



NATO Architecture Framework Version 4

Modeling Guidelines for use of the UAF DMM

Architecture CapabilityTeam
Consultation, Command & Control Board
March 2019

Document Version 1.0.0

Table of contents

1	Introduction.....	5
1.1	Introduction and Overview	5
1.2	How to read.....	8
2	Concept Viewpoints	10
2.1	Capability Taxonomy (C1).....	11
2.2	C2 – Enterprise Vision.....	13
2.3	C3 – Capability Dependencies	15
2.4	C4 – Standard Processes.....	16
2.5	C5 – Effect.....	18
2.6	C7 - Performance Parameters	20
2.7	C8 - Capability Assumptions	21
2.8	Cr – Capability Roadmap	22
2.9	Elements Concept View.....	23
3	Service Specification Viewpoints.....	25
3.1	S1 - Service Taxonomy.....	26
3.2	S3 – Service Interfaces.....	28
3.3	S4 – Service Functions	30
3.4	S5 – Service States	31
3.5	S6 – Service Interactions	33
3.6	S7 – Service Interface Parameters.....	35
3.7	S8 – Service Policy	37
3.8	Sr – Service Roadmap	38
3.9	C1 - S1 – Capability to Service Mapping.....	40
3.10	Elements	41
4	Logical Specification Viewpoints	42
4.1	L1 – Node Types	44
4.2	L2 – Logical Scenario.....	46
4.3	L2-L3	48
4.4	L3 – Node Interactions	49
4.5	L4 - Logical Activities	52
4.6	L5 - Logical States	54
4.7	L6 – Logical Sequence.....	56
4.8	L7 – Logical Data Model	58
4.9	L8 – Logical Constraints	60

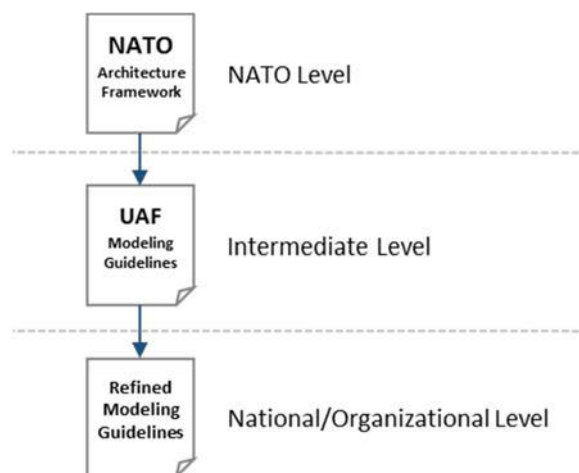
4.10	Lr – Lines of Deployment	62
4.11	Elements	64
5	Physical Resource specification Viewpoints.....	67
5.1	P1 – Resource Types.....	69
5.2	P2 – Resource Structure	71
5.3	P3 – Resource Connectivity	73
5.4	P4 – Resource Functions	75
5.5	P5 – Resource States	77
5.6	P6 – Resource Sequence	79
5.7	P7 – Physical Data Model	81
5.9	P8 – Resource Constraints.....	83
5.10	Pr – Configuration Management.....	85
5.11	L4-P4 – Activity to Function Mapping	87
5.12	Elements	89
6	Architecture Meta-Data Viewpoints	93
6.1	A1 – Meta-Data Definitions.....	94
6.2	A2 – Architecture Products	95
6.3	A3 – Architecture Correspondence	97
6.4	A4 – Methodology Used.....	99
6.5	A5 – Architecture Status.....	100
6.6	A6 – Architecture Version	101
6.7	A7 – Architecture Meta-Data	103
6.8	A8 – Standards.....	104
6.9	Ar – Architecture Roadmap.....	106
6.10	Elements.....	107
7	Bibliography.....	109
	Annex A Mapping of the NAF 3.1 Elements to the UAF DMM Elements.....	110
	Annex B NAFv4 SMM – One Pager	124
	Annex C NAFv4 SMM Elements.....	125

1 INTRODUCTION

1.1 Introduction and Overview

1.1.1 About this Document

This document refers to chapter 4 of the NATO Architecture Framework (NAF) version 4 and provides guidance how to develop NAFv4 compliant architectures based on the UAF domain meta-model (DMM). Guidance is provided on an intermediate level, i.e. meta-model specific but tool agnostic and not regarding nation or organization specific extensions.



The NATO Architecture Framework document concerns all nations or organizations using the NAF, regardless the used meta-model. It is provided and maintained by the NATO Architecture Capability Team (ACaT). The UAF modeling guidelines (i.e. this document) concern all nations or organizations using the UAF domain meta-model (DMM) for their architecture development. Because tools and specific extensions of framework and meta-model may differ between nations or organizations, these aspects are not considered in this document. Nations or organizations need to provide and maintain additional guidelines in order to cover these aspects by their own.

This document does not regard the entire UAF DMM but rather the parts needed for NAFv4 modeling (extracts). When extending the meta-model, it is advisable to first consider the parts that go beyond NAFv4 before implementing own extensions which may duplicate parts of UAF DMM.

Addressed readers are

- developers of national/organizational guidelines,
- implementers of tool specific UAF profiles,
- modeling enthusiasts and experts.

1.1.2 The NAFv4 Viewpoints

The NAF Grid Representation is a two-dimensional classification scheme for the standardized NAF viewpoints, which serve as the baseline for any NAF-Compliant architecture effort. However, the selection of Viewpoints must be tailored to the specific architecture effort, i.e. suitable Viewpoints need to be identified in the grid, and additional Viewpoints must be defined, if and when required. The grid approach presents the NAF viewpoints by Subjects of Concerns (rows) and by Aspects of Concerns

(columns). The NAF is arranged as a grid with columns as set of broad Model Kinds. For more information, see the figure below and the main document NATO Architecture Framework Version 4.

		Behaviour							
	Taxonomy	Structure	Connectivity	Processes	States	Sequences	Information	Constraints	Roadmap
Concepts	C1 Capability Taxonomy NAV-2, NCV-2	C2 Enterprise Vision NCV-1	C3 Capability Dependencies NCV-4	C4 Standard Processes NCV-6	C5 Effects NOV-6b		C7 Performance Parameters NCV-1	C8 Planning Assumptions	Cr Capability Roadmap NCV-3
	C1-S1 (NSOV-3)								
Service Specifications	S1 Service Taxonomy NAV-2, NSOV-1		S3 Service Interfaces NSOV-2	S4 Service Functions NSOV-3	S5 Service States NSOV-4b	S6 Service Interactions NSOV-4c	S7 Service I/F Parameters NSOV-2	S8 Service Policy NSOV-4a	Sr Service Roadmap
Logical Specifications	L1 Node Types NAV-2	L2 Logical Scenario NOV-2	L3 Node Interactions NOV-2, NOV-3	L4 Logical Activities NOV-5	L5 Logical States NOV-6b	L6 Logical Sequence NOV-6c	L7 Logical Data Model NSV-11a	L8 Logical Constraints NOV-6a	Lr Lines of Development NPV-2
				L4-P4 (NSV-5)					
Physical Resource Specifications	P1 Resource Types NAV-2, NSV-2a,7,9,12	P2 Resource Structure NOV-4, NSV-1	P3 Resource Connectivity NSV-2, NSV-6	P4 Resource Functions NSV-4	P5 Resource States NSV-10b	P6 Resource Sequence NSV-10c	P7 Physical Data Model NSV-11b	P8 Resource Constraints NSV-10a	Pr Configuration Management NSV-8
Architecture Meta-Data	A1 Meta-Data Definitions NAV-3	A2 Architecture Products	A3 Architecture Correspondence ISO42010	A4 Methodology Used NAF Ch2	A5 Architecture Status NAV-1	A6 Architecture Versions NAV-1	A7 Architecture Meta-Data NAV-1/3	A8 Standards NTV-1/2	Ar Architecture Roadmap

1.1.3 The UAF Viewpoints

The OMG UAF standard took the principles of UPDM and extended it to reach a wider audience. As well as encompassing DoDAF, MODAF, and NAF, UAF also includes views that can capture human machine interface and human factors concerns, security analysis and systems-of-systems lifecycle concepts. Because of this expansion in scope, it became necessary to change the way views were represented in UPDM. This resulted in the definition of a grid-based approach that separates the meta-model from the view specification and from a specific architecture framework representation.

As well as the NAF grid, the UAF grid presents viewpoints by Subject of Concerns (rows) and by Aspect of Concerns (columns). Comparing the both grids, the NAF viewpoints can be mapped to a subset of the UAF viewpoints (see figure below).

Consequently, the UAF based meta-model used in NAFv4 architectures also represents a subset of the complete UAF DMM. Therefore, this modeling guideline uses the NAF relevant meta-model extracts from the UAF DMM.

	Taxonomy Tx	Structure Sr	Connectivity Cn	Processes Pr	States St	Interaction Scenarios Is	Information If	Parameters Pm	Constraints Ct	Roadmap Rm	Traceability Tr		
Metadata Md	Metadata Definitions A1	Architecture Products A2	Architecture Correspondence A3	Methodology Used A4	Architecture Status A5	-	Logical Data Model L7 Physical Data Model P7	Performance Parameters C7 Service I/F Parameters S7	-	Architecture Roadmap Ar, A6	-		
Strategic St	Capability Taxonomy C1	Enterprise Vision C2	Capability Dependencies C3	Standard Processes C4-	Effects C5	-			Planning Assumptions C8	Capability Roadmap Cr	-		
Operational Op	Node Types L1	Logical Scenario L2, L2-L3	Node Interactions L3, L2-L3	Logical Activities L4	Logical States L5	Logical Sequence L6			Logical Constraints L8	-	-		
Services Sv	Service Taxonomy S1	-	Service Interfaces S3, S7	Service Functions S4	Service States S5	Service Interactions S6			Service Policy S8	Service Roadmap Sr	Service Traceability C1-S1		
Personnel Pr	Resource Types P1	Resource Structure P2	Resource Connectivity P3	Resource Functions P4	Resource States P5	Resource Sequence P6			-	-	-		
Resources Rs									Resource Constraints P8	Configuration Management Pr	Resource Traceability L4-P4		
Security Sc	-	-	-	-	-	-			-	-	-		
Projects PJ	-	-	-	-	-	-			-	Lines of Development Lr	-		
Standards Sd	Standards A8	-	-	-	-	-			-	Standards A8			
Actuals Resources Ar		Resource Structure P2	-		-					-	-	-	
Dictionary -													
Summary & Overview A5, A6, L2-L3													
Requirements -													

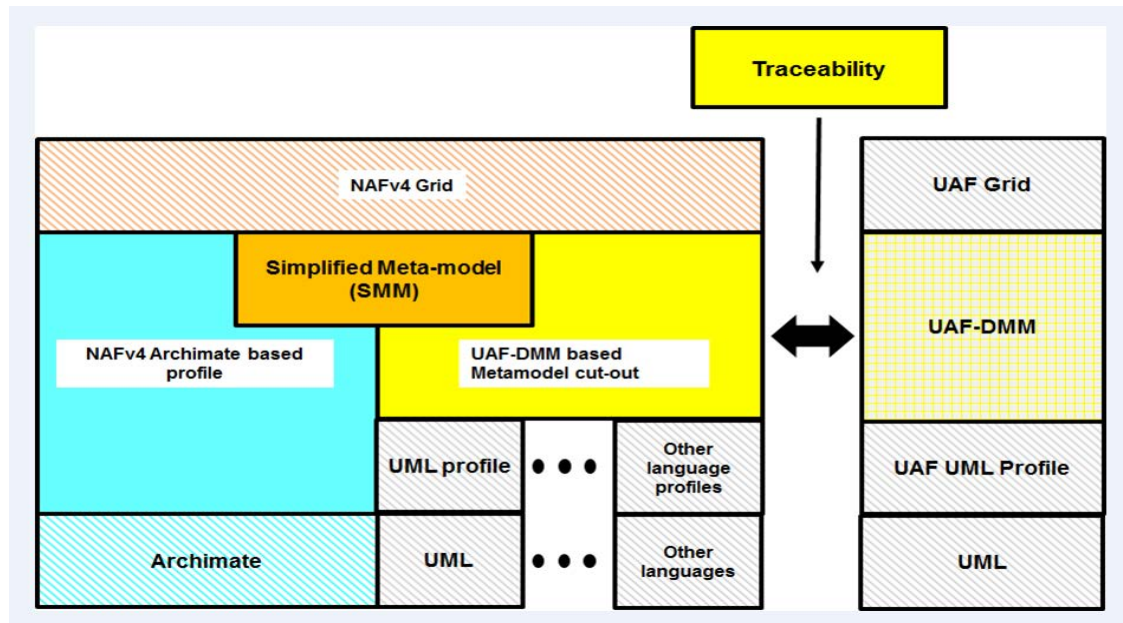
1.1.4 The NAFv4 Simplified Meta-Model

This Simplified Meta-Model (SMM) provides the terms and relationships for each viewpoint (NAFv4 Cell) of the NAFv4 Formalism (NAFv4 Grid).

The SMM allows development of NAFv4 modeling environment based on

- detailed meta-models and specifications like UAF Domain Meta-Model and Archimate specification,
- existing standardized languages (e.g. UML, sysML or BPMN), and
- NAF specific or tool specific languages.

Each SMM diagram is associated to a NAFv4 viewpoint. Additional diagrams are also included in the SMM in order to express canonic patterns used in several SMM diagrams.



1.2 How to Read

1.2.1 The Structure of this Document

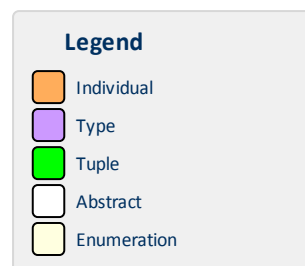
For each subject of concern (row of the NAFv4 grid) a DMM extract is provided. Each Viewpoint (cell in the NAFv4 grid) is described by separate subsections containing

- short description,
- concerns addressed,
- NAFv4 SMM,
- UAF DMM extract regarding the viewpoint, and
- modeling example.

At the end of each chapter, a detailed list of applicable concepts (elements and relations) is provided.

1.2.2 The Domain Meta-Model (DMM) Diagrams and Examples

The color scheme of DMM extracts and examples is provided in the following legend. It does correspond to the color scheme of UAF and MODEM.

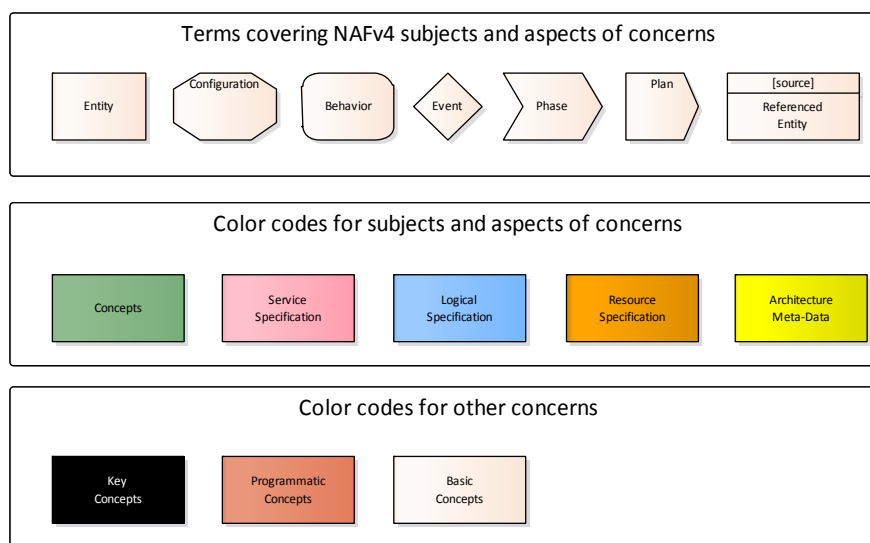


1.2.3 The NAFv4 SMM

Conventions to read the NAFv4 SMM are

- symbols which express the semantic of the terms,

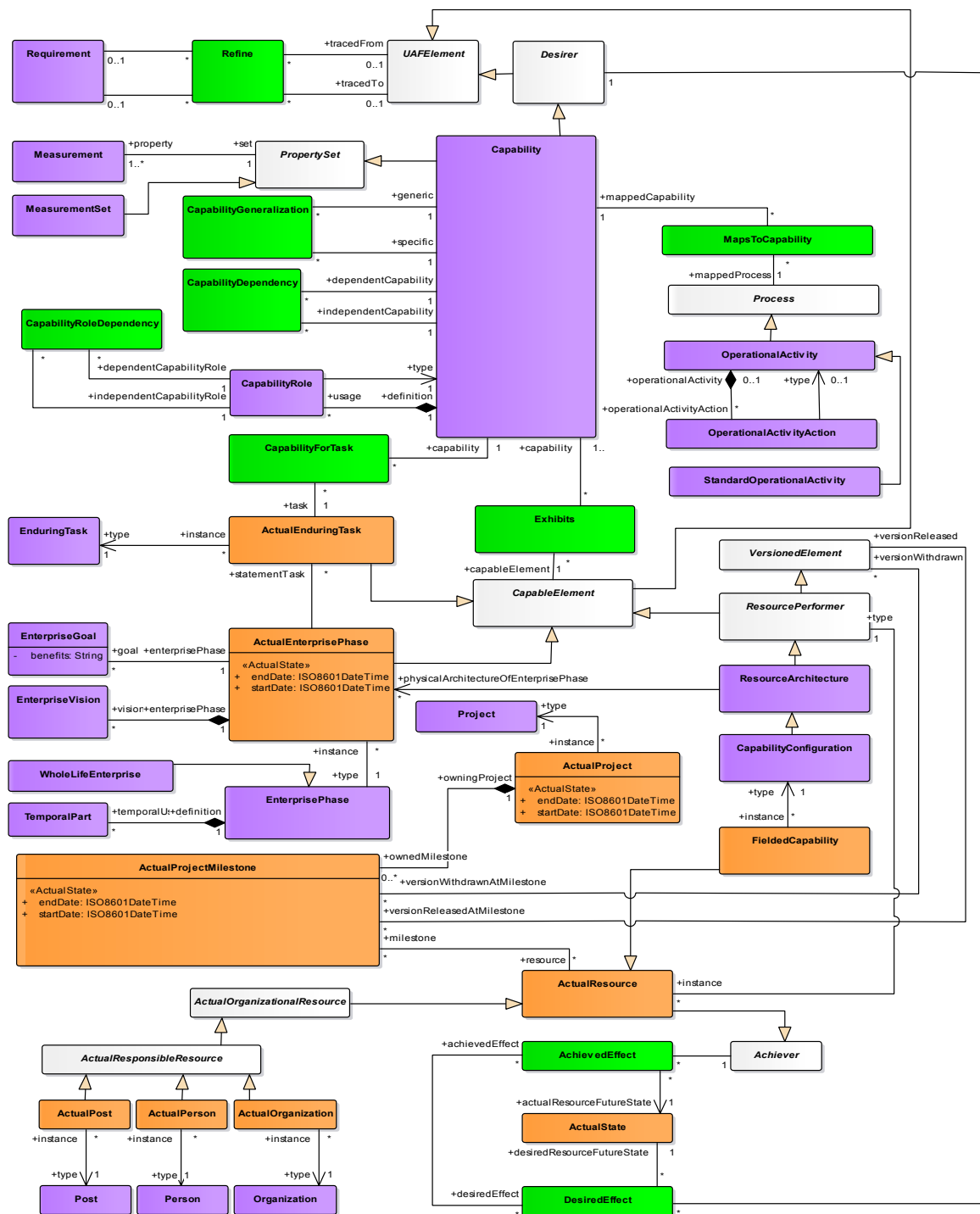
- colors which express the subjects of concerns, according to the NAFv4 Grid and
- other colors which express other useful concerns.



2 CONCEPT VIEWPOINTS

	Taxonomy		Behaviour				Information	Constraints	Roadmap
	C1	C2	C3	C4	C5				
Concepts	Capability Taxonomy NAV-2, NCV-2	Enterprise Vision NCV-1	Capability Dependencies NCV-4	Standard Processes NCV-6	Effects NOV-6b		Performance Parameters NCV-1	Planning Assumptions C8	Capability Roadmap NCV-3

The Viewpoints in the Concepts row of the NAF grid support the process of analyzing and optimizing the delivery of capability in line with the enterprise strategic intent.



2.1 Capability Taxonomy (C1)

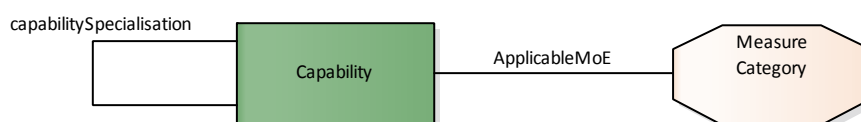
The C1 Viewpoint specifies all the capabilities that are referenced throughout one or more architectures – i.e. one C1 may provide the definitive list of capabilities for a number of logical and resource architectures. Capabilities are enduring and apply to all Enterprise phases listed in a C2, Enterprise Vision.

The capabilities may be organized into specialization hierarchies (taxonomies). Measures of Effectiveness (MoE) may be specified for each capability. Note that MoEs are inherited down a capability taxonomy.

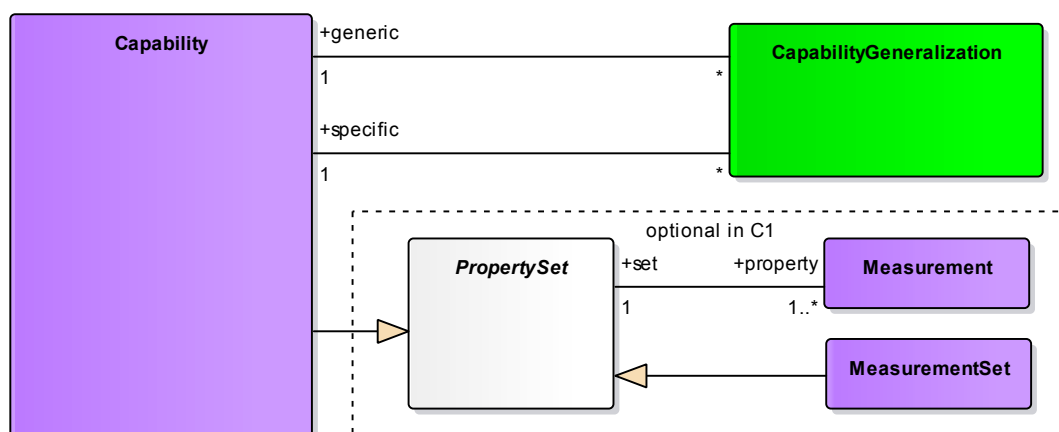
2.1.1 Concerns

- identification of capabilities,
- organization of capabilities into specialization hierarchies (taxonomies), and
- (optional) Measures of Effectiveness (MoE).

