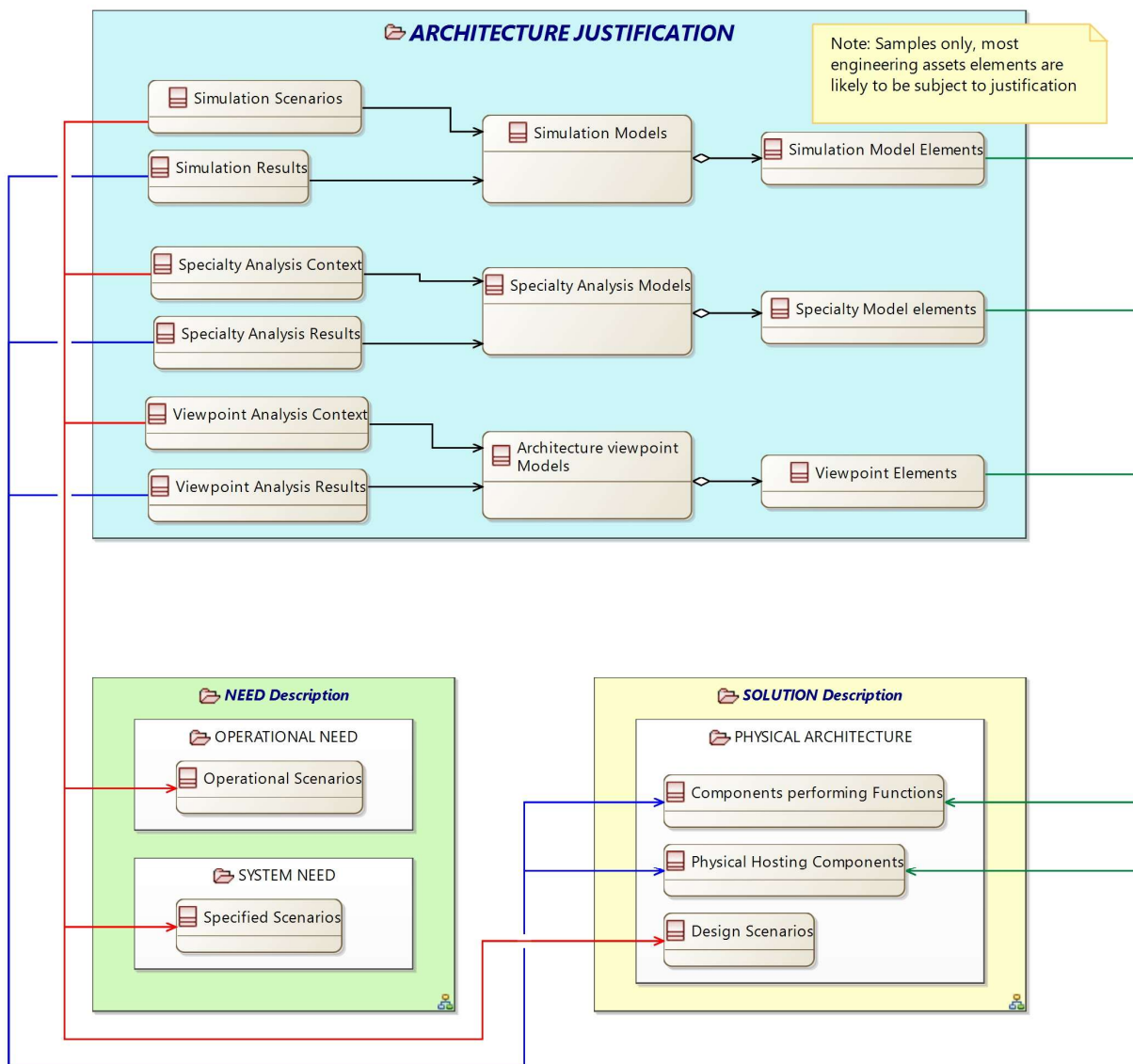
4.2.4 **Non-functional Property**

A characterisation of a model element, such as a value, an attribute, etc.

4.3 ARCHITECTURE JUSTIFICATION

This data set groups data and documents related to analyses and demonstrations verifying expected properties and quality of the solution as designed, including verification conditions and results.

This figure describes major concepts involved in Architecture Justification, and their relations.