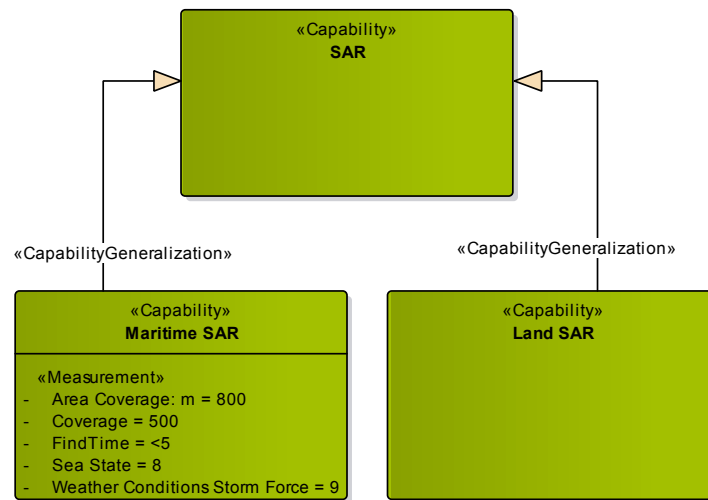
2.1.2 NAFv4 SMM



2.1.3 UAF DMM Extract



2.1.4 Example



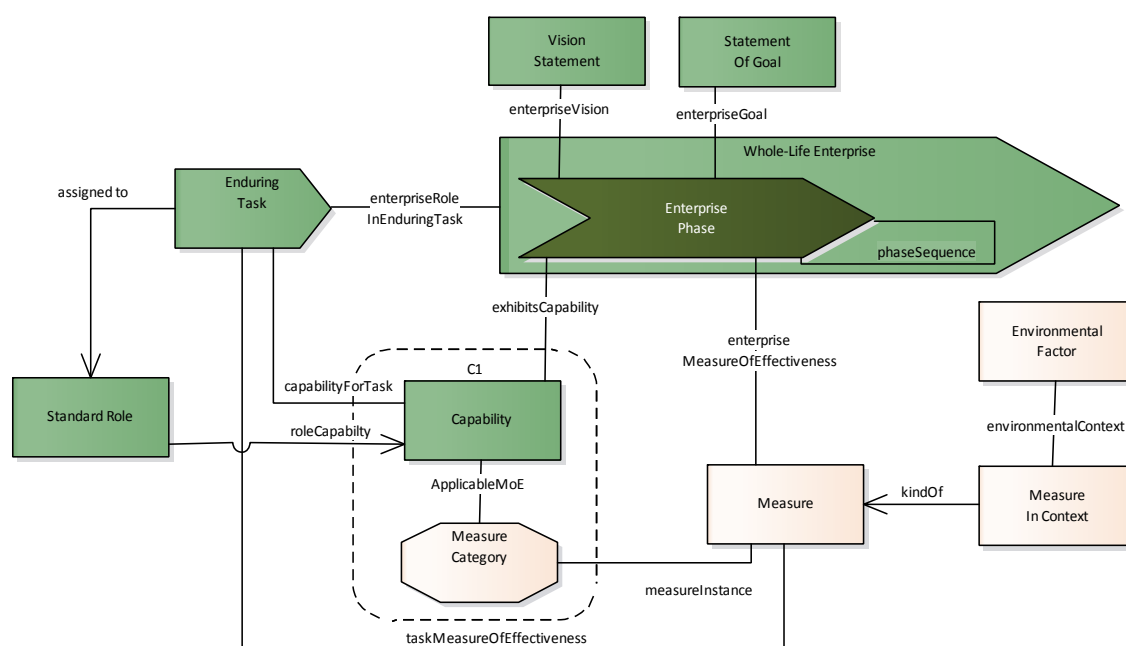
2.2 C2 – Enterprise Vision

The purpose of the C2 Viewpoint is to provide a strategic context for the capabilities described in the architecture and to specify the scope for the architecture. The C2 Viewpoint is high-level and describes the vision, goals, enduring tasks and capabilities using terminology that is easily understood by non-technical readers.

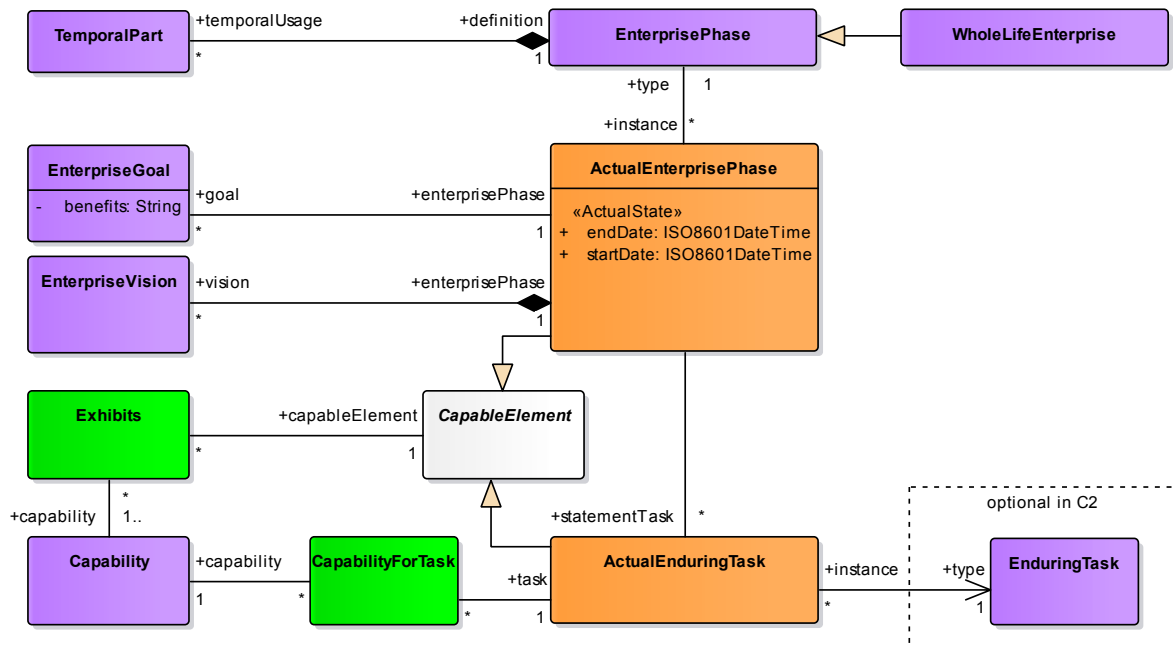
2.2.1 Concerns

- provision of the strategic context for the capabilities composed by vision, goal and period of time,
- identification of capabilities required to enable the strategic vision and goals,
- (optional) identification of desired outcomes and benefits associated with the goal, and
- (optional) linkage of capabilities to enduring tasks.

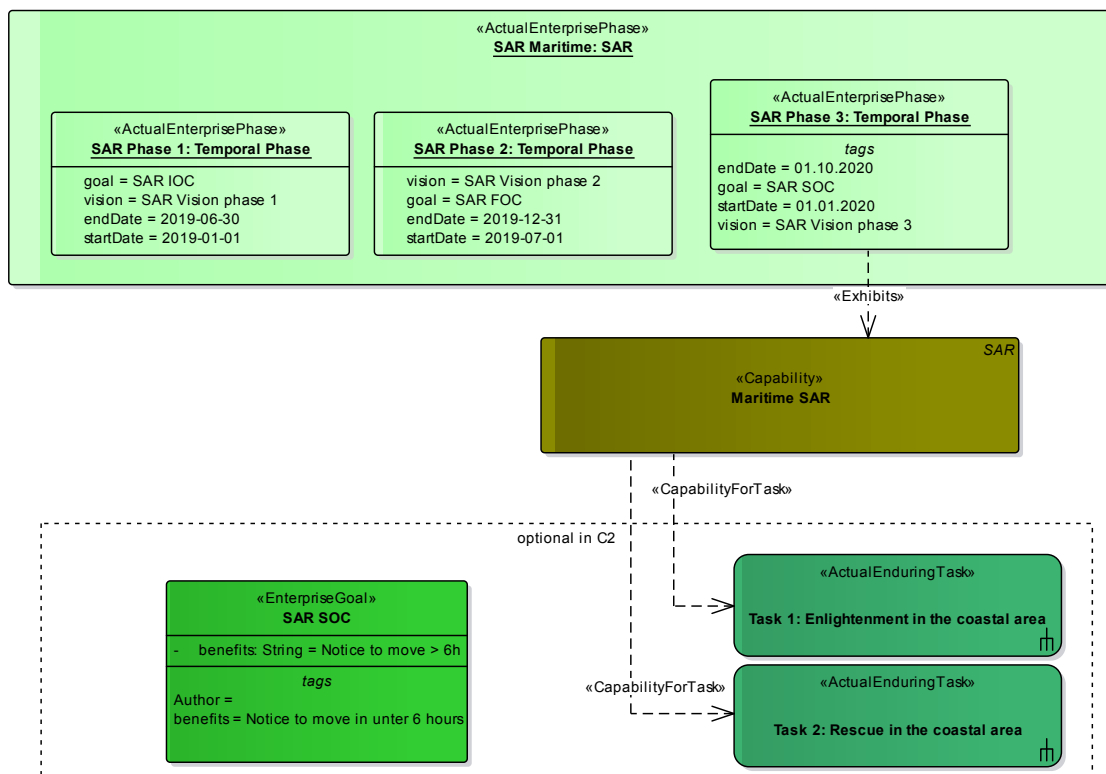
2.2.2 NAFv4 SMM



2.2.3 UAF DMM Extract



2.2.4 Example



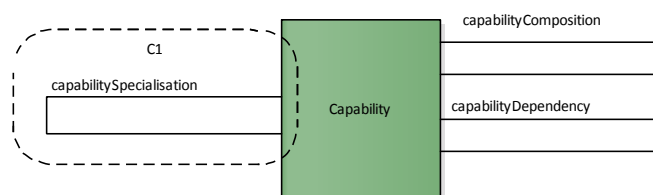
2.3 C3 – Capability Dependencies

The C3 Viewpoint is intended to provide a means of analysing the dependencies between capabilities and between capability clusters. The composition of capabilities (into clusters) is logical and the purpose of the clusters is to guide enterprise management.

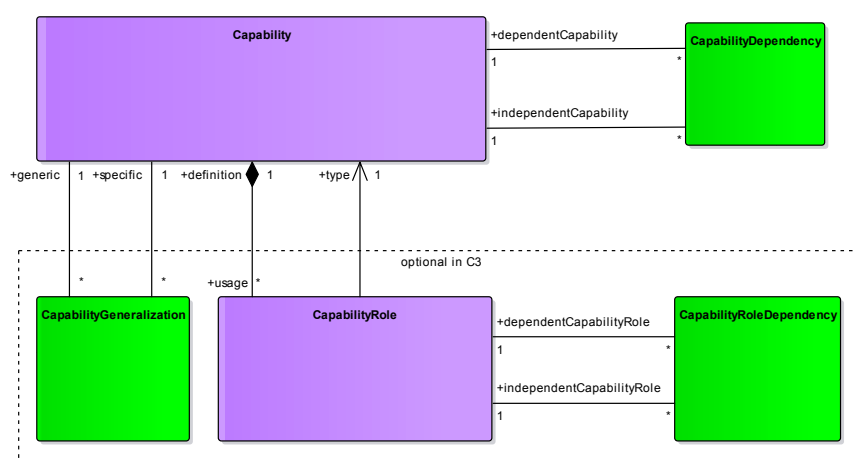
2.3.1 Concerns

- dependencies between capabilities,
- (optional) logical composition of capabilities, and
- (optional) capability specialization.

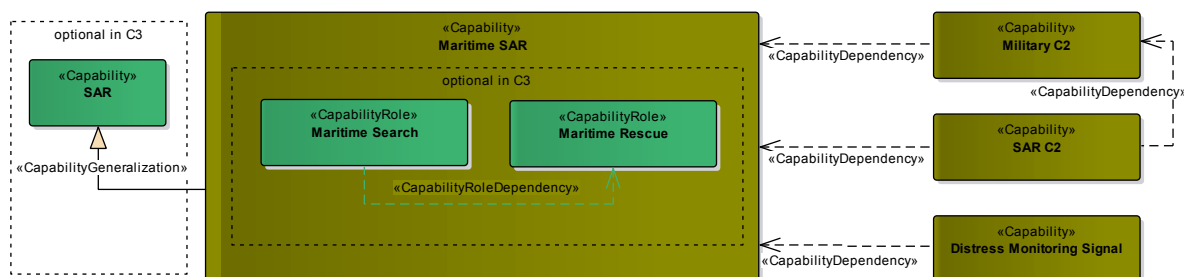
2.3.2 NAFv4 SMM



2.3.3 UAF DMM Extract



2.3.4 Example



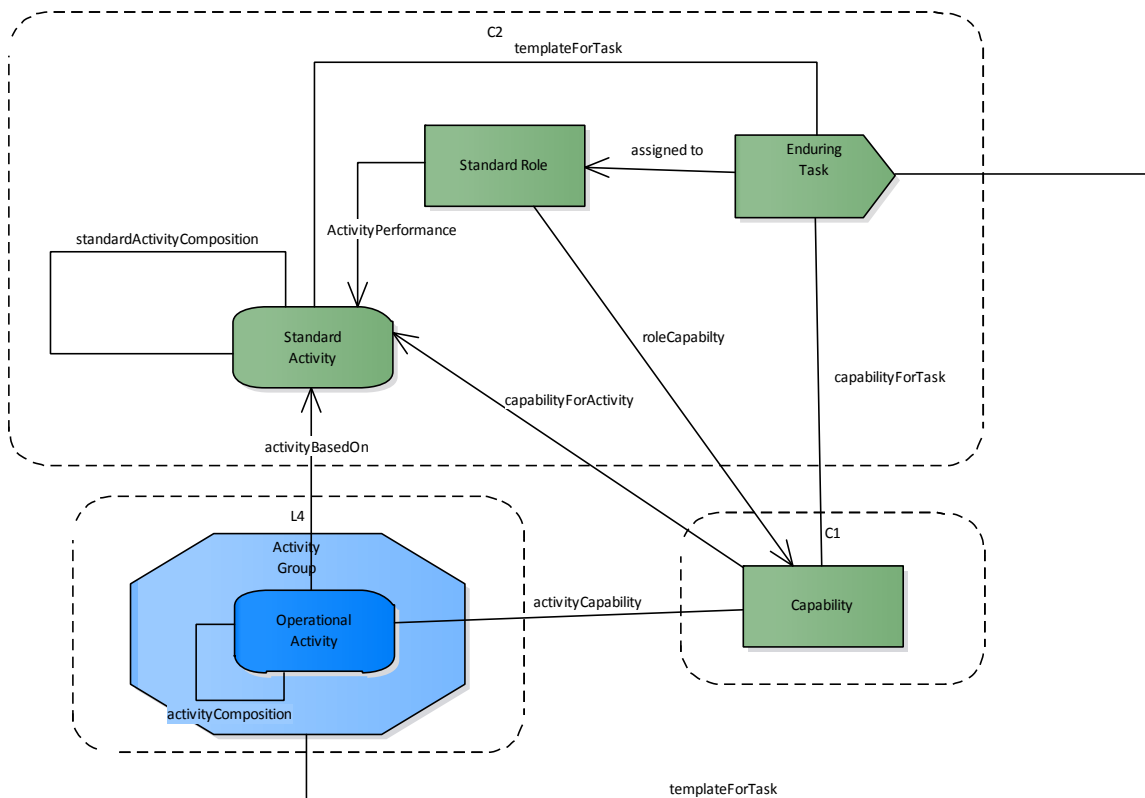
2.4 C4 – Standard Processes

Some processes are standard across the whole enterprise, or even more than one enterprise. The C4 Viewpoint specifies Standard Operational Activities that can be re-used across multiple logical architectures (e.g. in L4, Logical Activities).

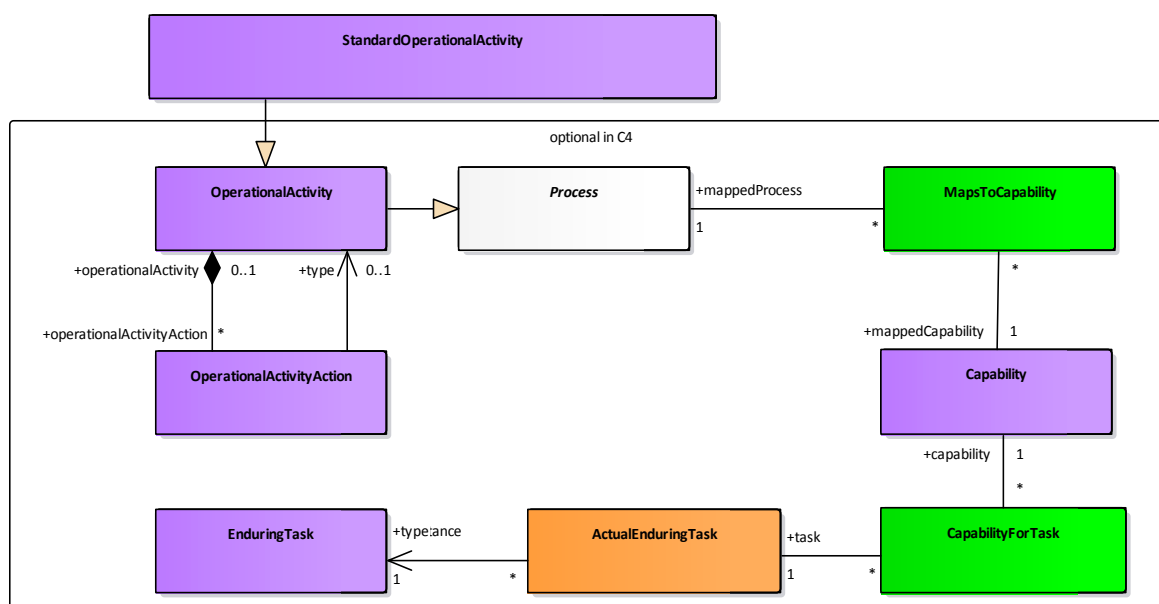
2.4.1 Concerns

- identification of standard activities,
- (optional) composition of these standard activities, and
- (optional) linkage of standard activities to supported capabilities.

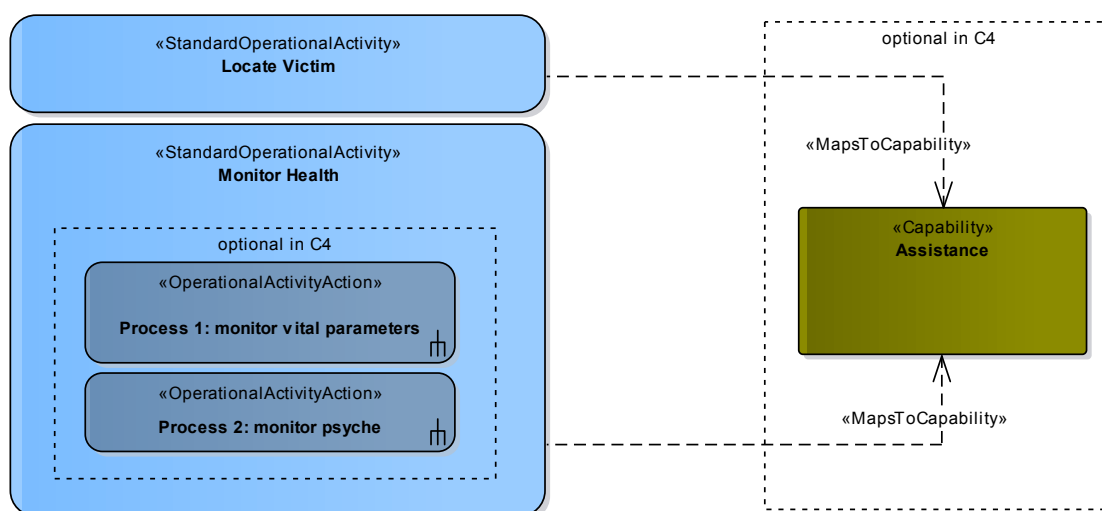
2.4.2 NAFv4 SMM



2.4.3 UAF DMM Extract



2.4.4 Example



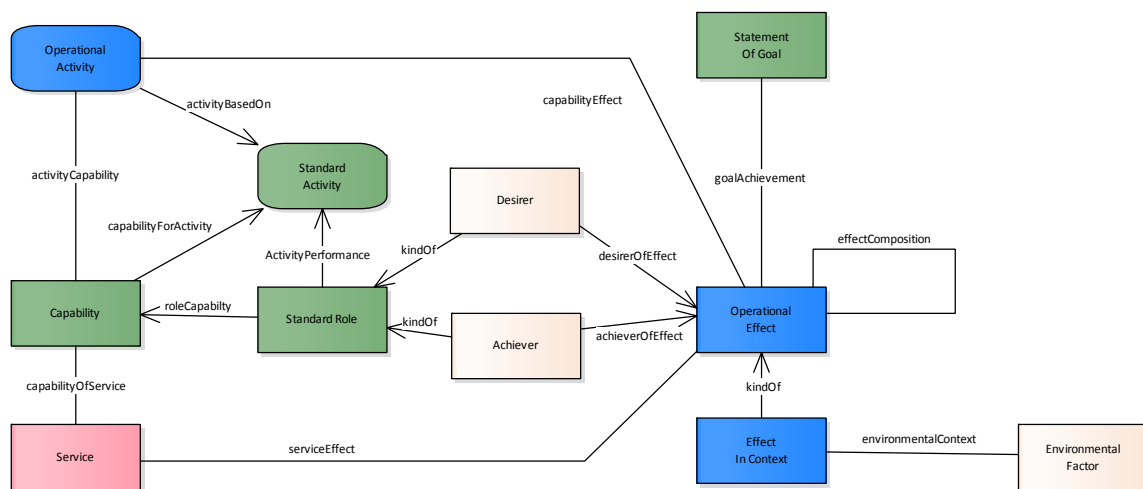
2.5 C5 – Effect

The purpose of the C5 Viewpoint is to describe the operational effect of the capabilities, activities and services according to the expected goals. The C5 Viewpoint is high-level description that should be easily understood by non-technical readers.

2.5.1 Concerns

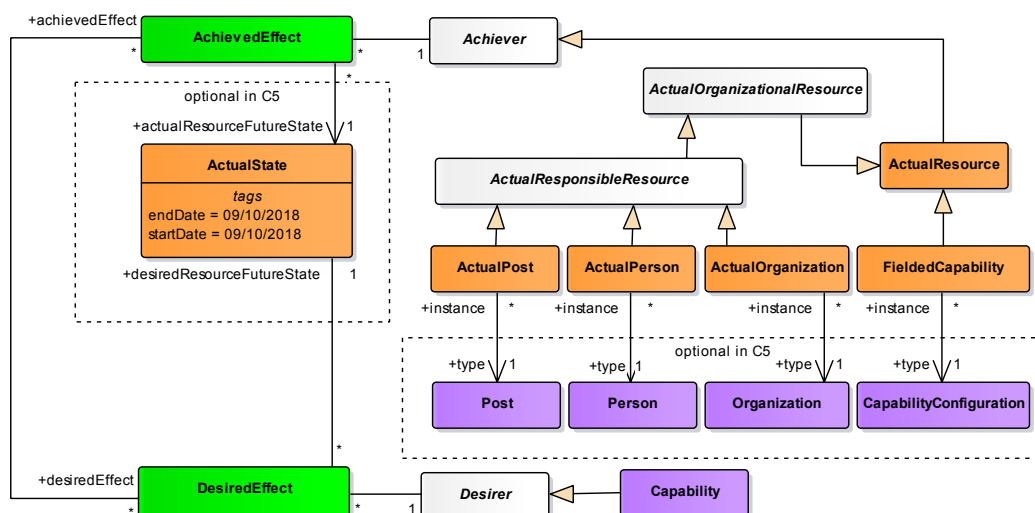
- definition of effects relevant for the architecture effort,
- assignment of effects to capabilities,
- (optional) start and/or end dates of effects,
- (optional) resource types associated to start and end dates of effect, and
- (optional) specialization hierarchy of effects.

2.5.2 NAFv4 SMM

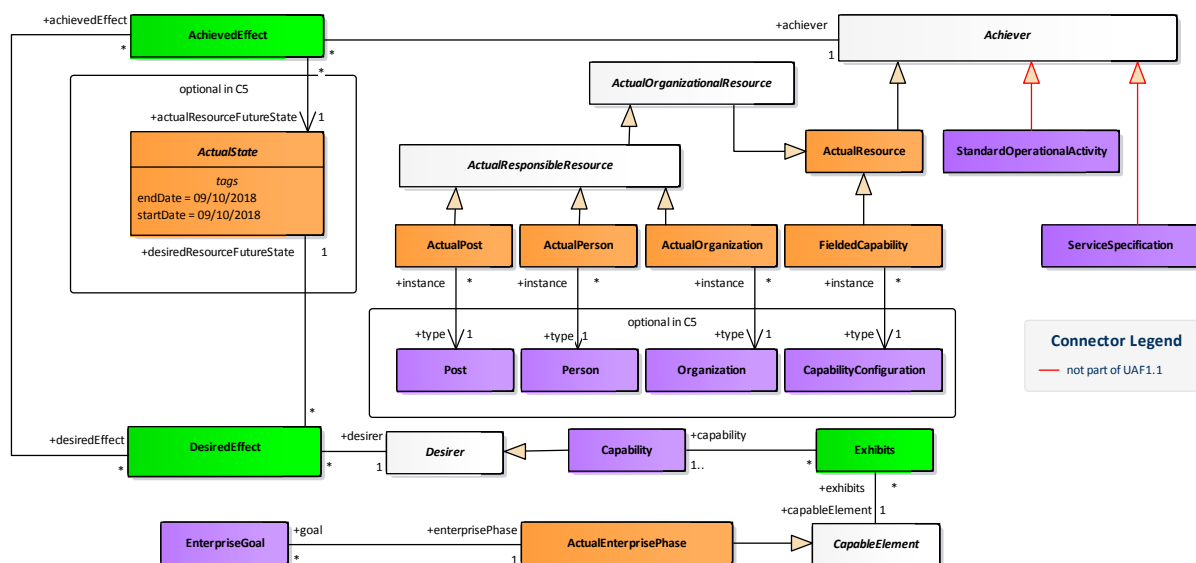


2.5.3 UAF DMM Extract

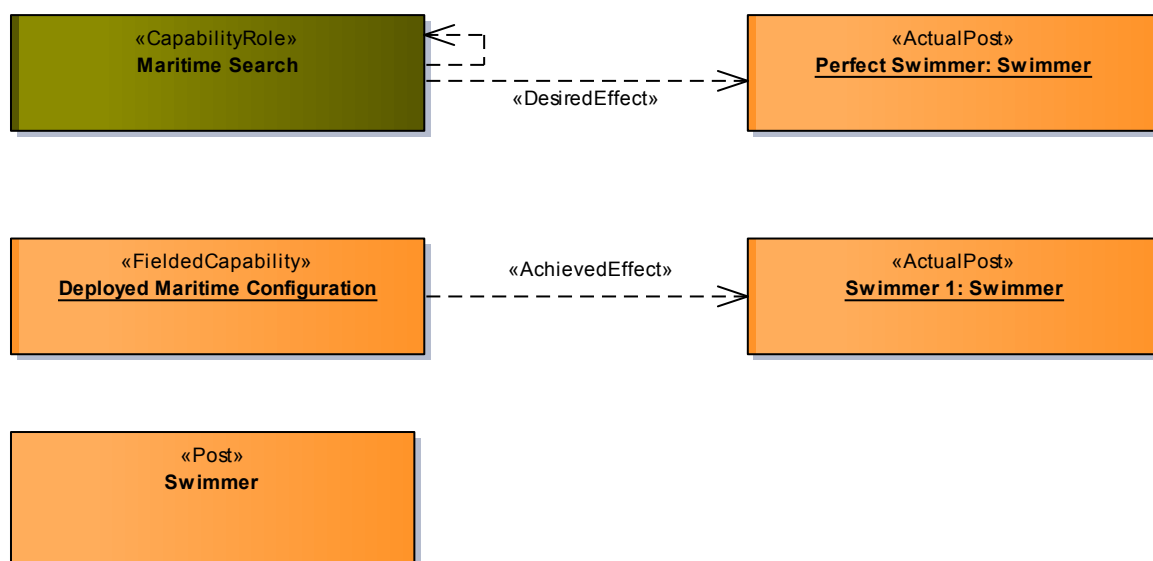
The first diagram represents the current UAF DMM extract which has been recognized by the ACoT as insufficient regarding the NAFv4 (see viewpoint description and SMM).



The second diagram offers an alternative UAF DMM extract which meets the requirements of NAFv4 but needs to be implemented by changing the UAF DMM (change request in progress).



2.5.4 Example



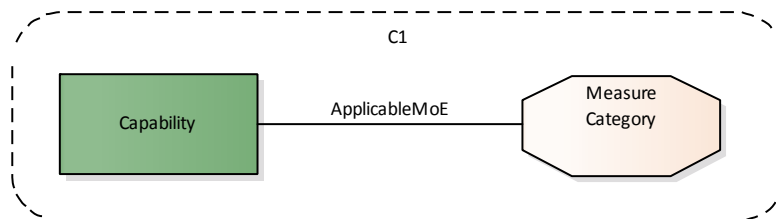
2.6 C7 - Performance Parameters

The capability requirements (and existing capabilities) are often expressed in terms of MoEs. These are high-level metrics used to judge the level of capability.

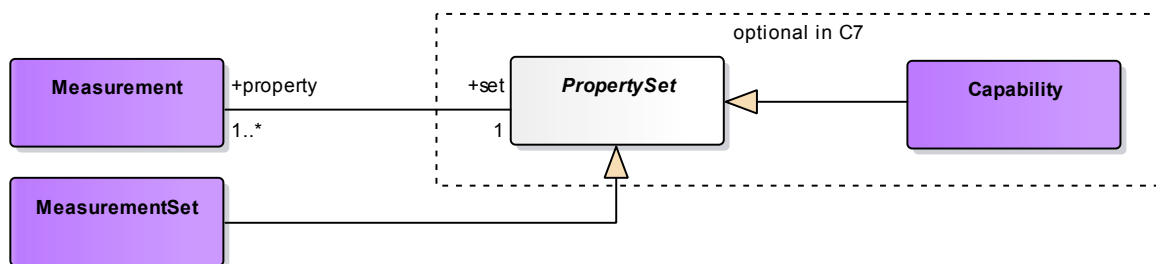
2.6.1 Concerns

- identification of all measure categories, and
- (optional) linkage of measure categories to capability.

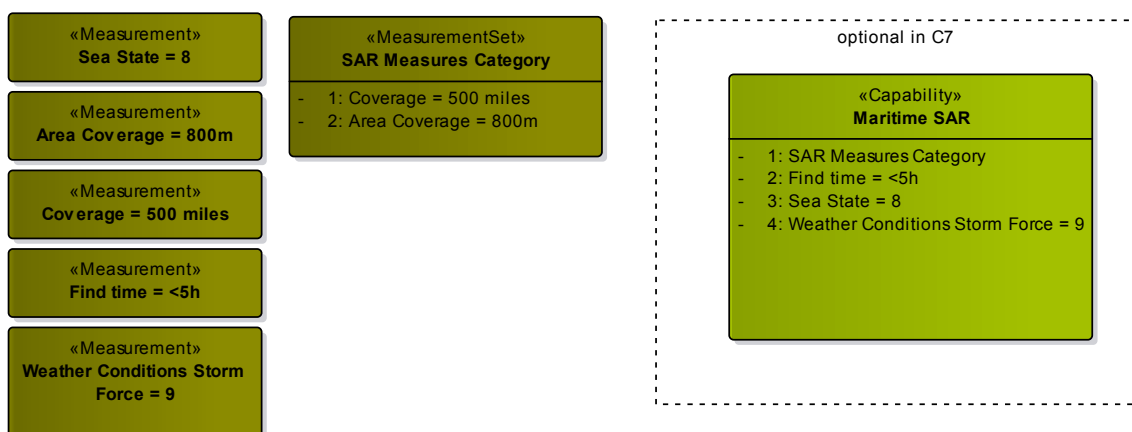
2.6.2 NAFv4 SMM



2.6.3 UAF DMM Extract



2.6.4 Example



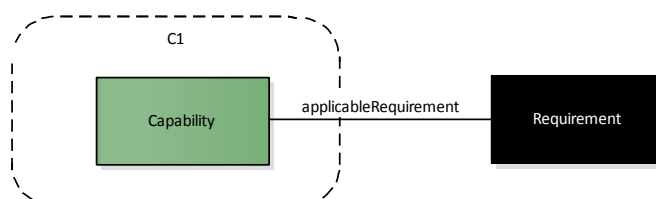
2.7 C8 - Capability Assumptions

The C8 Viewpoint is concerned with identification and description of assumptions that have been made for the implementation of capabilities. Assumptions can be expressed by means of requirements.

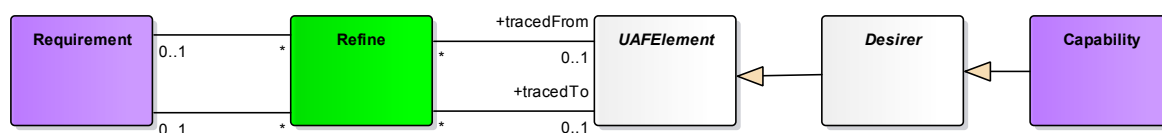
2.7.1 Concerns

- identification of assumptions for the implementation of capabilities, and
- (optional) goals and benefits associated with the assumptions.

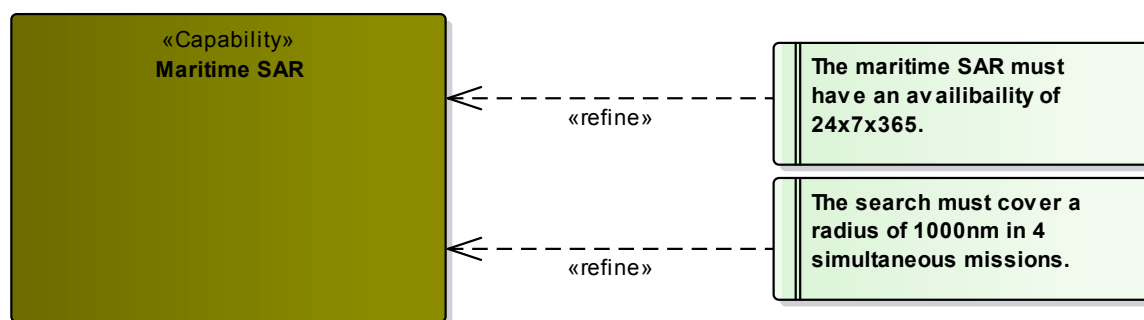
2.7.2 NAFv4 SMM



2.7.3 UAF DMM Extract



2.7.4 Example



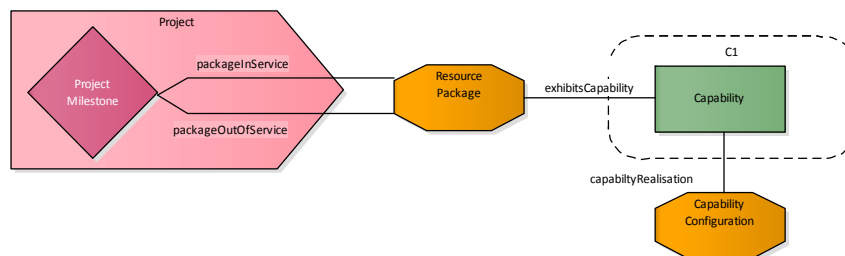
2.8 Cr – Capability Roadmap

The Cr Viewpoint supports the Capability Audit process by providing a method to identify gaps or duplications in capability provision. Cr indicates capability increments, which are derived from delivery milestones within acquisition projects.

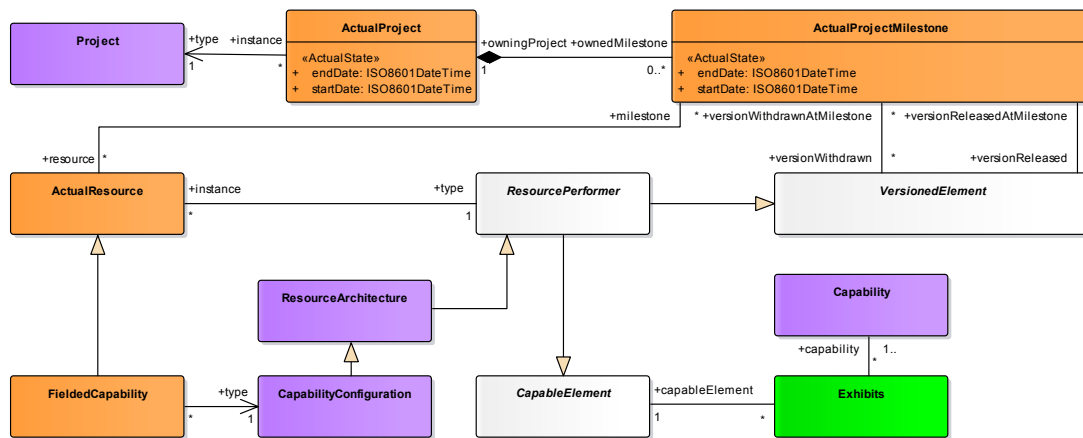
2.8.1 Concerns

- identification of the availability of capabilities and associated capability increments,
- (optional) identification of programmes or projects associated with the capability increment, and
- (optional) association of capability increments with specific periods of time.

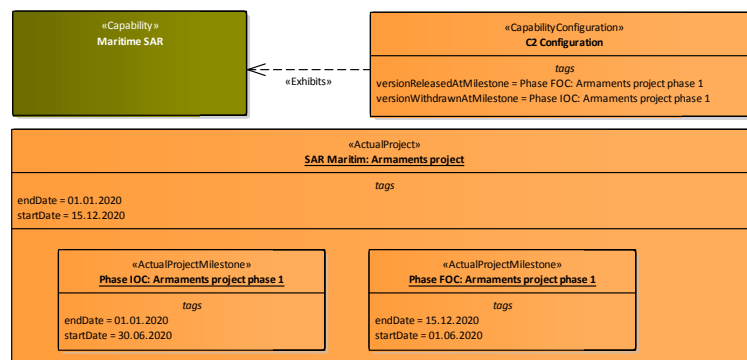
2.8.2 NAFv4 SMM



2.8.3 UAF DMM Extract



2.8.4 Example



2.9 Elements Concept View

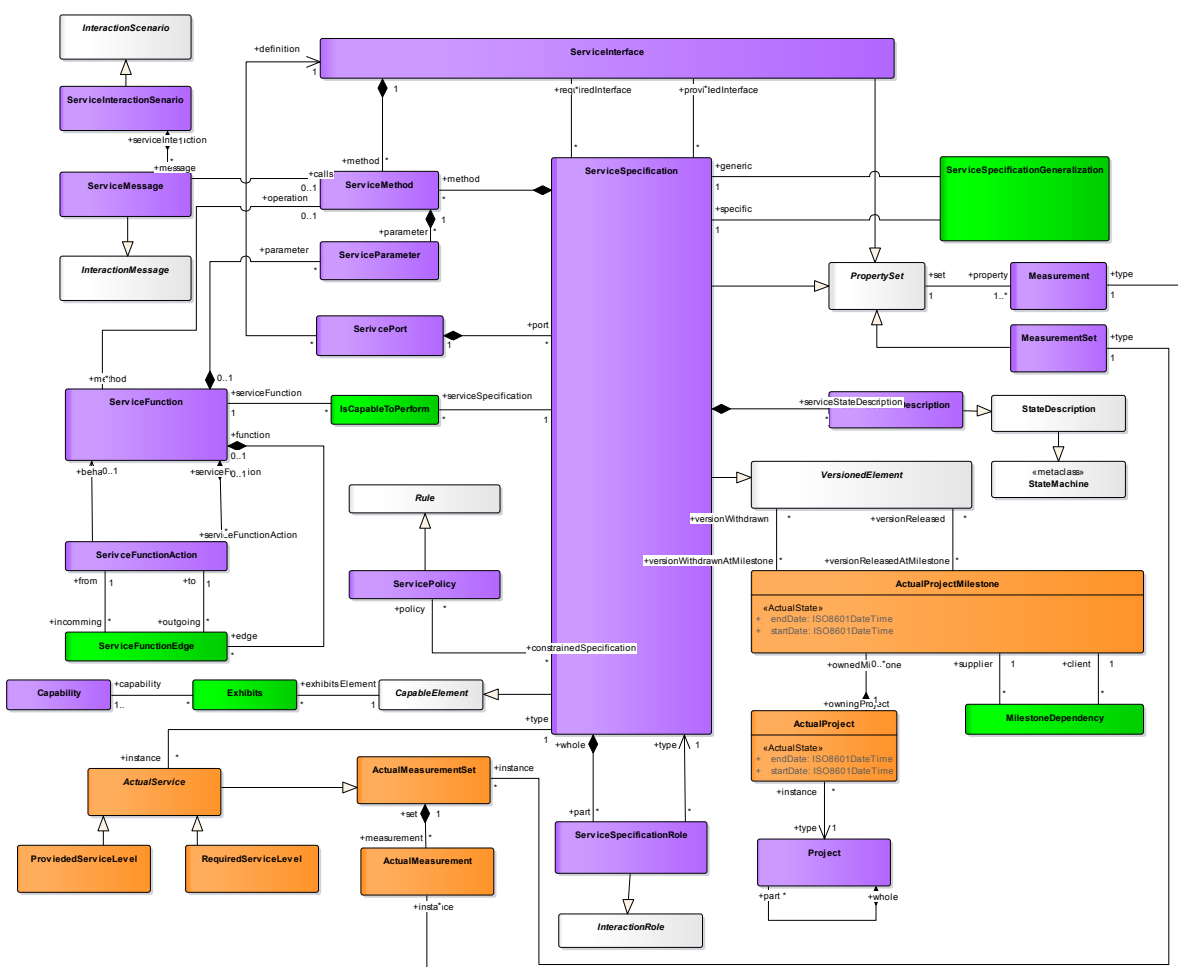
NAME	DEFINITION
AchievedEffect	A tuple that exists between an ActualState (e.g., observed/measured during testing) of an element that attempts to achieve a DesiredEffect and an Achiever.
ActualEnduringTask	An actual undertaking recognized by an enterprise as being essential to achieving its goals - i.e. a strategic specification of what the enterprise does.
ActualEnterprisePhase	The ActualState that describes the phase of an Enterprise endeavour.
ActualOrganisation	An actual formal or informal organizational unit, e.g. "Driving and Vehicle Licensing Agency", "UAF team Alpha".
ActualPerson	An individual human being.
ActualPost	An actual, specific post, an instance of a Post "type" - e.g., "President of the United States of America." where the Post would be president.
ActualProject	A time-limited endeavour to provide a specific set of ActualResource that meet specific Capability needs.
ActualProjectMilestone	An event with a start date in an ActualProject from which progress is measured.
ActualResource	Role in an Organization, where the role carries the authority to undertake a function - though the ActualOrganizationalResource given the role has the responsibility.
ActualState	Abstract element that applies temporal extent to a set of elements realized as Instance Specifications.
Capability	A high-level specification of the enterprise's ability to execute a specified course of action.
CapabilitiyConfiguration	A composite structure representing the physical and human resources (and their interactions) in an enterprise, assembled to meet a capability).
CapabilityDependency	No definition.
CapabilityForTask	A tuple that asserts that a Capability is required in order for an Enterprise to conduct a phase of an EnduringTask.
CapabilityGeneralization	No definition
CapabilityRole	A high-level specification of the enterprise's ability to execute a specified course of action.
CapabilityRoleDependency	No definition.
DesiredEffect	A tuple relating the Desirer (a Capability or OrganizationalResource) to an ActualState.
EnduringTask	A type of template behaviour recognized by an enterprise as being essential to achieving its goals - i.e. a template for a strategic specification of what the enterprise does.
EnterpriseGoal	A statement about a state or condition of the enterprise to be brought about or sustained through appropriate Means. An EnterpriseGoal amplifies an EnterpriseVision that is, it indicates what must be satisfied on a continuing basis to effectively attain the EnterpriseVision.
EnterprisePhase	A current or future state of the wholeLifeEnterprise or another EnterprisePhase.
EnterpriseVision	A Vision describes the future state of the enterprise, without regard to how it is to be achieved.
Exhibits	A tuple that exists between a CapableElement and a Capability that it meets under specific environmental conditions.
FieldedCapability	An individual, fully-realized capability.
MapsToCapability	A tuple denoting that an Activity contributes to providing a Capability.
Measurement	A property of an element representing something in the physical world, expressed in amounts of a unit of measure.
MeasurementSet	A collection of Measurements.
OperationalActivity	An Activity that captures a logical process, specified independently of how the process is carried out.
OperationalActivityAction	A call of an OperationalActivity in the context of another OperationalActivity.
Project	A9 type that describes types of time-limited endeavours that are required to meet one or more Capability needs.
Refine	No definition.
Requirement	SysML: A requirement specifies a capability or condition that must (or should) be satisfied. A requirement may specify a function that a system must perform or a performance condition that a system must satisfy. Requirements are used to establish a contract between the customer (or other stakeholder) and those responsible for designing and implementing the system. A requirement is a stereotype of Class. Compound requirements can be created by using the nesting capability of the class definition mechanism. The default interpretation of a compound requirement, unless

	stated differently by the compound requirement itself, is that all its subrequirements must be satisfied for the compound requirement to be satisfied. Subrequirements can be accessed through the “nestedClassifier” property of a class. When a requirement has nested requirements, all the nested requirements apply as part of the container requirement. Deleting the container requirement deleted the nested requirements, a functionality inherited from UML.
ResourceArchitecture	A type used to denote a model of the Architecture, described from the ResourcePerformer perspective.
StandardOperationalActivity	A sub-type of OperationalActivity that is a standard operating procedure.
StructuralPart	No description.
TemporalPart	A current or future state of the wholeLifeEnterprise or another EnterprisePhase.
WholeLifeEnterprise	A WholeLifeEnterprise is a purposeful endeavour of any size involving people, organizations and supporting systems. It is made up of TemporalParts and StructuralParts.