4.3.1 Simulation Scenarios

The description of environment, context and conditions in which some characteristics of the system or design have to be assessed, along with expected results

4.3.2 **Simulation Models**

A model representative of the system from a specific perspective, that can be subject to execution in order to check system properties

4.3.3 **Simulation Model Elements**

Elements of the simulation model, subject of simulation and analyses

4.3.4 **Simulation Results**

The result of assessment of the system or design under simulation scenarios

4.3.5 **Specialty Analysis Context**

The description of environment, context and conditions in which some characteristics of the system or design have to be assessed, along with expected results

4.3.6 **Specialty Analysis Models**

A model representative of the system from a specific perspective, that can be subject to dedicated analysis in order to check system properties

4.3.7 **Specialty Model elements**

Elements of the specialty analysis model, subject of analyses

4.3.8 **Specialty Analysis Results**

The result of assessment of the system or design under specialty analysis

4.3.9 **Viewpoint Analysis Context**

The description of environment, context and conditions in which some characteristics of the system or design have to be assessed, along with expected results

4.3.10 **Architecture viewpoint Models**

A model representative of the system from a specific viewpoint perspective, that can be subject to analysis and confrontation to other viewpoints in order to find the best compromise

4.3.11 **Viewpoint Elements**

Elements of the viewpoint analysis model, subject of analyses

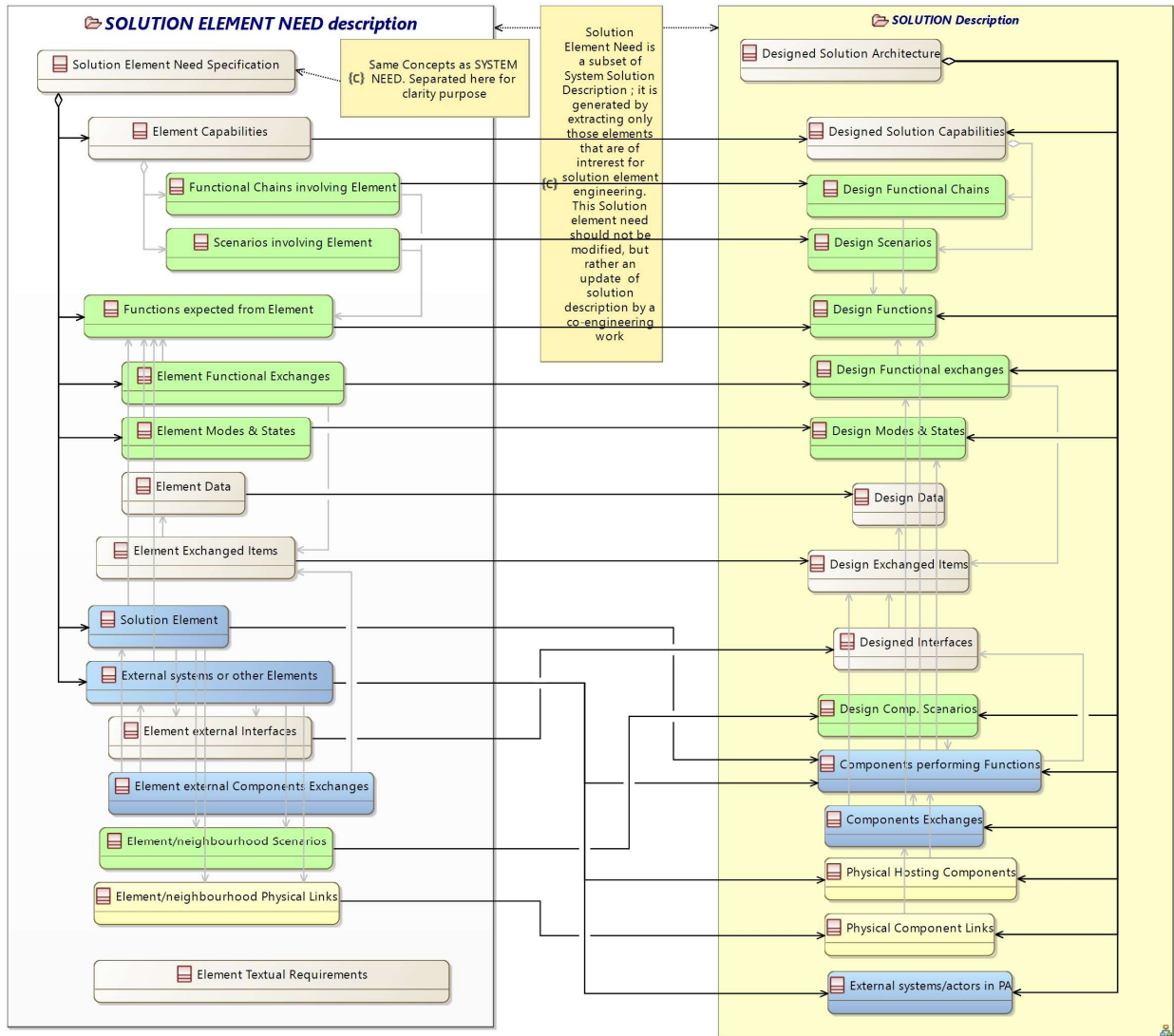
4.3.12 **Viewpoint Analysis Results**

The result of assessment of the system or design under viewpoint analysis

4.4 **SOLUTION ELEMENT NEED description**

This data set groups data and documents related to the definition of expectations on a system element, based on the former system solution definition.

This figure describes major concepts involved in Subsystems SW HW Definition, and their relations.



4.4.1 Solution Element Need Specification

The solution element need analysis and specification focuses on the element itself, in order to define how it can contribute to satisfy the former system solution description, along with its expected behavior and qualities.

4.4.2 Solution Element

The element, part of the solution, subject of this engineering

4.4.3 External systems or other Elements

An external actor is an external entity, organisation, system, or human, interacting with the system element via its interfaces