3 SERVICE SPECIFICATION VIEWPOINTS



The Viewpoints in the Service Specifications row of the NAF grid support the description of services independently of how they are implemented or used. A service is understood in its broadest sense as a unit of work through which a provider provides a useful result to a consumer.



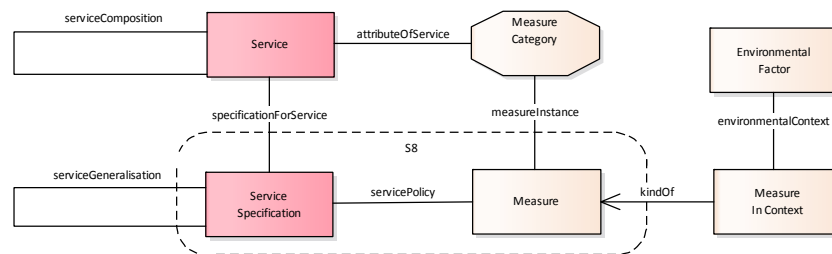
3.1 S1 - Service Taxonomy

The purpose of the S1 Viewpoint is to provide a governance structure for a Service-Oriented Architecture. Along with S3, Service Interfaces, it specifies a standard library of service specifications for an enterprise, to which service implementers are expected to conform.

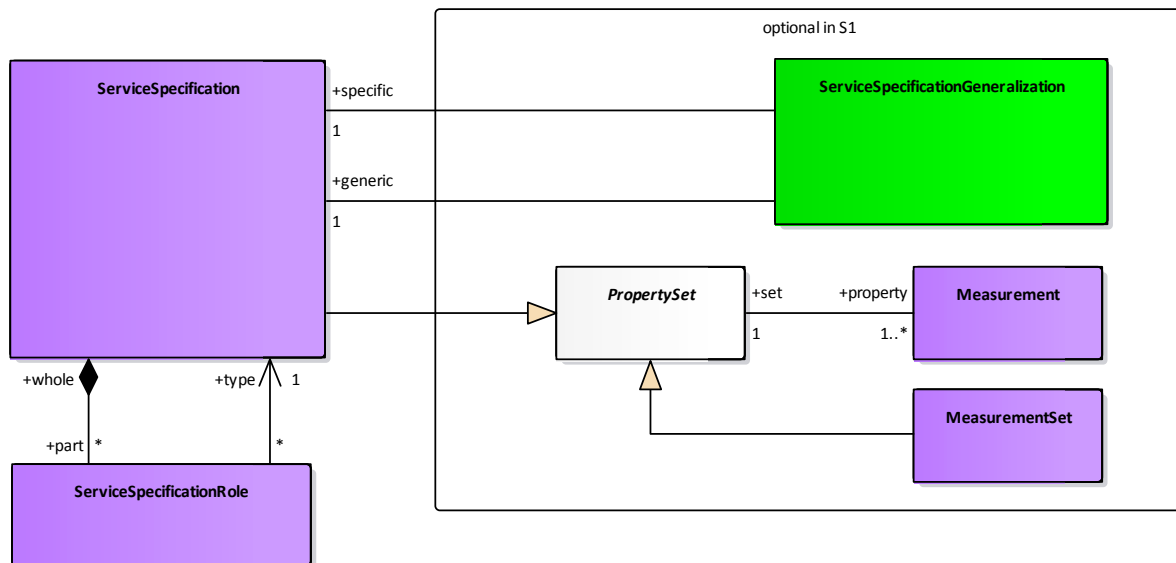
3.1.1 Concerns

- identification of service specifications,
- (optional) organization of service specifications into specialization hierarchies (taxonomies),
- (optional) measures for the service specifications, and
- (optional) attributes for the service specifications.

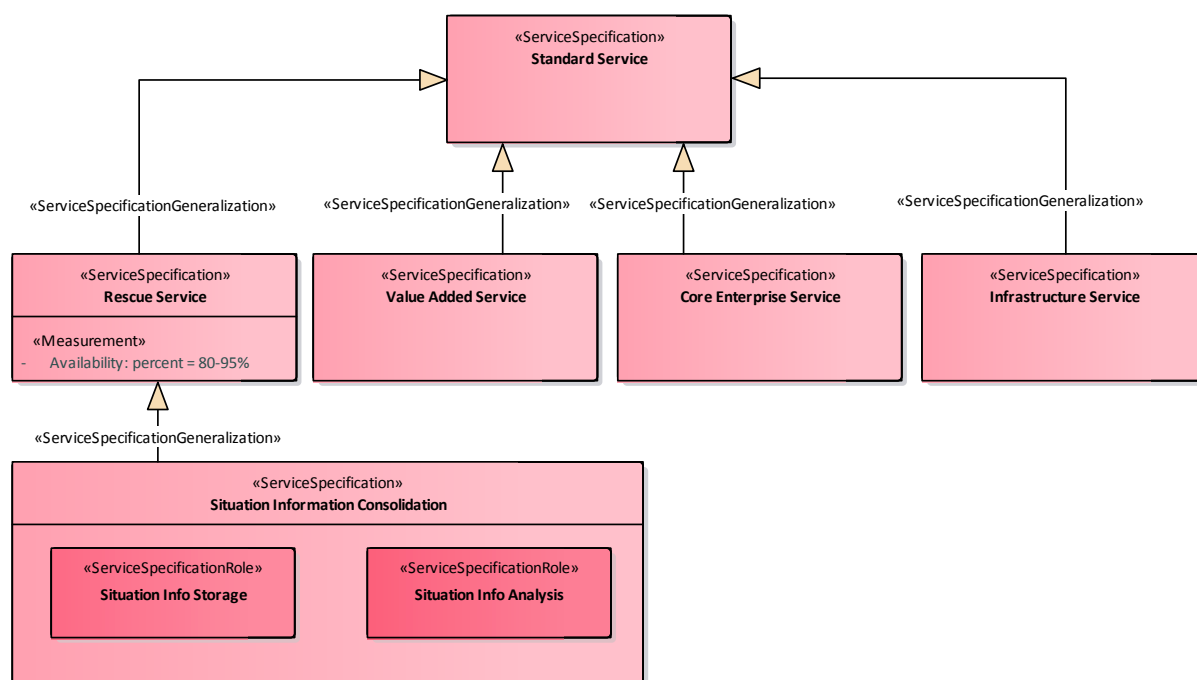
3.1.2 NAFv4 SMM



3.1.3 UAF DMM Extract



3.1.4 Example



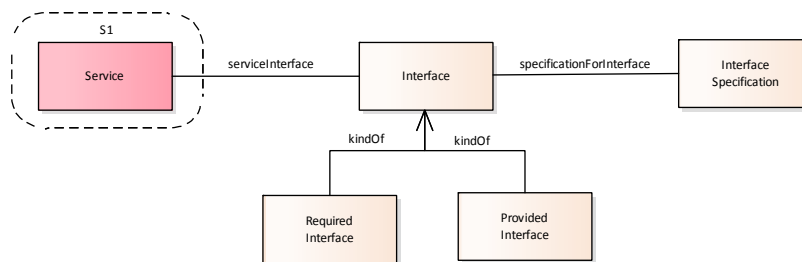
3.2 S3 – Service Interfaces

The S3 Viewpoint is concerned with the identification and definition of service interfaces.

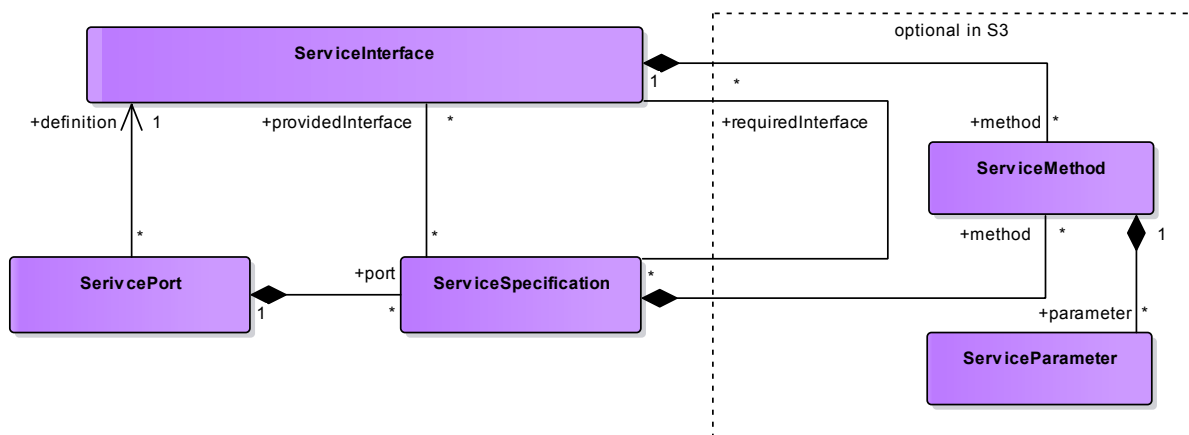
3.2.1 Concerns

- identification and definition of service interfaces,
- identification of provided service interfaces,
- (optional) identification of required service interfaces,
- (optional) identification of operations for service interfaces, and
- (optional) specification of parameters for service interface operations.

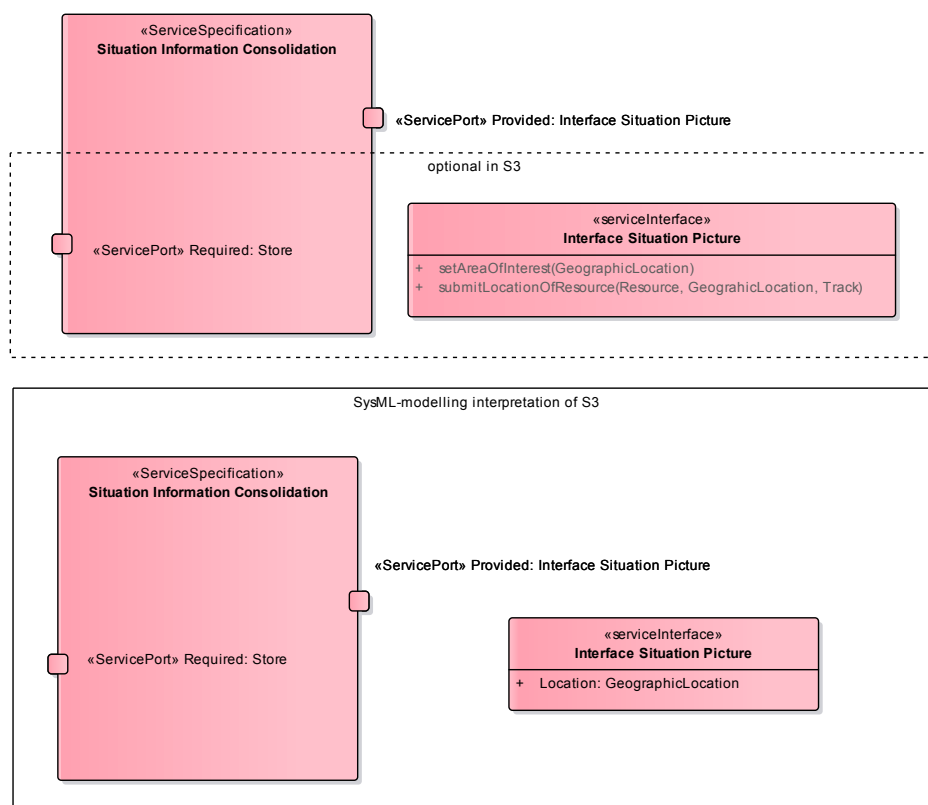
3.2.2 NAFv4 SMM



3.2.3 UAF DMM Extract



3.2.4 Example



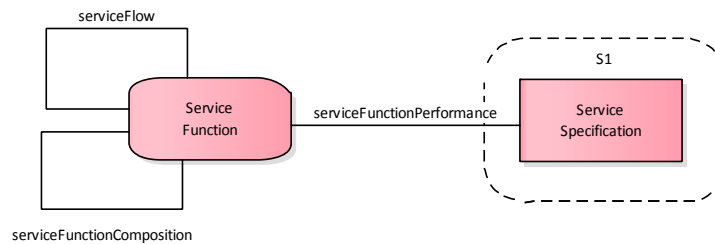
3.3 S4 – Service Functions

The S4 Viewpoint is the key behavioural specification for services. Equivalent in nature to L4, Logical Activities and P4, Resources Functions, it specifies a set of functions that a service implementation is expected to perform. Implementation of that behaviour is represented in P4, Resource Functions, and L4-P4, Activity to Function Mapping.

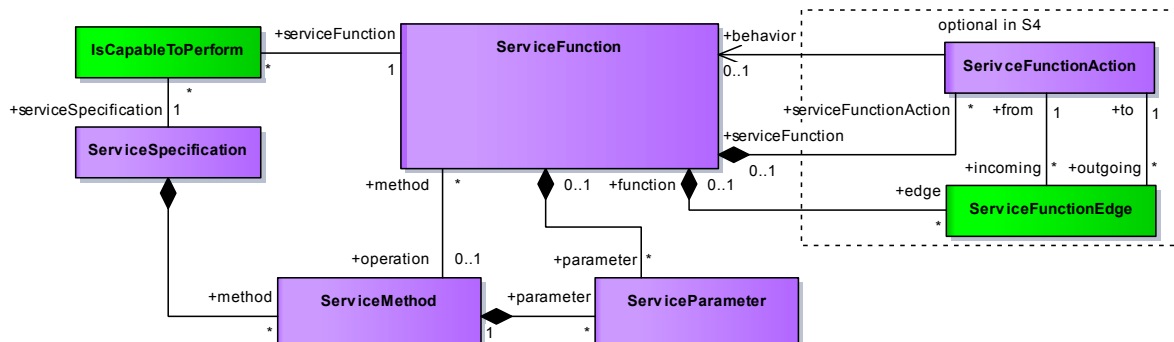
3.3.1 Concerns

- identification of all functions a service is performing, and
- (optional) specification of composition of service functions.

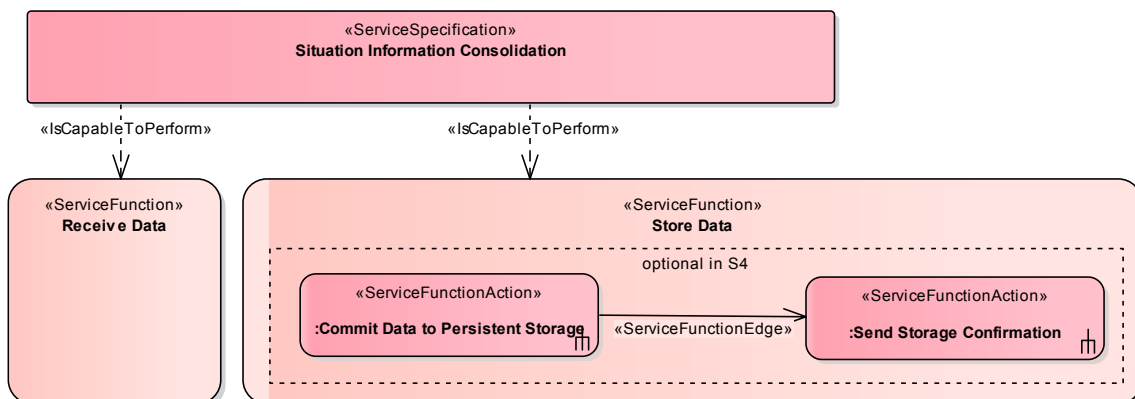
3.3.2 NAFv4 SMM



3.3.3 UAF DMM Extract



3.3.4 Example



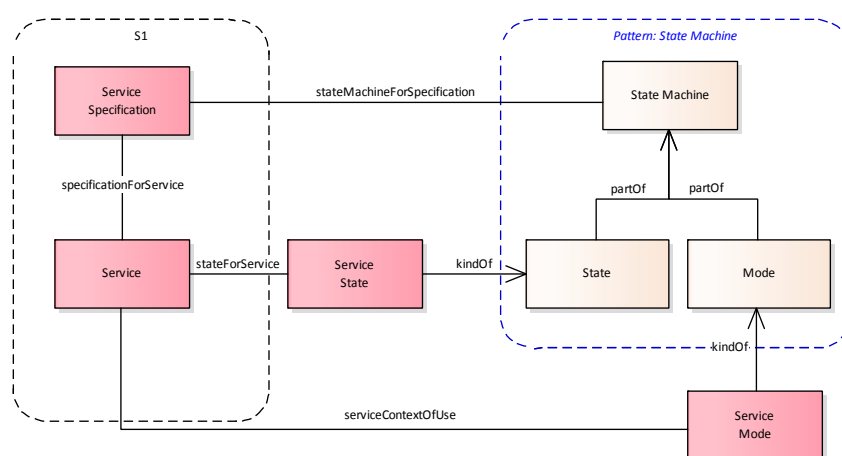
3.4 S5 – Service States

It is generally considered good practice to make services stateless – i.e. consumers of a service are not aware of what state the service is in. However, in specifying a service, it is often necessary to specify the *allowable* states so as to constrain how implementations of the service will behave. The S5 Viewpoint is a specification of those states, and the possible transitions between them.

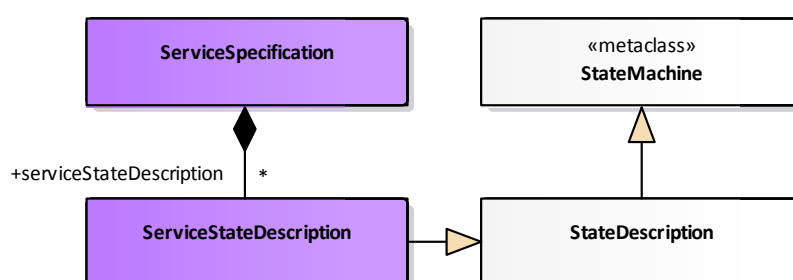
3.4.1 Concerns

- identification and definition of all allowable states of a service,
- (optional) description of possible state transitions, and
- (optional) description of performance constraints.

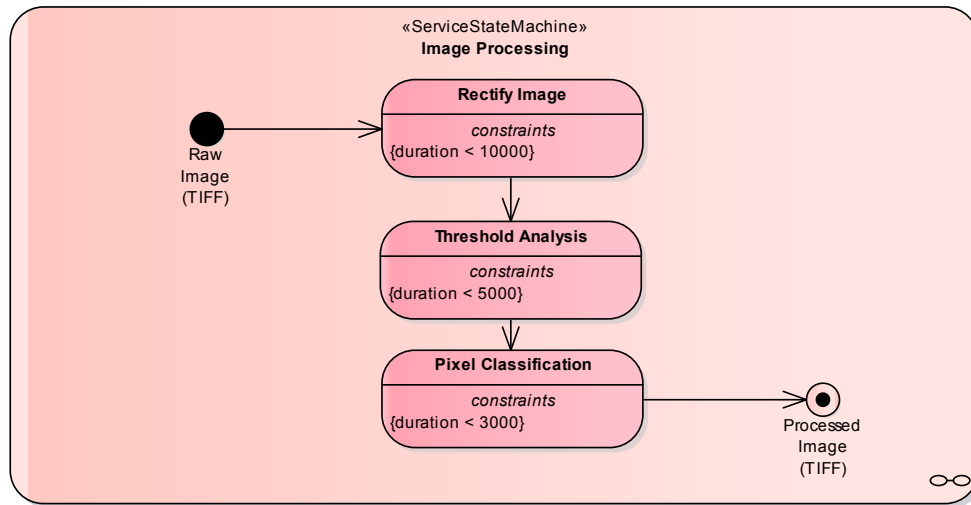
3.4.2 NAFv4 SMM



3.4.3 UAF DMM Extract



3.4.4 Example



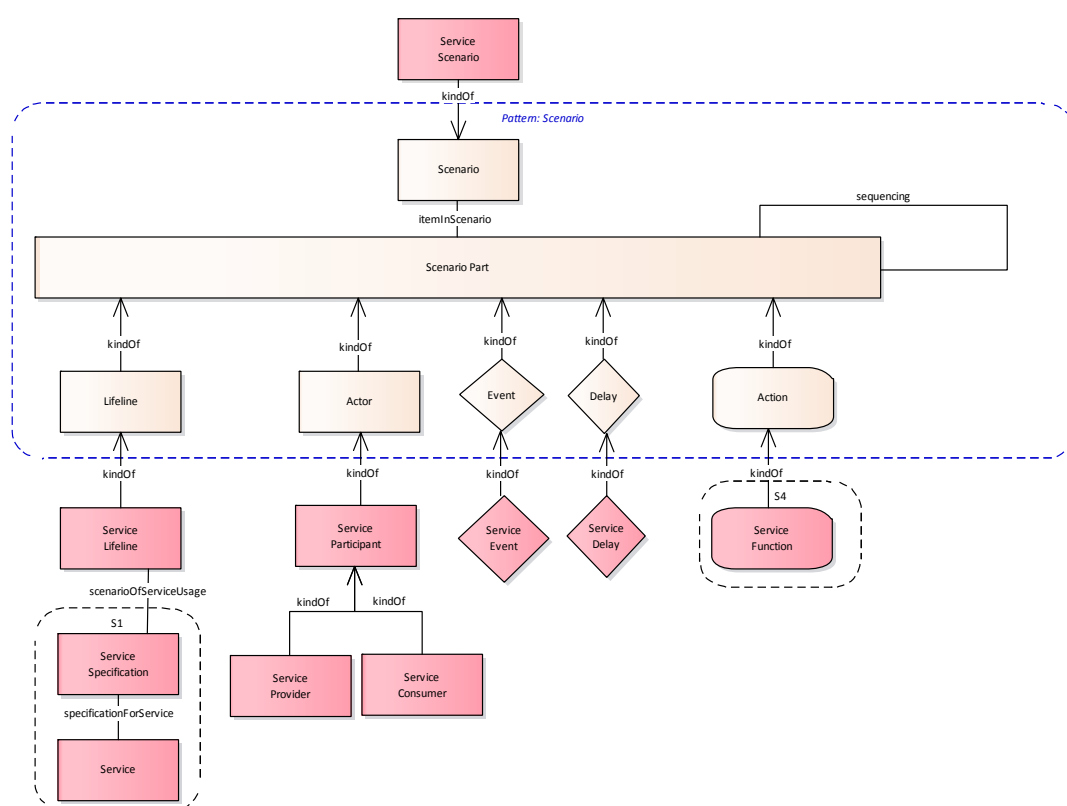
3.5 S6 – Service Interactions

Service Interaction Descriptions, sometimes called sequence diagrams, event scenarios or timing diagrams, allow the tracing of interactions between services in a composition or critical sequence of events.

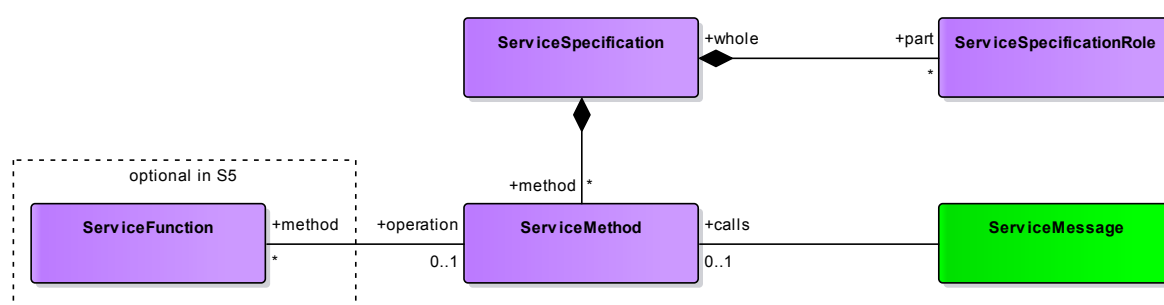
3.5.1 Concerns

- description of interactions of a service with service consumers, and the sequence and dependencies of those interactions,
- (optional) service operations and sequence of service operations, and
- (optional) service functions and sequence of service function.

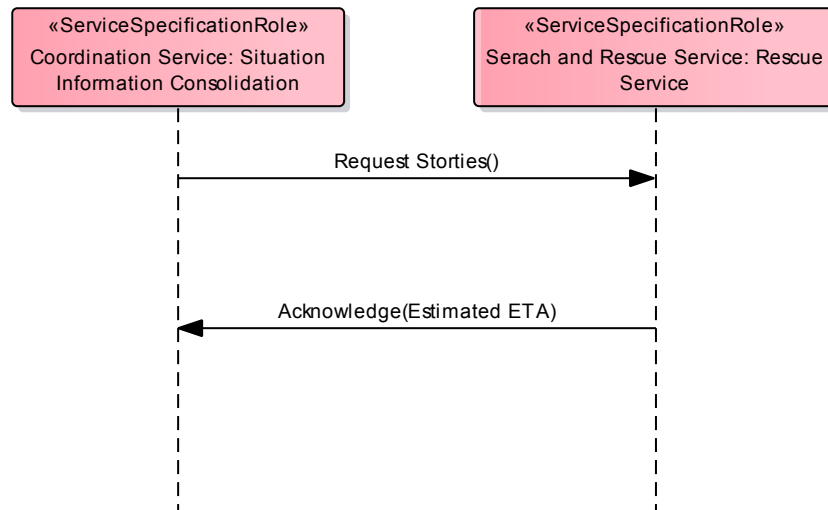
3.5.2 NAFv4 SMM



3.5.3 UAF DMM Extract



3.5.4 Example



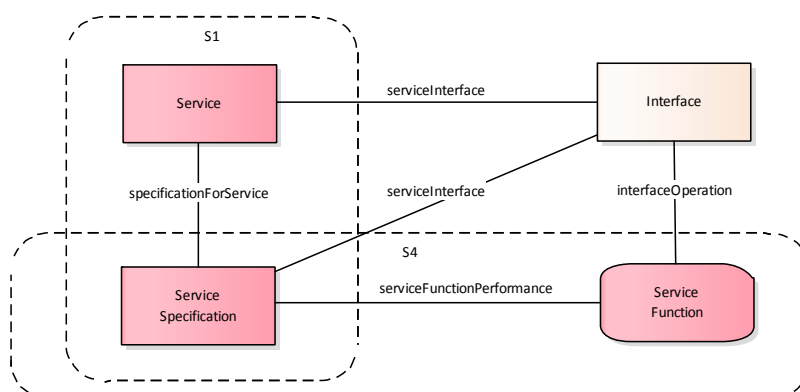
3.6 S7 – Service Interface Parameters

Specify the interfaces that a service provides and uses, defines which services are compatible with which other services. If Service A provides an interface, X, and Service B requires Interface X, then Service B can call upon at least some of the functionality of Service A.

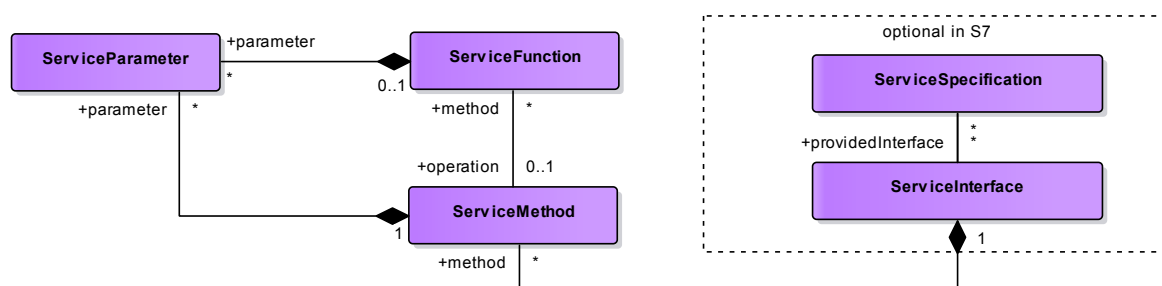
3.6.1 Concerns

- identification of parameters for service operations,
- (optional) specification of the data types of each parameter, and
- (optional) assignment of service operation to service interface.

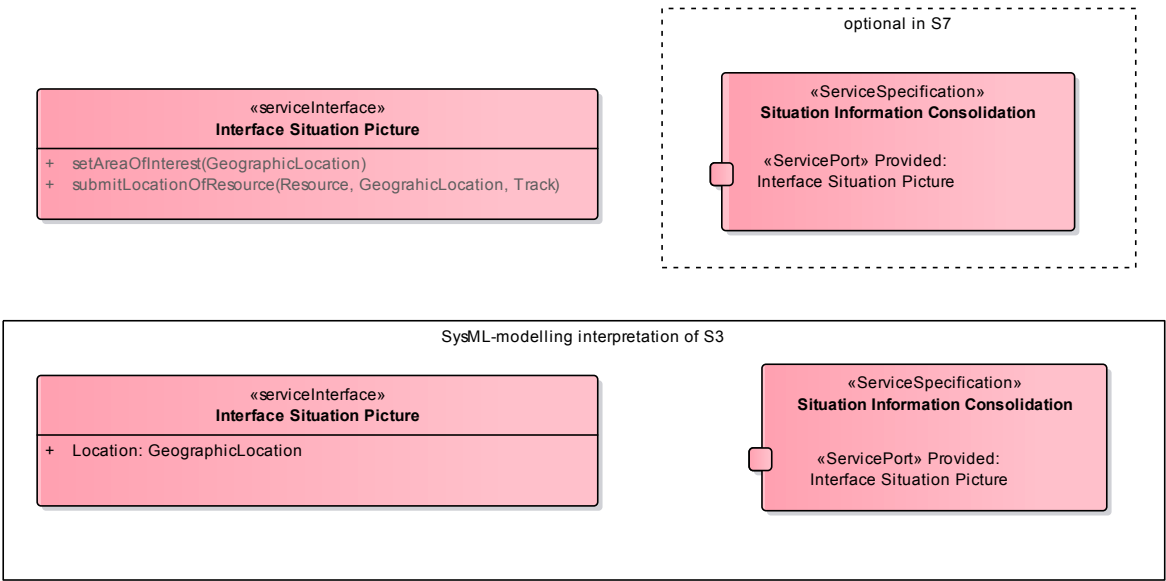
3.6.2 NAFv4 SMM



3.6.3 UAF DMM Extract



3.6.4 Example



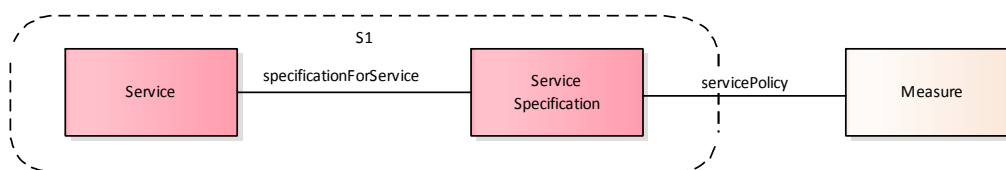
3.7 S8 – Service Policy

To better enable consistency and re-use of service specifications, it is important to set constraints on how a service should behave. The S8 Viewpoint specifies constraints against services to which implementations must conform.

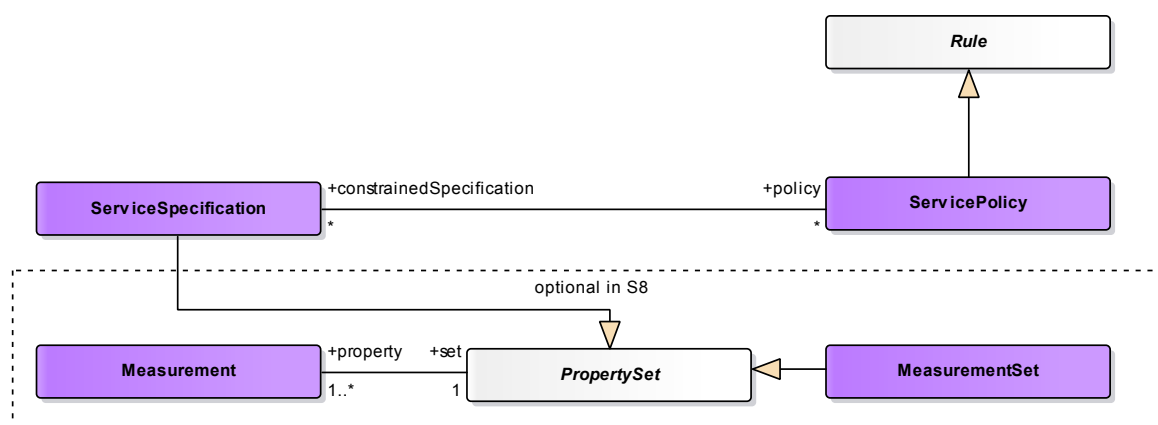
3.7.1 Concerns

- identification and description of constraints that apply to service implementation,
- (optional) inclusion of measures for the service specification, and
- (optional) inclusion of attributes for the service specification.

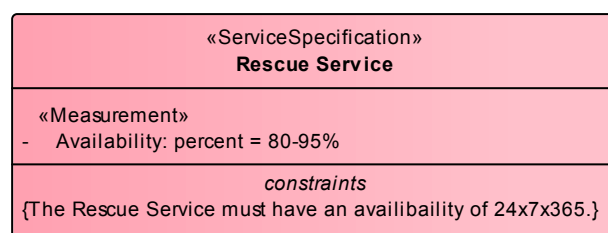
3.7.2 NAFv4 SMM



3.7.3 UAF DMM Extract



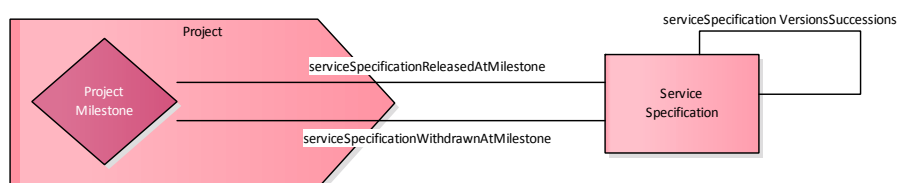
3.7.4 Example



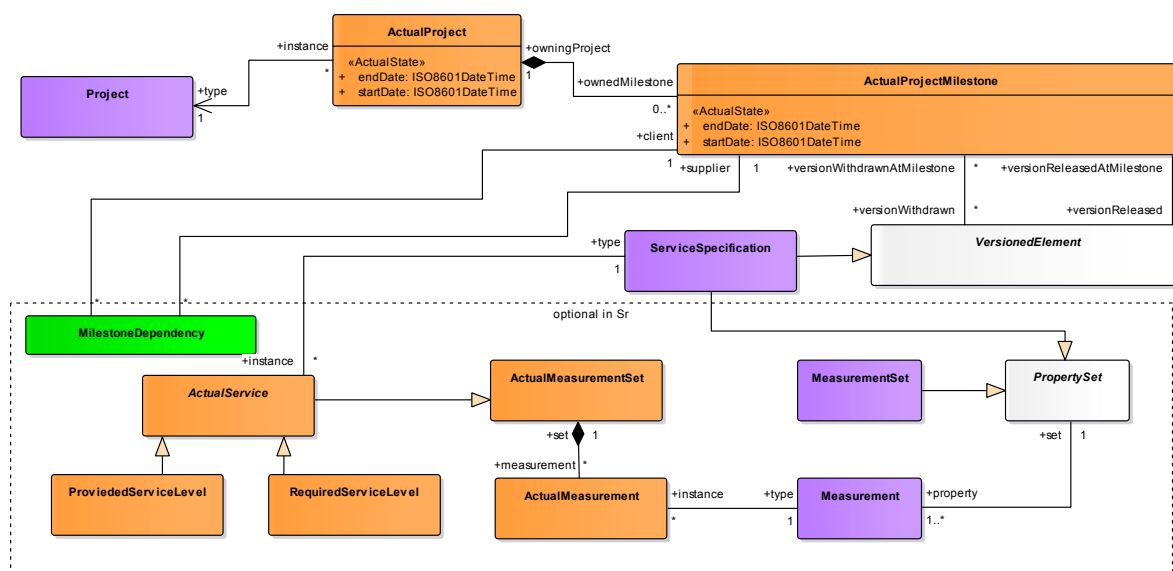
3.8.1 Concerns

- identification of service specifications related to the roadmap,
- definition of start and end dates of service specification support,
- (optional) identification of programmes or projects associated with the service specification delivery/withdrawal,
- (optional) identification of service levels,
- (optional) identification of service attributes, and
- (optional) identification of measures to service attributes.

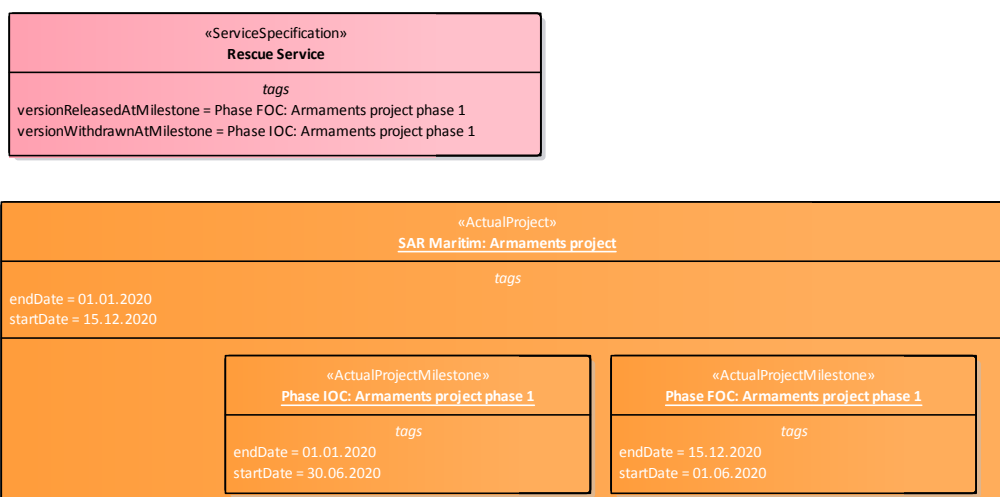
3.8.2 NAFv4 SMM



3.8.3 UAF DMM Extract



3.8.4 Example



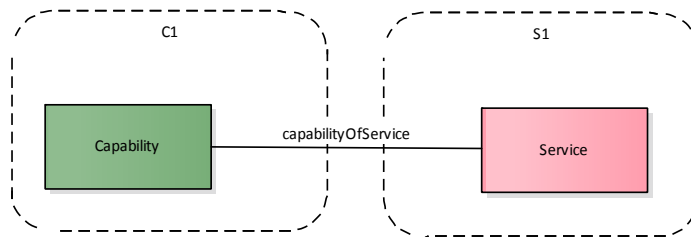
3.9 C1 - S1 – Capability to Service Mapping

If network enabled capability is to be delivered by the orchestration of loosely couple services (i.e. a service-oriented architecture), it is important to know which services have the potential to support particular capabilities. A C1-S1 Viewpoint presents a simple mapping of services to capabilities, showing which services contribute to which capability.

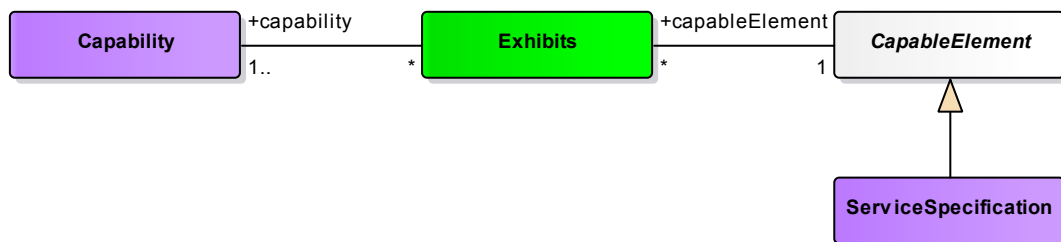
3.9.1 Concerns

- association of service specifications to capabilities they enable.

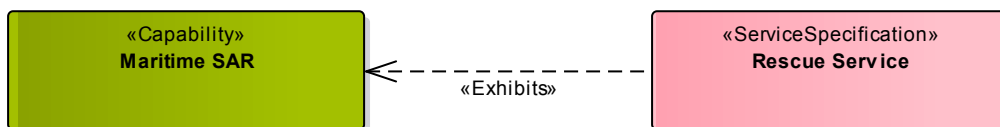
3.9.2 NAFv4 SMM



3.9.3 UAF DMM Extract



3.9.4 Example



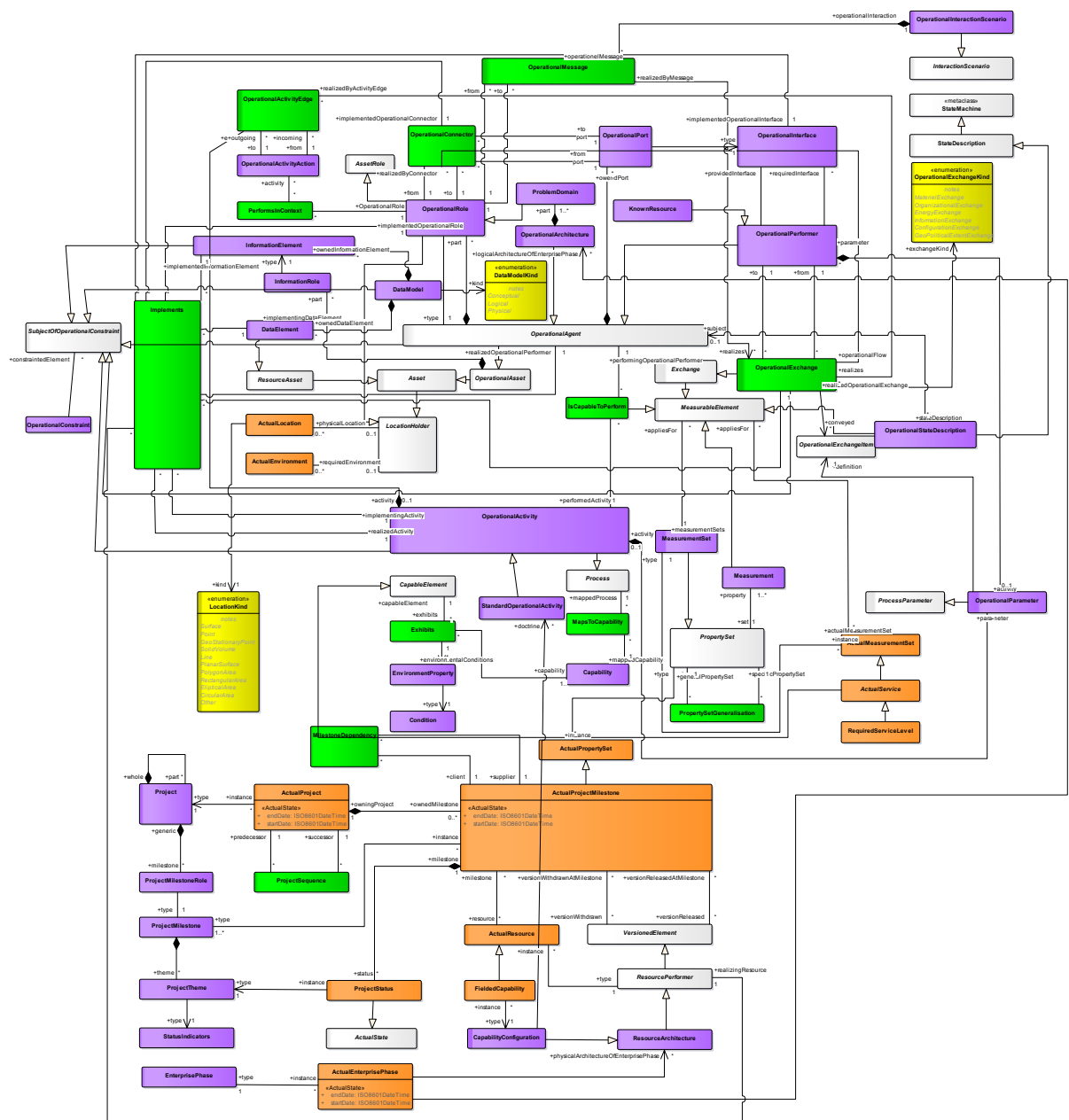
3.10 Elements

NAME	DEFINITION
ActualMeasurement	An actual value that is applied to a Measurement.
ActualMeasurementSet	A set of ActualMeasurements.
ActualProject	A time-limited endeavour to provide a specific set of ActualResource that meet specific Capability needs.
ActualProjectMilestone	An event with a start date in an ActualProject from which progress is measured.
Capability	A high-level specification of the enterprise's ability to execute a specified course of action.
Exhibits	A tuple that exists between a CapableElement and a Capability that it meets under specific environmental conditions.
IsCapableToPerform	A tuple defining the traceability between the CapableElements to the Activities that they can perform.
Measurement	A property of an element representing something in the physical world, expressed in amounts of a unit of measure.
MeasurementSet	A collection of Measurements.
PropertySet	An abstract type grouping architectural elements that can own Measurements.
ProvidedServiceLevel	A sub type of ActualService that details a specific service level delivered by the provider.
RequiredServiceLevel	A sub type of ActualService that details a specific service level required of the provider.
ServiceFunction	An Activity that describes the abstract behaviour of ServiceSpecifications, regardless of the actual implementation.
ServiceFunctionAction	A behavioural feature of a ServiceSpecification whose behaviour is specified in a ServiceFunction.
ServiceFunctionEdge	A tuple that shows the flow of Resources (objects/information) between OperationalActivityActions.
ServiceInterface	A contract that defines the ServiceMethods and ServiceMessageHandlers that the ServiceSpecification realizes.
ServiceMessage	Message for use in a Service Event-Trace.
ServiceMethod	A behavioural feature of a ServiceSpecification whose behaviour is specified in a ServiceFunction.
ServiceParameter	A type that represents inputs and outputs of a ServiceFunction, represents inputs and outputs of a ServiceSpecification.
ServicePolicy	A constraint governing the use of one or more ServiceSpecifications.
ServicePort	An interaction point for a ServiceSpecification through which it can interact with the outside environment and which is defined by a ServiceInterface.
ServiceSpecification	The specification of a set of functionalities provided one element for the use of others.
ServiceSpecificationGeneralisation	No definition
ServiceSpecificationRole	A behavioural feature of a ServiceSpecification whose behaviour is specified in a ServiceFunction.
ServiceStateDescription	A state machine describing the behaviour of a ServiceSpecification, depicting how the ServiceSpecification responds to various events and the actions.