4.4.4 **Element Capabilities**

A specified capability is the ability required from the element, to provide (or contribute to) a service that supports the achievement of a mission.

4.4.5 **Functional Chains involving Element**

A functional chain describes a usage of the element in a given context. It is a logical organization of functions and functional exchanges to fulfil a Capability.

4.4.6 **Scenarios involving Element**

A specified scenario describes a usage of the element in a given context. It is a time-ordered set of functional exchanges between functions, or between the element and external actors or elements, to fulfil a Capability.

4.4.7 **Functions expected from Element**

A specified function is an action, an operation or a service expected from the element, or fulfilled by an actor interacting with the system or element.

4.4.8 **Element Functional Exchanges**

A specified functional exchange is an interaction from a source function likely to deliver exchange items to another function (information, signal, material, torque...).

4.4.9 **Element Exchanged Items**

A specified exchange contents describes what is expected to be exchanged by the element, actors and functions allocated to each of them.

4.4.10 **Element Data**

elements of interactions or communications between the element and actors (or other elements inside the system).

Data can be referenced and grouped into exchange items, the latter being conveyed by interactions or exchanges

4.4.11 **Element Modes & States**

An automaton describes how modes & states are linked by transitions; it is linked to the element;

functions or operational activities being available in this mode or state can be described.

4.4.12 **Element external Components Exchanges**

A (behavioral) component exchange is an interaction between two elements or components likely to exchange some items (information, signal, material, torque...). It realises functional exchanges between functions allocated to source and target components.

4.4.13 **Element external Interfaces**

Interfaces group various exchange items, and are allocated to behavioural component exchanges

4.4.14 **Element/neighbourhood Scenarios**

A scenario describes a usage of the element in a given context. It is a time-ordered set of functional exchanges between functions, or between the element, other elements and external actors, to fulfil a Capability.

4.4.15 **Element/neighbourhood Physical Links**

A physical link is a communication means between two physical hosting components (here, between the element and its neighbourhood).