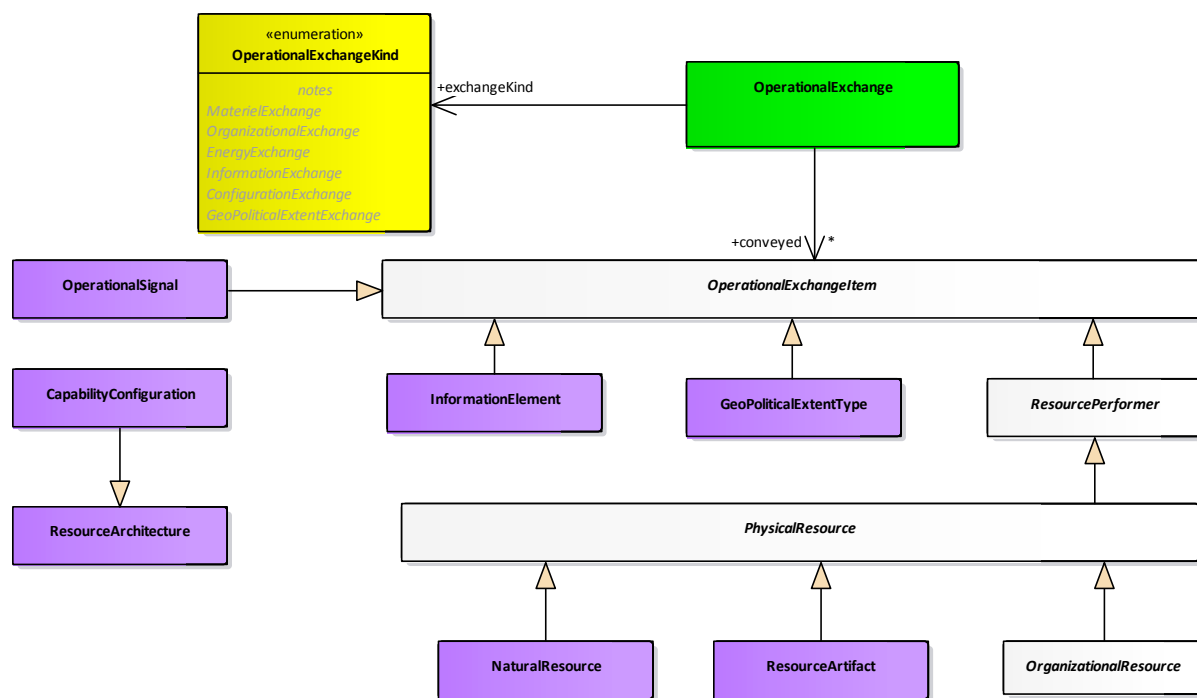
DECLASSIFIED - PUBLIC DISCLOSURE / DÉCLASSIFIÉ - MIS EN LECTURE PUBLIQUE

	Taxonomy	Structure	Connectivity	Processes	States	Sequences	Information	Constraints	Roadmap
Logical Specifications	L1 Node Types NAV-2	L2 Logical Scenario NOV-2	L3 Node Interactions NOV-2, NOV-3	L4 Logical Activities NOV-5	L5 Logical States NOV-6b	L6 Logical Sequence NOV-6c	L7 Logical Data Model NSV-11a	L8 Logical Constraints NOV-6a	Lr Lines of Development NPV-2
	L2-L3 (NOV-1)								

The Viewpoints in the Logical Specifications row of the NAF grid support the solution-independent description of the logical nodes (elements of capability), activities, and resource/information exchanges required to accomplish missions.



Additional DMM extract OperationalExchangeItem:



4.1 L1 – Node Types

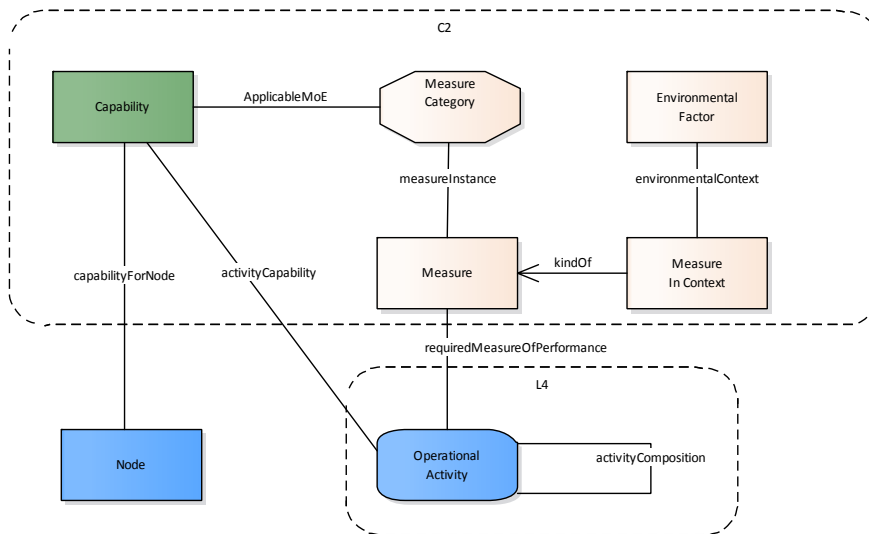
The L1 Viewpoint defines all of the Nodes that will appear in a Logical Architecture. Nodes are elements of capability assembled and orchestrated in the Logical Architecture (see L2 - Logical Scenario and L4 - Logical Activities views). The levels of capability provided by each node are expressed as Measures of Performance (MoPs) and these may be dependent on the environments in which the Node is required to operate.

Nodes are traced to the Capabilities they manifest using the CapabilityForNode relationship. MoPs may be specified for each node that maps to the MoE of the Capabilities. Each Measure can be expressed in terms of the environment – e.g. higher performance may be expected in clear weather than in bad.

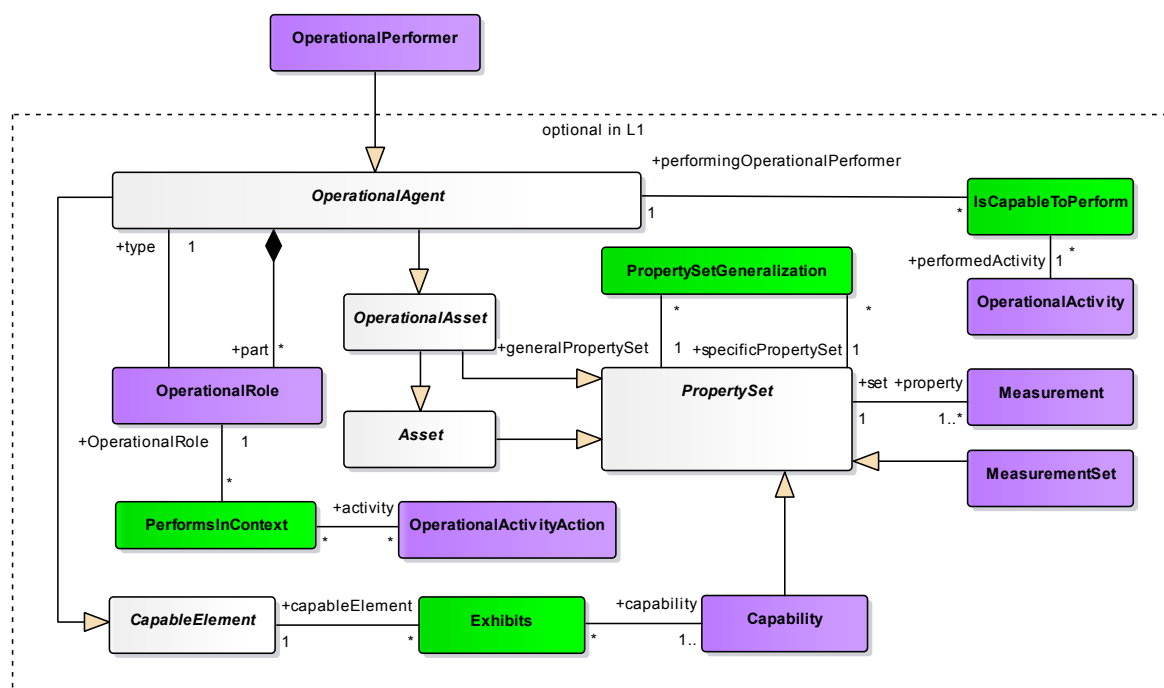
4.1.1 Concerns

- identification all nodes relevant for the architecture,
- (optional) specialization hierarchy for nodes,
- (optional) trace of nodes to capabilities they need,
- (optional) trace of nodes to roles they are performing in activities, and
- (optional) inclusion of Measures of Performance (MoP).

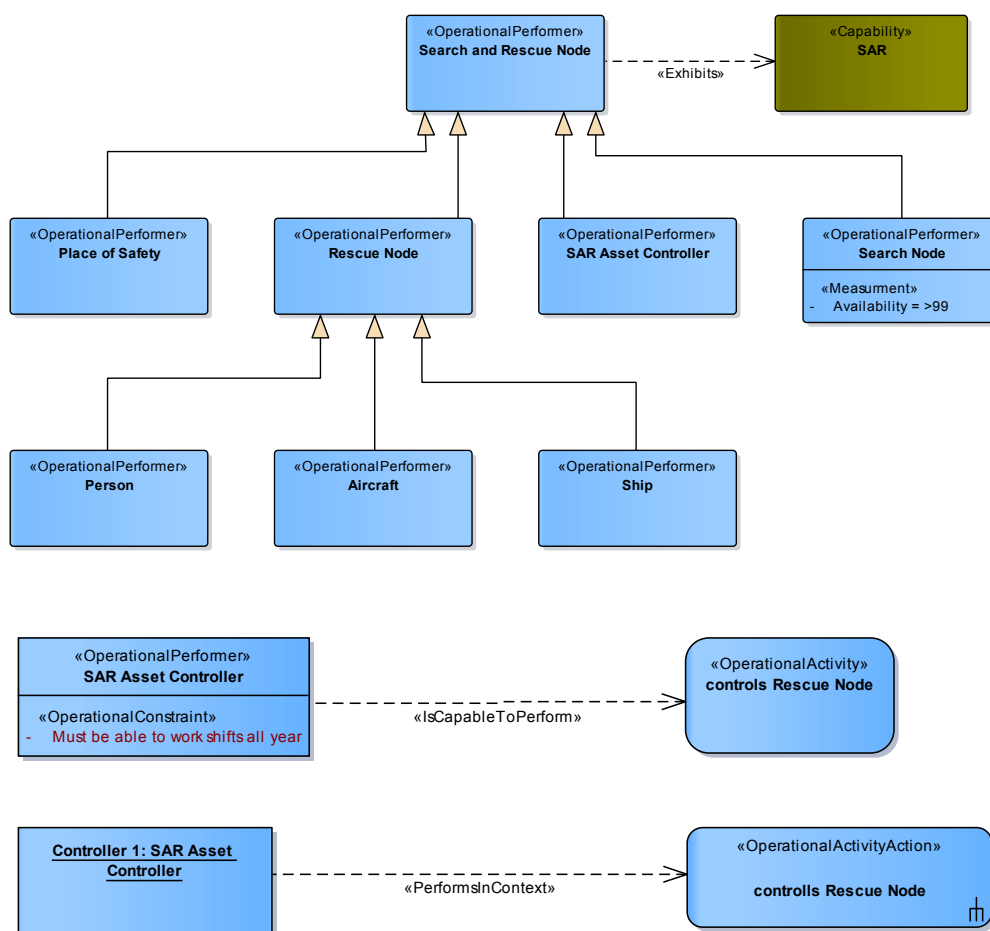
4.1.2 NAFv4 SMM



4.1.3 UAF DMM Extract



4.1.4 Example



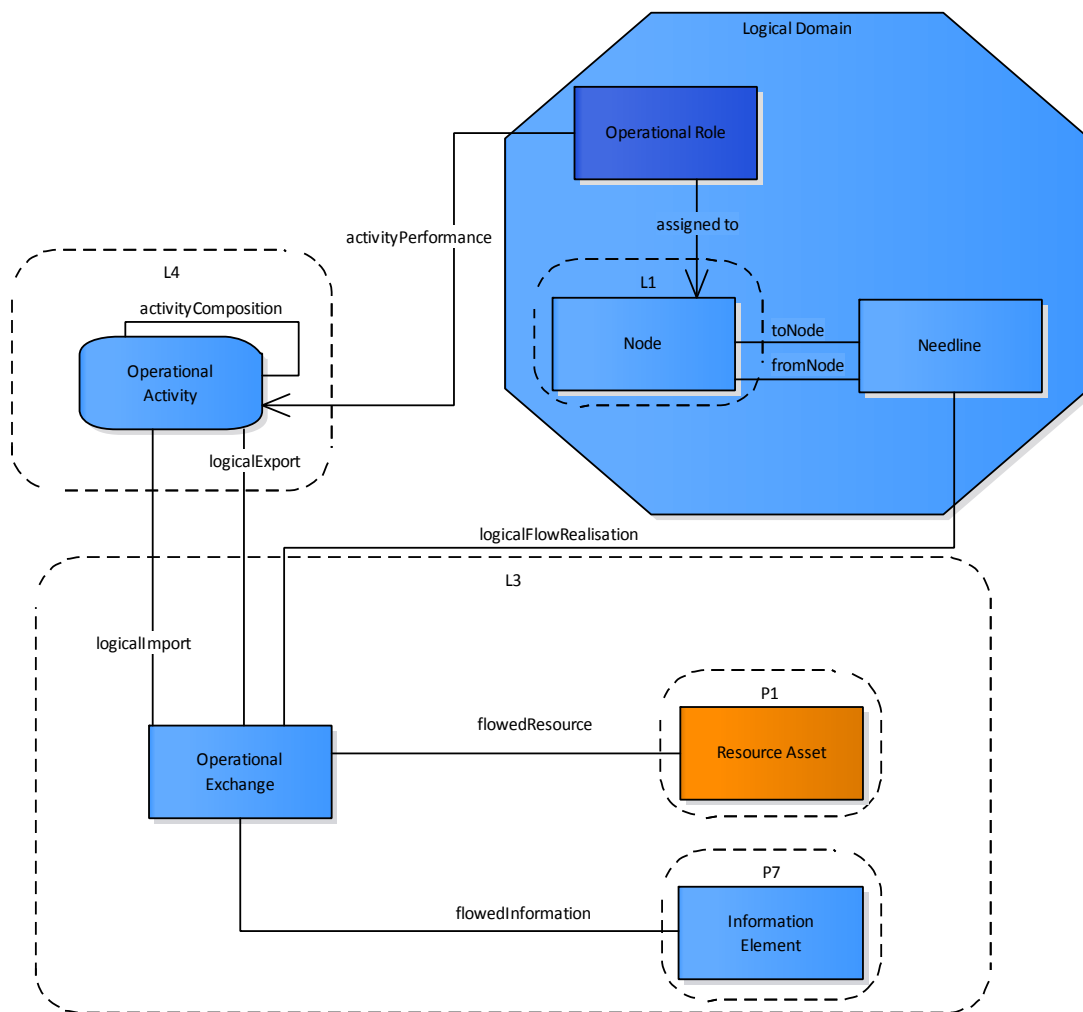
4.2 L2 – Logical Scenario

The L2 Viewpoint specifies Nodes (elements of capability) in context of each other. The context is usually expressed in terms of the information that flows between the nodes (e.g. the information flow requirements between capabilities in a given scenario) but may also be flows of materiel, human resource or energy.

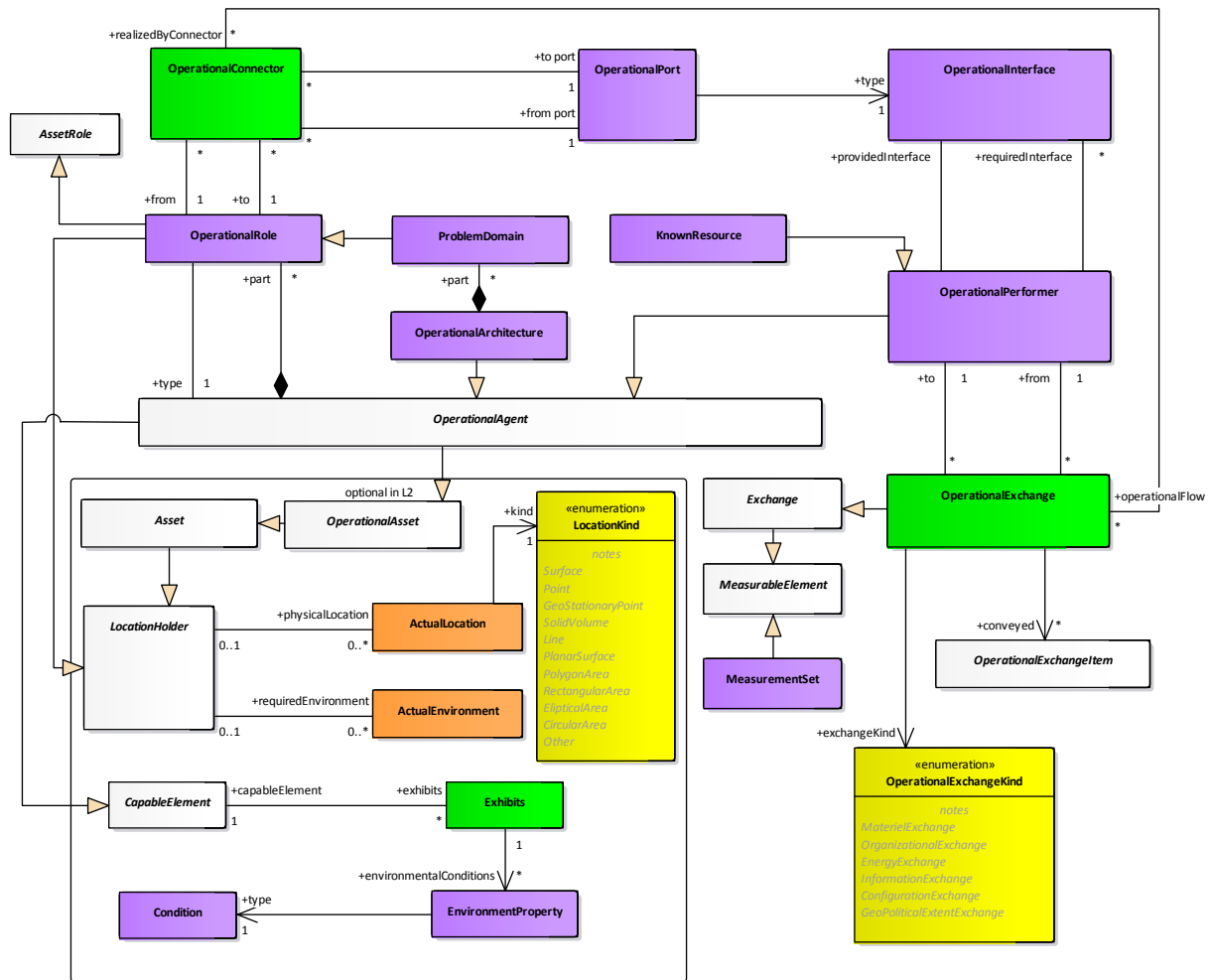
4.2.1 Concerns

- identification of key or aggregated interactions between nodes, and
- (optional) mapping of node to locations.

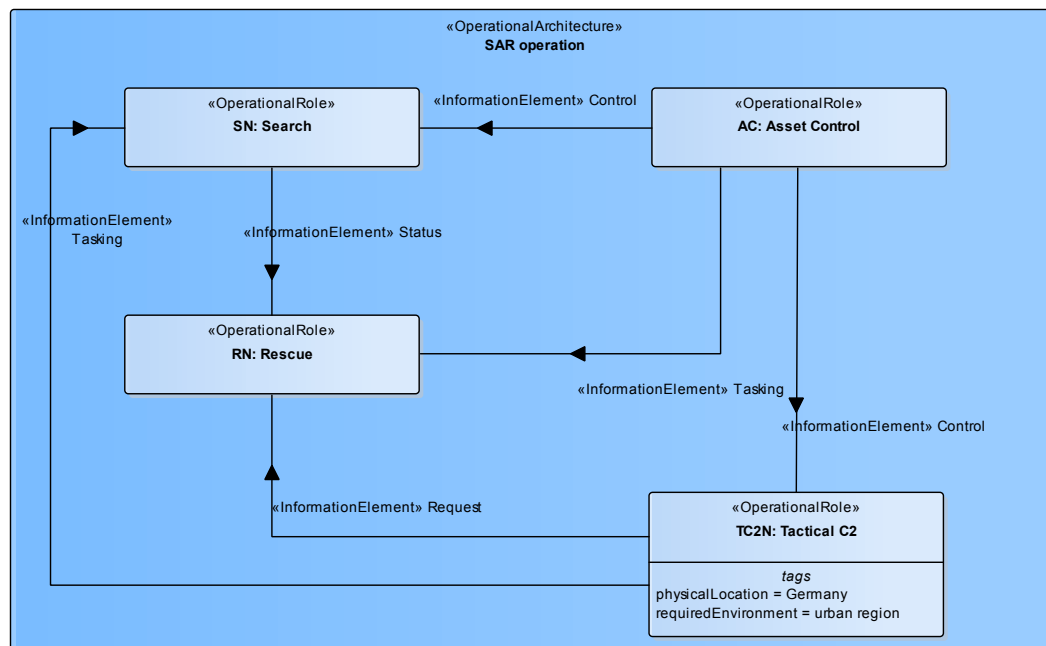
4.2.2 NAFv4 SMM



4.2.3 UAF DMM Extract



4.2.4 Example



4.3 L2-L3

The L2-L3 Viewpoint is concerned with providing an executive level, scenario-based communication of the architecture purpose, scope and content.

4.3.1 Concerns

- presentation of the main elements in scope of the Architecture Description,
- presentation of the main interactions of these elements,
- (optional) presentation of the interactions with elements outside the scope,
- (optional) inclusion of any meta-model element, and
- (optional) inclusion of rich pictures or graphics.

4.3.2 NAFv4 SMM

For this release, no formalized SMM is provided.

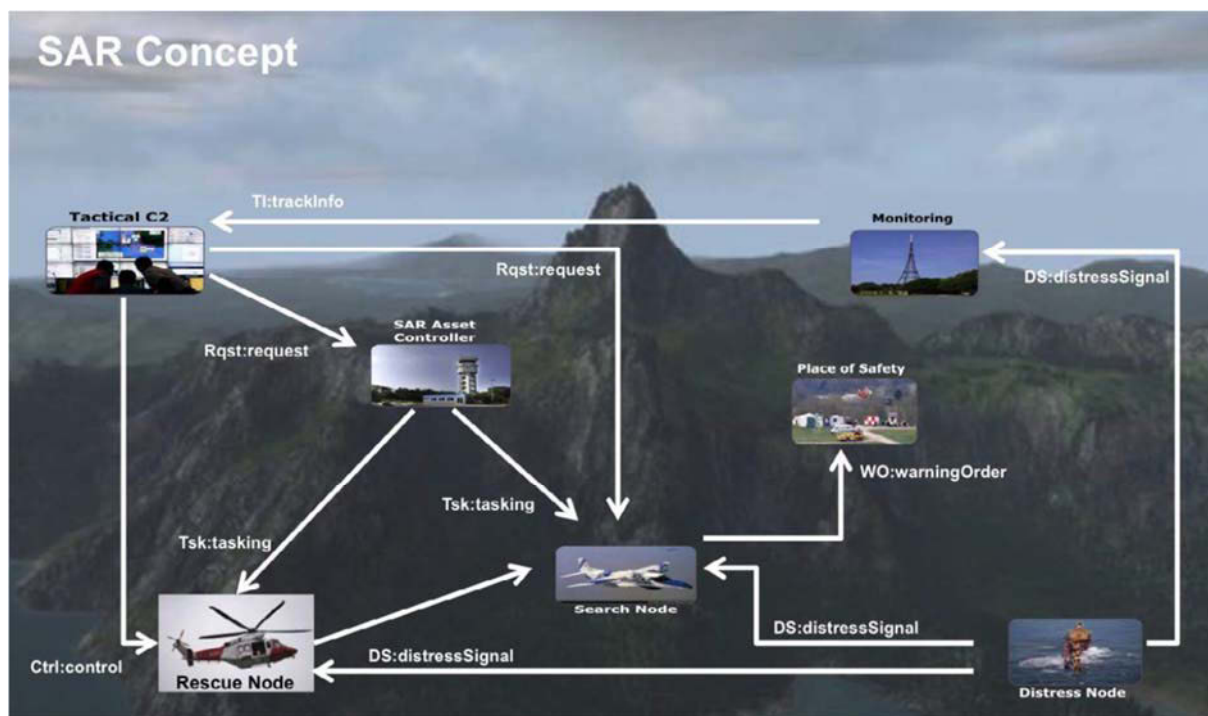
4.3.3 UAF DMM Extract

For this release, no formalized DMM extract is provided. The L2-L3 is supposed to be a graphical representation of the high-level operational concept. Basically, all modeling concepts and also graphical representation is possible.

4.3.4 Elements

All concepts and graphical notations, text is only recommended for short descriptions.

4.3.5 Example



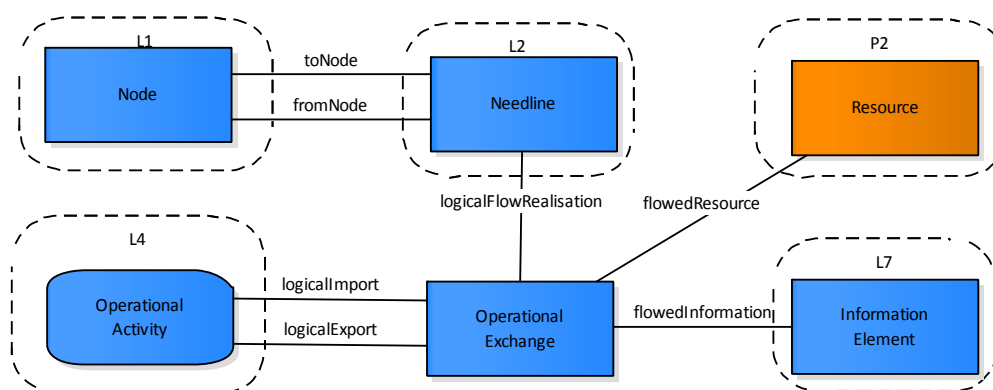
4.4 L3 – Node Interactions

L3 is used to provide further detail of the interoperability requirements associated with the operational capability of interest. The focus is on logical flows that cross the capability boundary. Although the primary purpose of the L3 Viewpoint is to specify information exchanges, the L3 may also list flows of materiel, energy and human resources.

4.4.1 Concerns

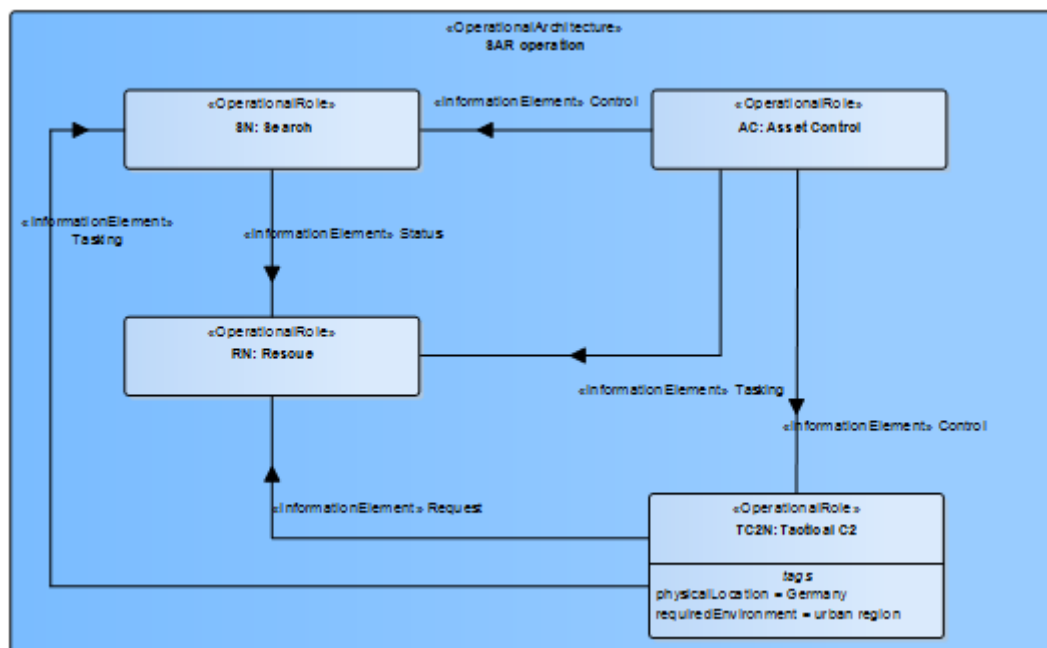
- identification of all logical flows between nodes,
- (optional) association of logical flows to logical activities,
- (optional) define properties of the logical flows, and
- (optional) define measure of the logical flows.

4.4.2 NAFv4 SMM





4.4.4 Example



	Purpose	Source	Target	Transaction Type	Media Type	Interoperability Level Required	Availability	Associated Collaborations	Information Element is conveyed by Information Exchange
Ctrl:control	control rescue	Tactical C2	Rescue Node		Voice	Level 3	95	lex:Controls	Assignment Search Object
D3:distressSignal	signal rescue node	Person in Distress	Rescue Node		Voice			lex:DistressSignal	Last Known Position
D3:distressSignal	signal monitoring station	Person in Distress	Monitoring					lex:DistressSignal	Last Known Position
D3:distressSignal	signal search node	Person in Distress	Search Node					lex:DistressSignal	Last Known Position
Rqst:request	request search	Tactical C2	Search Node					lex:Request	SAR Operation
Rqst:request	asset required	Tactical C2	SAR Asset Controller					lex:Request	SAR Operation
Stat:status	status update	Rescue Node	Search Node					lex:Status	Search Status
TI:trackingInfo	tracking information	Monitoring	Tactical C2					lex:Tracking	Last Known Position
Tsk:tasking	task search	SAR Asset Controller	Search Node					lex:Tasking	Assignment
Tsk:tasking	task rescue	SAR Asset Controller	Rescue Node					lex:Tasking	Assignment
WO:warningOrder	warning	Search Node	Place of Safety					lex:WarningOrder	Search Object

4.5 L4 - Logical Activities

The L4 Viewpoint describes the operational activities that are being conducted within the mission or scenario. These activities are defined at a logical, solution-neutral level so as to enable different solutions in the physical layer. The L4 Viewpoint describes the activities associated with the logical architecture, as well as the

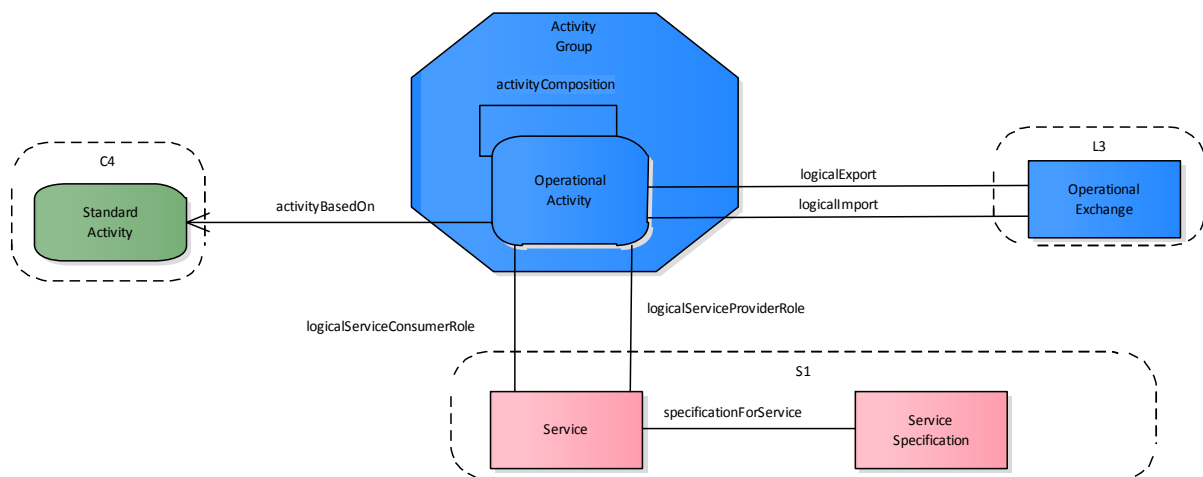
- relationships or dependencies among the operational or business processes,
- information exchanged between operational or business processes, and
- external interchanges (from/to operational or business processes that are outside the scope of the architecture).

An operational activity is a logical process, specified independently of how it is carried out. To maintain this independence from implementation, logical nodes in the L2 Viewpoint are used to represent the structure which carries out the operational activities. Operational activities are realized as resource functions (P4) which are the “how” to the Operational Activities’ “what” – i.e. they are specified in terms of the resources that carry them out.

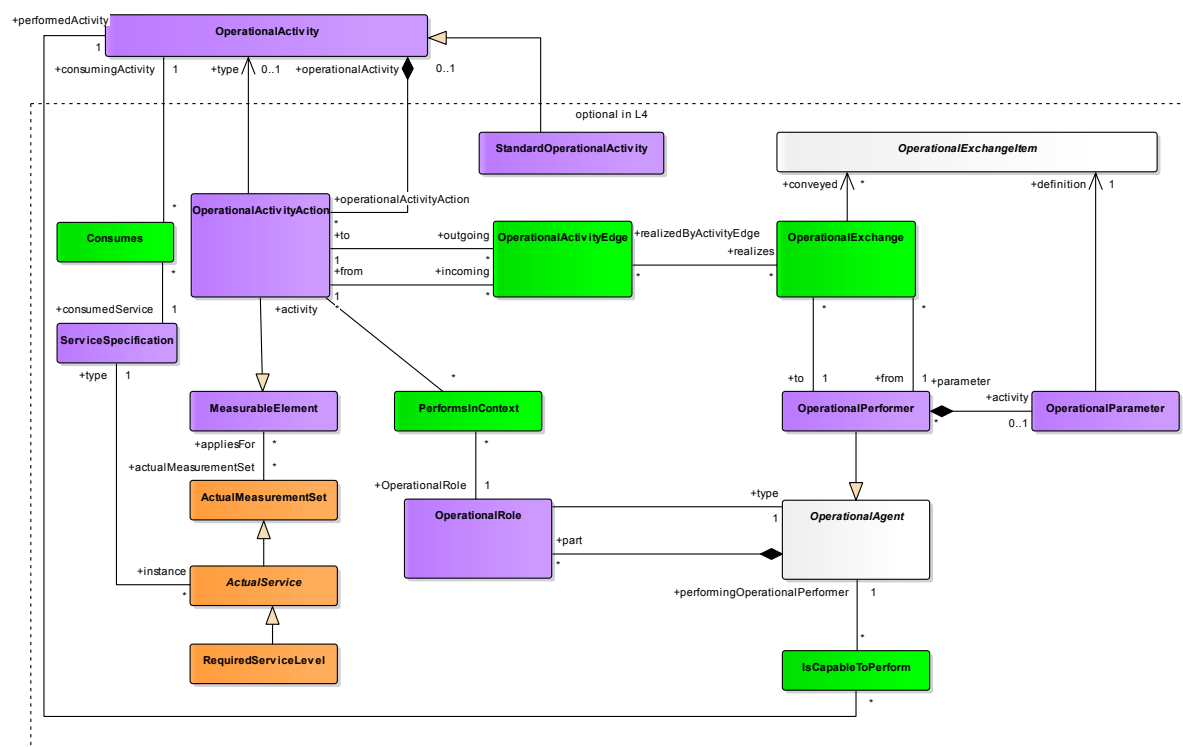
4.5.1 Concerns

- identification of logical activities relevant for the architecture,
- (optional) grouping of activities,
- (optional) composition of activities,
- (optional) association of logical activities to nodes, and
- (optional) identification of logical flows between activities.

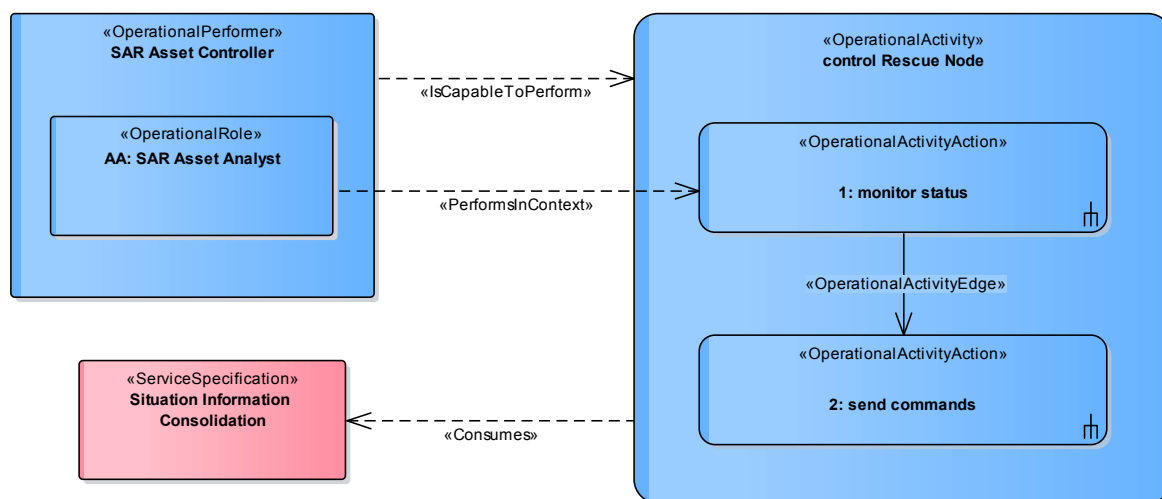
4.5.2 NAFv4 SMM



4.5.3 UAF DMM Extract



4.5.4 Example



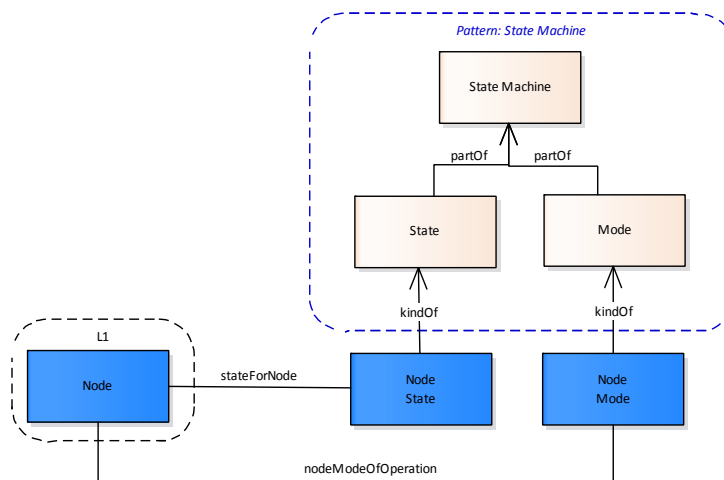
4.6 L5 - Logical States

The L5 Viewpoint specifies the typical states a node may have and the possible transitions between those states (i.e. changes of state). Triggers for state changes may also be defined. Actions may be associated with a given state or with the transition between states in response to stimuli (e.g. triggers and events).

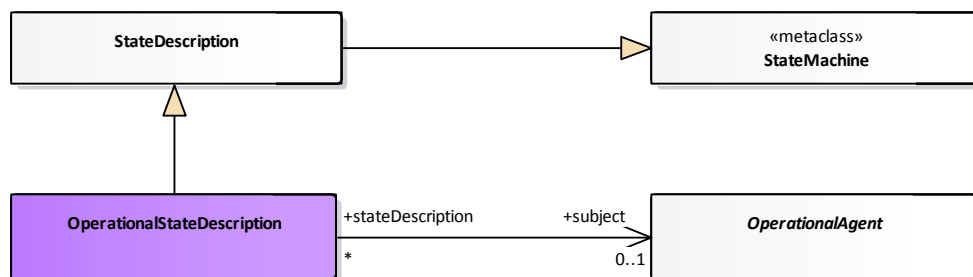
4.6.1 Concerns

- identification and definition of the possible states a node may have and the possible transitions between those states.