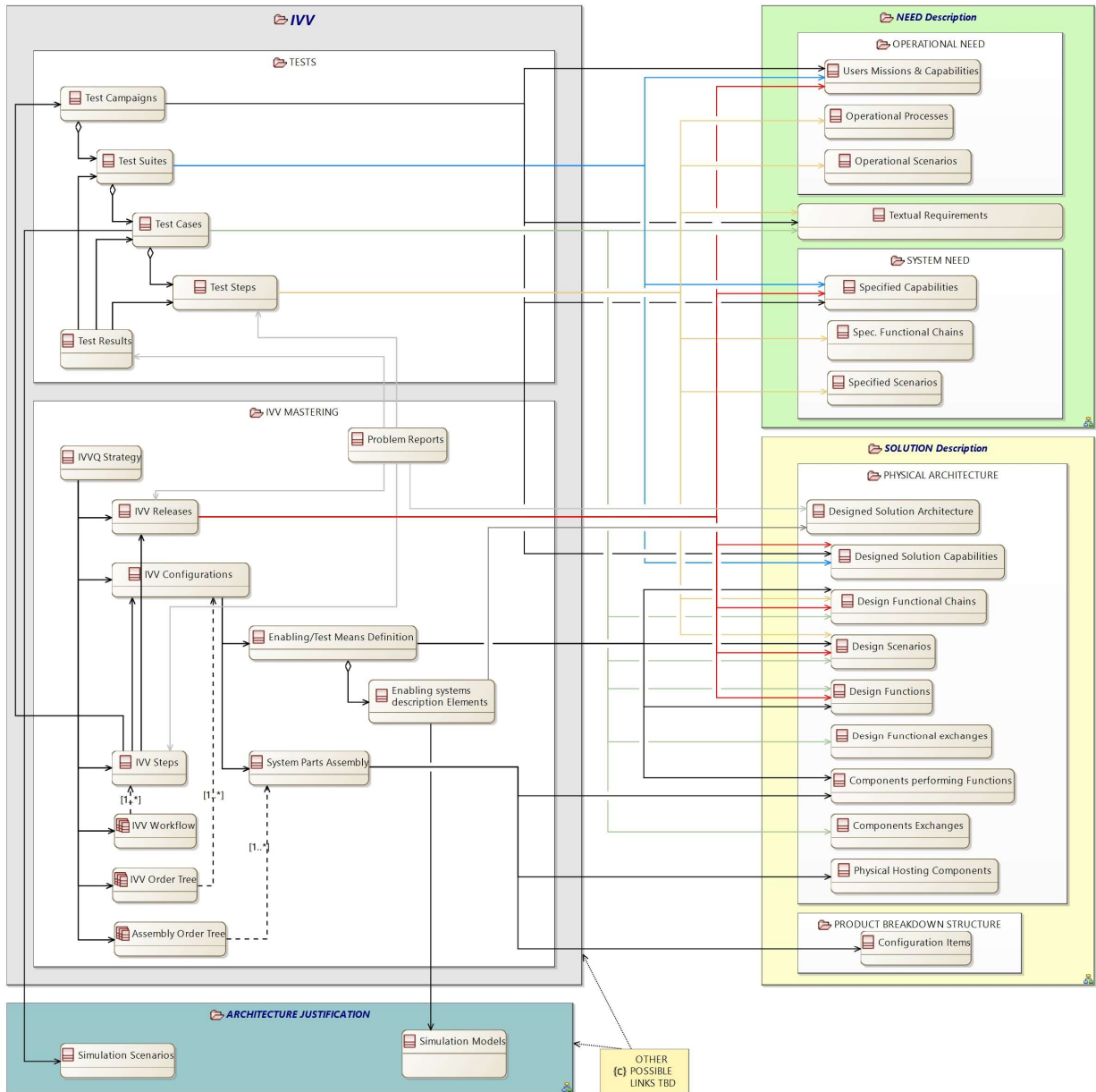
4.4.16 **Element Textual Requirements**

A textual requirement is an unformalised description of an expectation on the element, system or solution to be delivered.

4.5 **IVV**

This data set groups data and documents related to Integration, Verification and Validation (IVV) tasks.

This figure describes major concepts involved in Integration Verification Validation, and their relations.



4.5.1 TESTS

This data set groups data and documents related to the definition and run of tests in IVV.

4.5.1.1 Test Campaigns

a set of test suites required to run an IVV step

4.5.1.2 Test Suites

a set of test cases sharing same test objectives and to be run together.

4.5.1.3 **Test Cases**

a scenario describing an ordered sequence of test steps stimulating the system and analysing its behavior

4.5.1.4 **Test Steps**

an elementary interaction between the system and external users, systems or test means, either stimulating the system or analysing its behavior

4.5.1.5 **Test Results**

The collection of observed behavior of the system at each step of test cases, compared to expected behavior

4.5.2 **IVV MASTERING**

This data set groups data and documents related to the definition of the IVV strategy and elements supporting IVV.

4.5.2.1 **IVVQ Strategy**

The justified set of all steps in IVVQ and the required enablers and prerequisites of each step: configuration items, expected functional contents, workflow/order of assembly and of integration, enablers systems and test means, etc.

Justification relies notably on dependencies between required elements, customer and industrial value analysis. The strategy may be reconsidered according to IVV ups and downs, delays, difficulties, and so on.

4.5.2.2 **IVV Releases**

The description of the system contents of a step in the IVV flow, according to the IVV strategy, in terms of required components, functional contents and non functional properties/performances

4.5.2.3 **IVV Configurations**

The description of the system contents and test means required for an IVV step

4.5.2.4 **IVV Steps**

The description of a step in the IVV flow, according to the IVV strategy, in order to setup the required test environment

4.5.2.5 **System Parts Assembly**

A group of system elements to be assembled in order to build a larger system component

4.5.2.6 **Enabling systems description Elements**

Elements describing the enabling systems and test means. The description is similar to the system description, and in many cases, can be deduced/specified from the system description itself.

4.5.2.7 **Enabling/Test Means Definition**

The definition of elements not included into the set of system elements to be tested, that are necessary in order to test system elements. They can be external systems, benches, simulation software, stimulation and outputs analysis devices, and also system elements already verified/validated, etc.

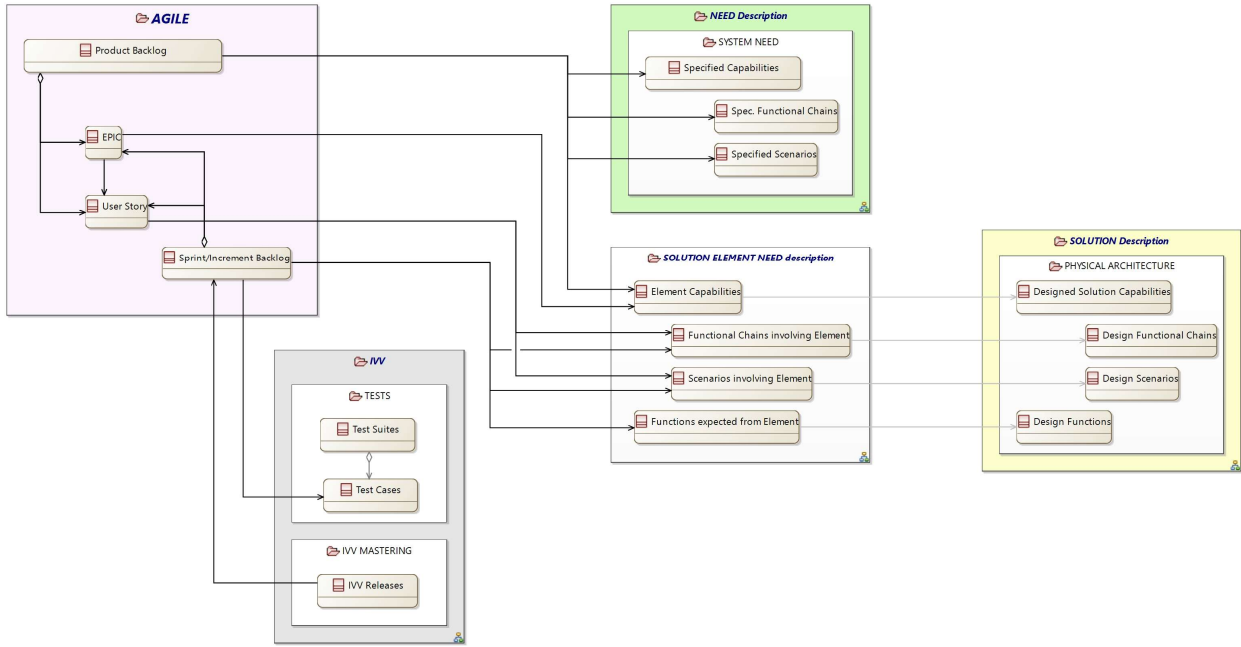
4.5.2.8 **Problem Reports**

The recording of an unexpected property or behavior of the system

4.6 **AGILE**

This data set groups data and documents related to the support of agile practices, by means of EPIC and user stories defined incrementally through various sprints.

This figure describes major concepts involved in Agile & Software, and their relations.



4.6.1 Product Backlog

The Product Backlog is an ordered list of everything that is known to be needed in the product. It is the single source of requirements for any changes to be made to the product. The Product Backlog lists all features, functions, requirements, enhancements, and fixes that constitute the changes to be made to the product in future releases.

Source www.scrumguides.org

4.6.2 EPIC

An epic is a large user story that cannot be delivered as defined within a single iteration or is large enough that it can be split into smaller user stories.

Source www.agilealliance.org

4.6.3 User Story

In consultation with the customer or product owner, the team divides up the work to be done into functional increments called “user stories.”

Each user story is expected to yield, once implemented, a contribution to the value of the overall product, irrespective of the order of implementation.

Source www.agilealliance.org

4.6.4 Sprint/Increment Backlog

The Sprint Backlog is the set of Product Backlog items selected for the Sprint, plus a plan for delivering the product Increment and realizing the Sprint Goal.

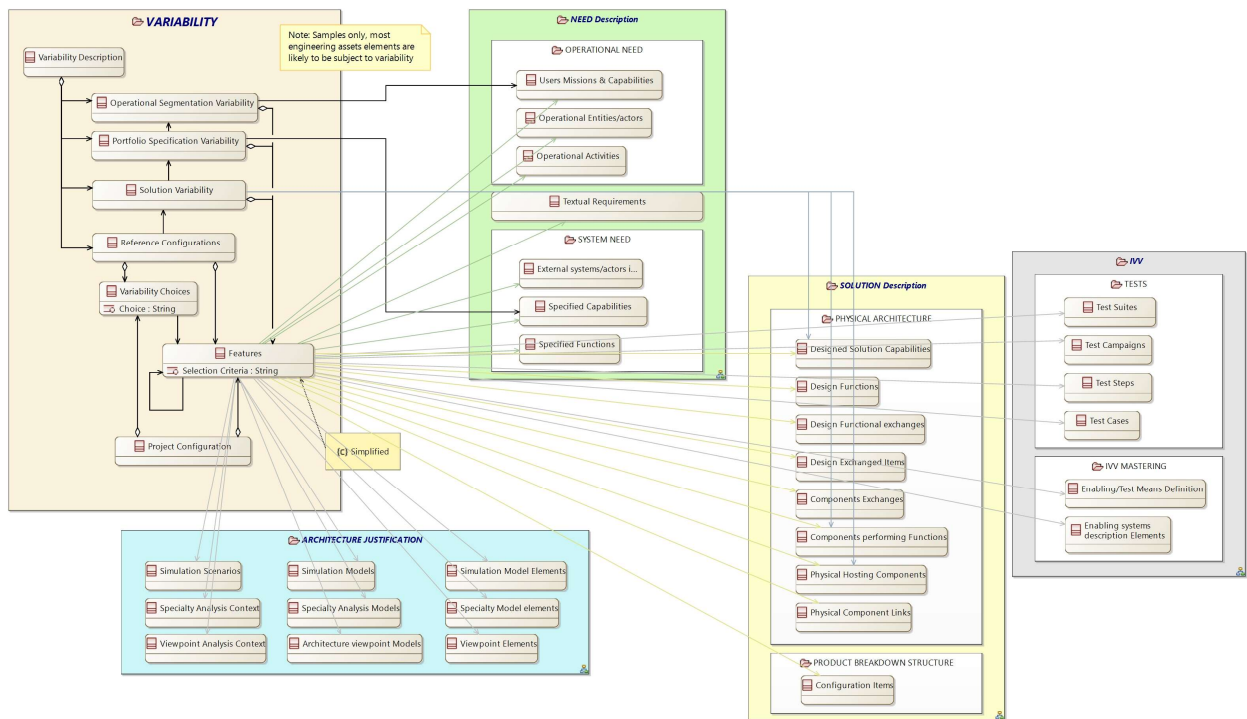
The Increment is the sum of all the Product Backlog items completed during a Sprint and the value of the increments of all previous Sprints.

Source www.scrumguides.org

4.7 VARIABILITY

This data set groups data and documents related to Product Line Engineering and associated Variability support.

This figure describes major concepts involved in Product Line Engineering, and their relations.



4.7.1 Variability Description

The description of parts or characteristics of the need and solution, that are either common to a set of customers or projects, or specific of one or another

4.7.2 Operational Segmentation Variability

The description of user needs and capabilities that are either common to a market segment, or specific of one or another

4.7.3 **Portfolio Specification Variability**

The description of user-oriented features and capabilities that are either common to a set of system portfolio elements, or specific of one or another

4.7.4 **Solution Variability**

The description of parts or characteristics of the system that are either common to a set of customers or projects, or specific of one or another

4.7.5 **Reference Configurations**

The description of a selection of parts or characteristics of the system for a given set of customers or projects, or specific of one or another

4.7.6 **Variability Choices**

The selection of one alternative among several possible regarding a given feature or option

4.7.7 **Features**

a characteristic of the need or solution that can either be optional, or subject to several alternatives

4.7.8 **Project Configuration**

The description of a selection of parts or characteristics of the system for a given project