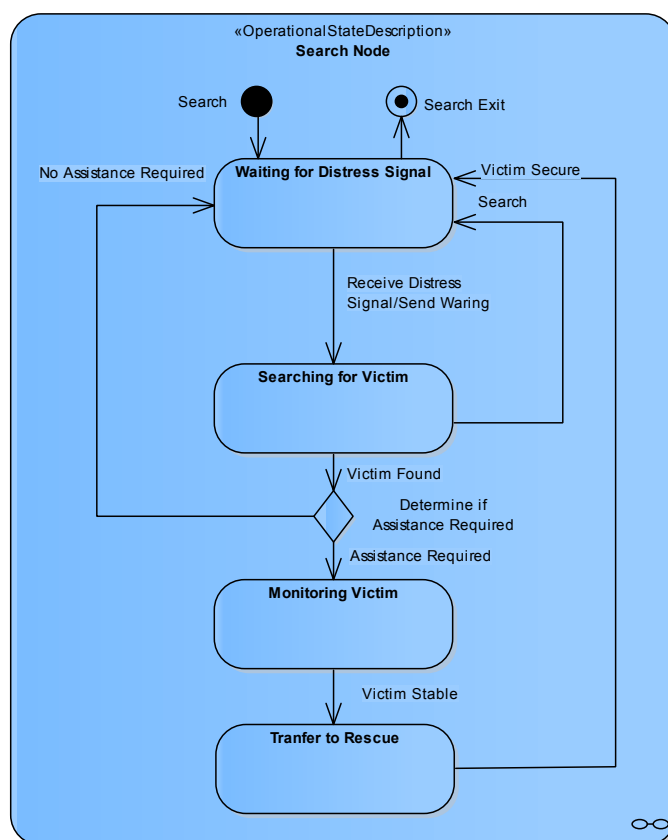
4.6.2 NAFv4 SMM



4.6.3 UAF DMM Extract



4.6.4 Example



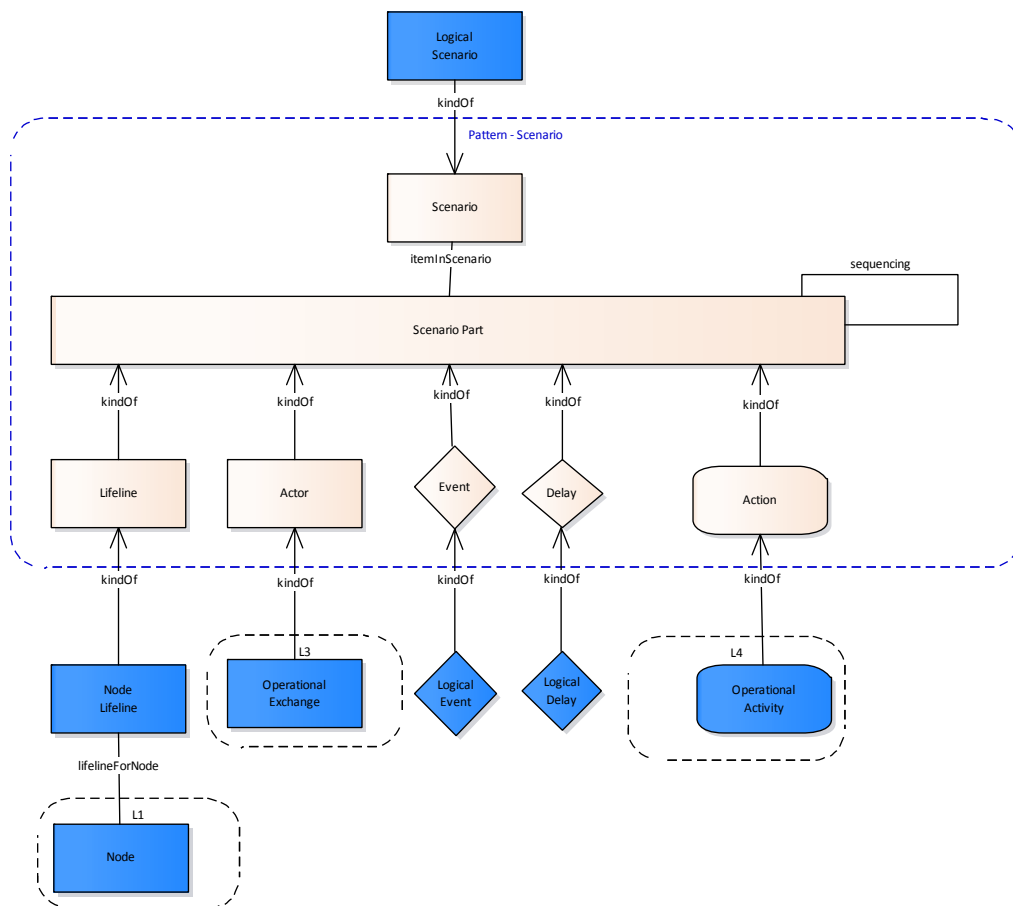
4.7 L6 – Logical Sequence

Operational Event-Trace Descriptions, sometimes called sequence diagrams, event scenarios or timing diagrams, allow the tracing of interactions between nodes in a scenario or critical sequence of events. The node interactions usually correspond to flows of information but may describe flows of energy, materiel or people specified in the L2, Logical Scenario.

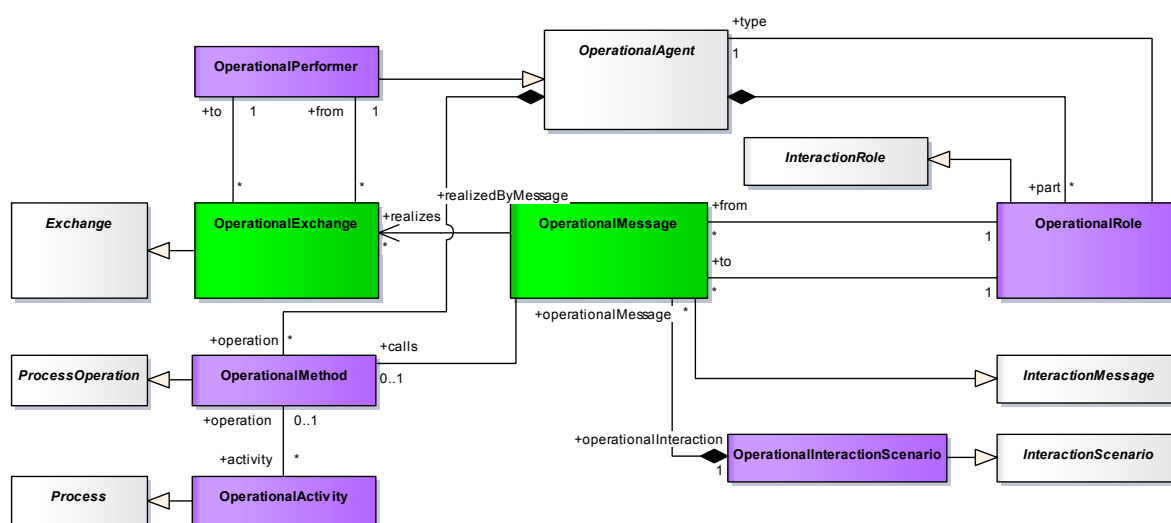
4.7.1 Concerns

- identification and description of chronological sequence of activities and or logical flows in a scenario,
- (optional) identification of source and target nodes of logical flows, and
- (optional) identification of start and end events of a sequence.

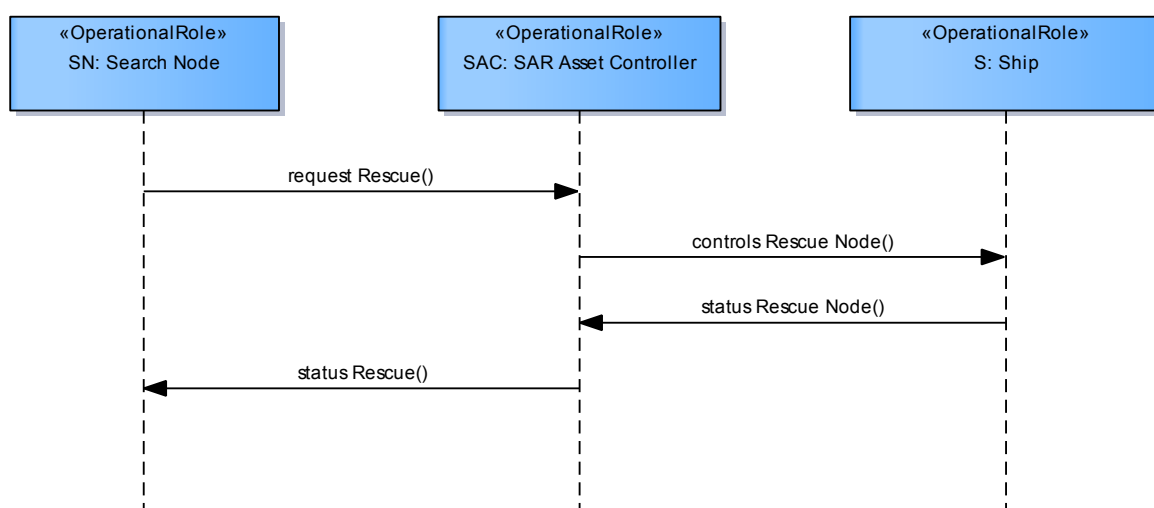
4.7.2 NAFv4 SMM



4.7.3 UAF DMM Extract



4.7.4 Example



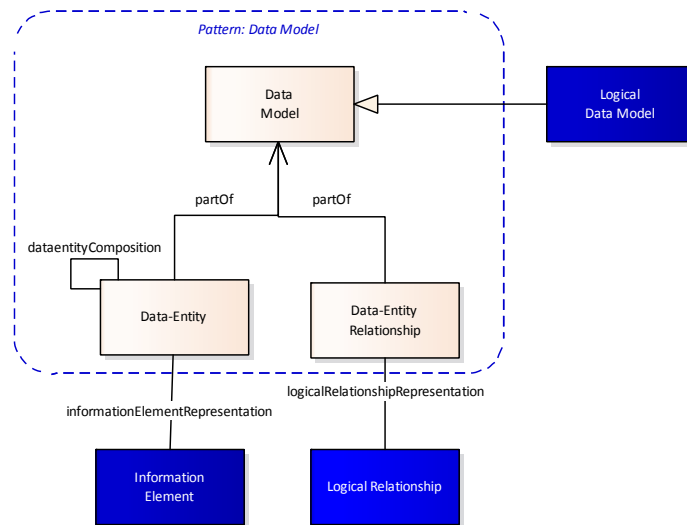
4.8 L7 – Logical Data Model

The L7 Viewpoint is used to document the business information. It describes the information that can be exchanged along the logical flows in the architecture, specified in L3 Node Interactions.

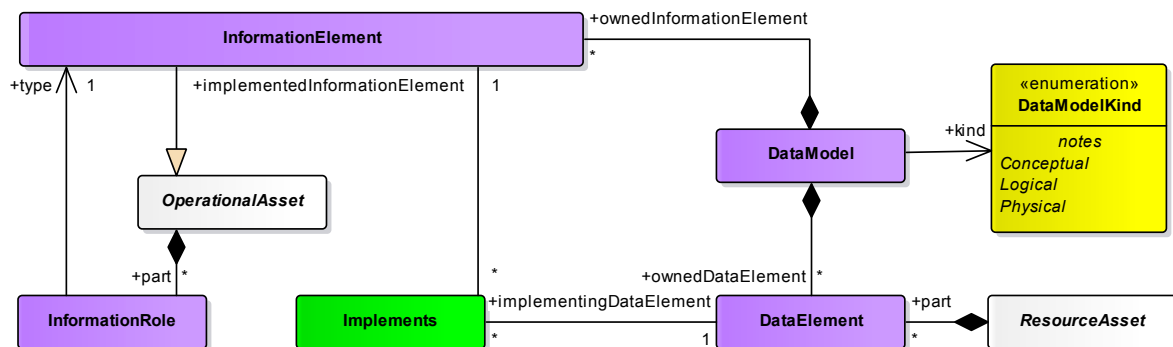
4.8.1 Concerns

- identification of information elements,
- (optional) identification of relationships between information elements,
- (optional) identification of information element attributes, and
- (optional) association of attributes with data entities.

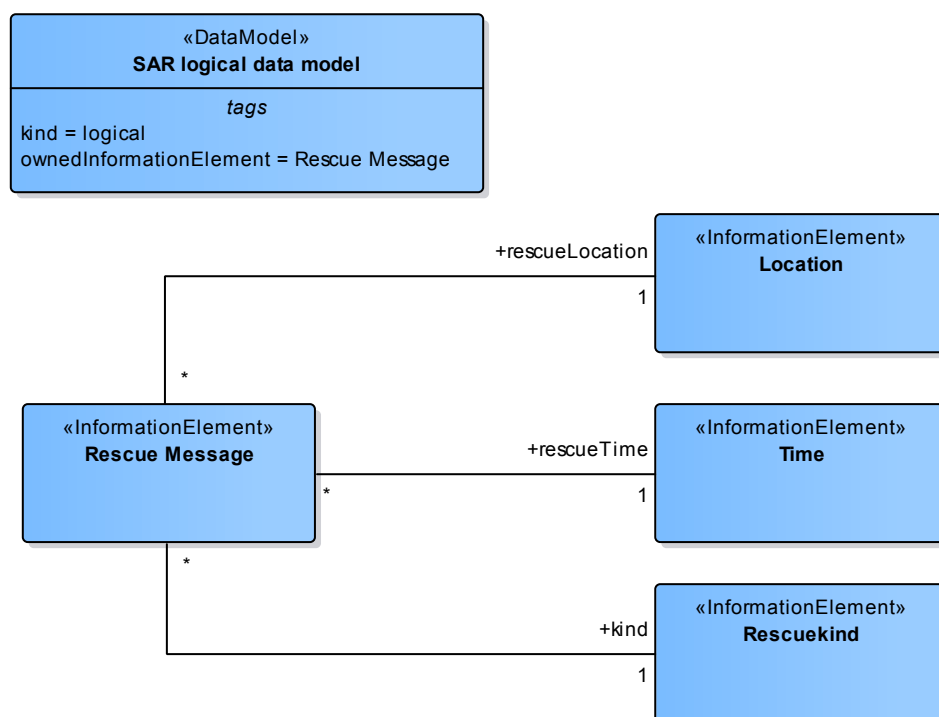
4.8.2 NAFv4 SMM



4.8.3 UAF DMM Extract



4.8.4 Example



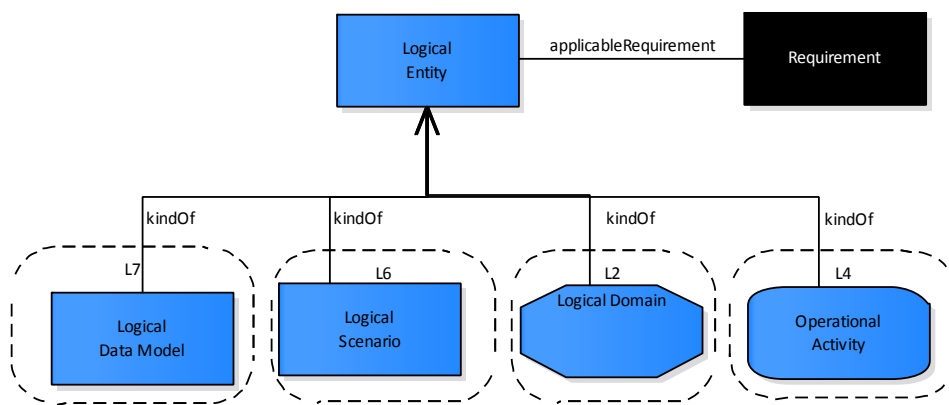
4.9 L8 – Logical Constraints

The L8 Viewpoint is used to constrain the logical architecture without forcing a particular solution. L8 is used for rules which are not expressed as behavioural models, interactions or measures of effectiveness, i.e. they are textual statements of requirements that constrain the architecture.

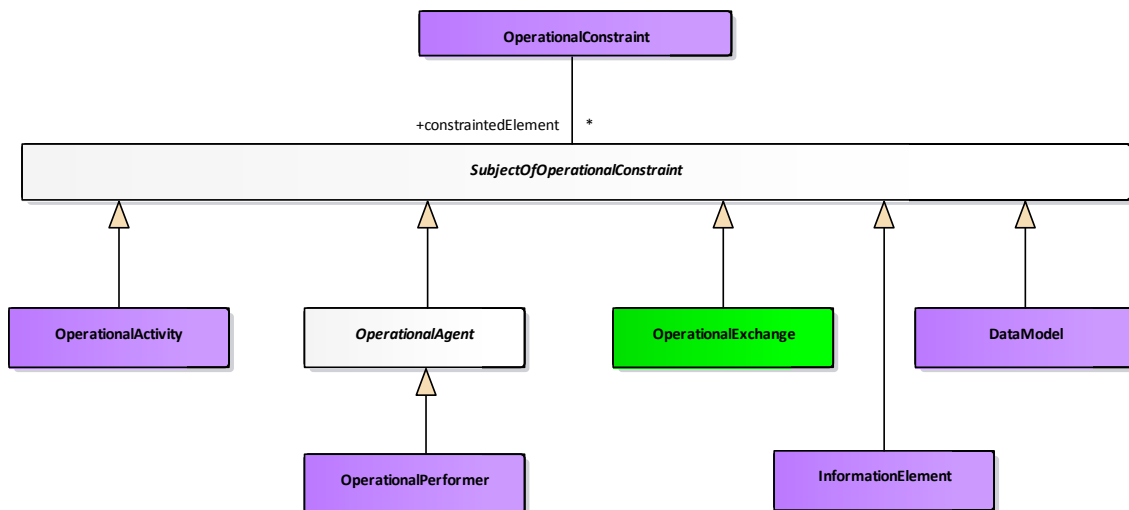
4.9.1 Concerns

- Identification of operational or business rules relevant for the architecture (these rules may be behavioural, or may simply be non-functional constraints) , and
- assignment of these rules to nodes, activities and/or logical flows.

4.9.2 NAFv4 SMM



4.9.3 UAF DMM Extract



4.9.4 Example

«Operational Performer» SAR Asset Controller
«Operational Constraint» - Must be able to work shifts all year

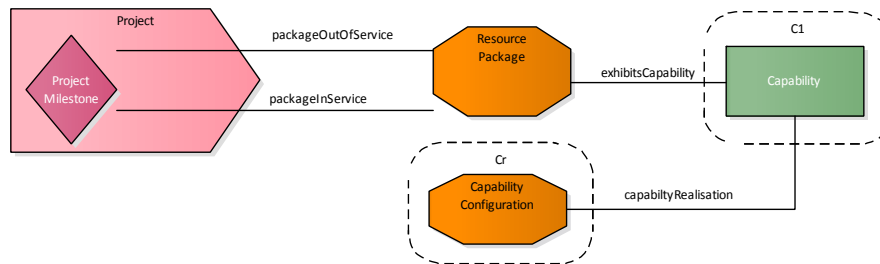
4.10 Lr – Lines of Deployment

The Lr Viewpoint is primarily intended to support the acquisition process across multiple projects or programmes, highlighting dependencies the logical dependencies between capabilities, projects and the integration of all lines of development to achieve a successfully integrated military capability. Use of the Lr Viewpoint should support the management of capability delivery and be aligned with Cr, Capability Roadmap.

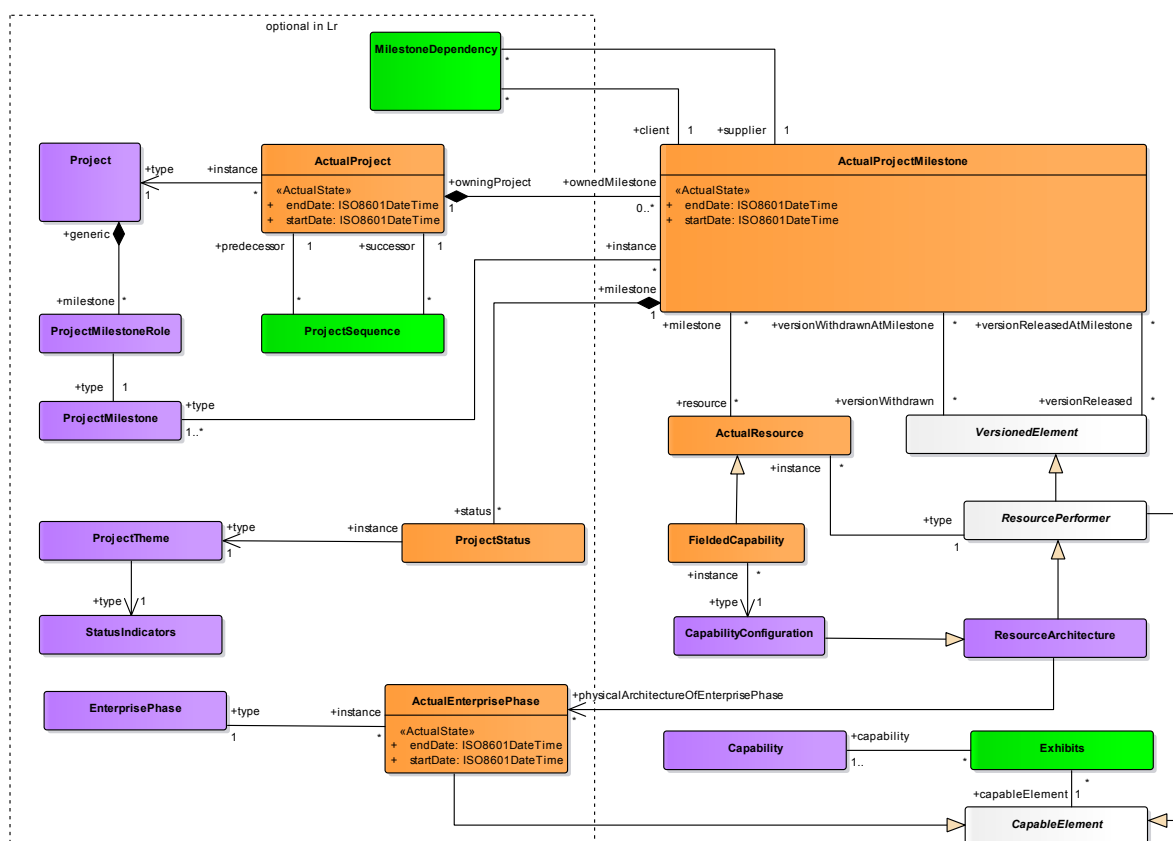
4.10.1 Concerns

- identification of project deliverables,
- association of project deliverables with project milestone,
- (optional) presentation of states of deliverables at project milestones,
- (optional) association of project deliverables with enterprise phase, and
- (optional) presentation of project milestone dependencies.

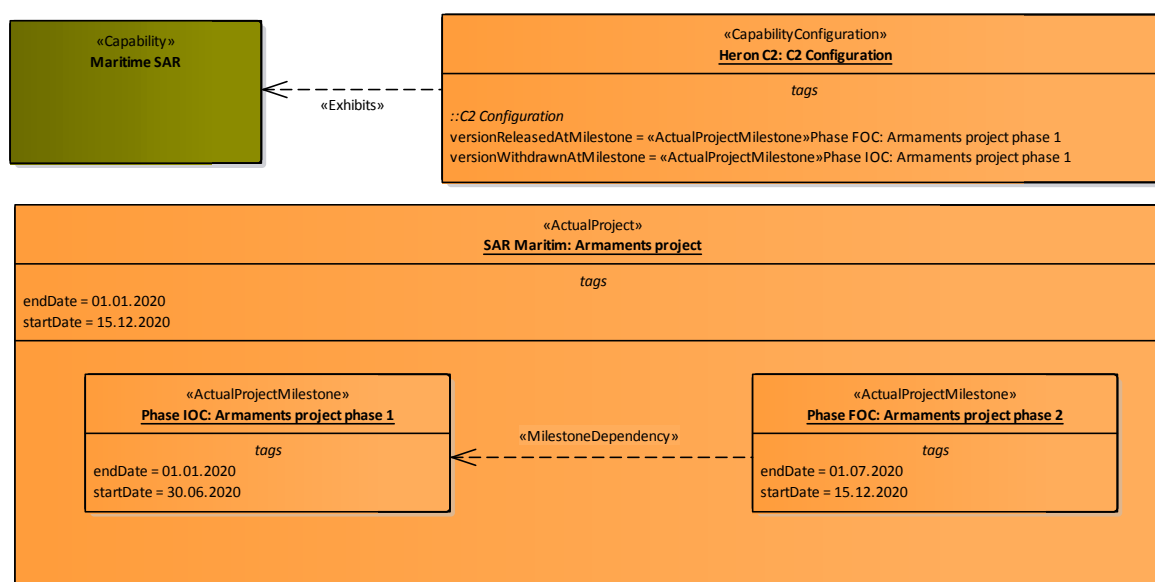
4.10.2 NAFv4 SMM



4.10.3 UAF DMM Extract



4.10.4 Example



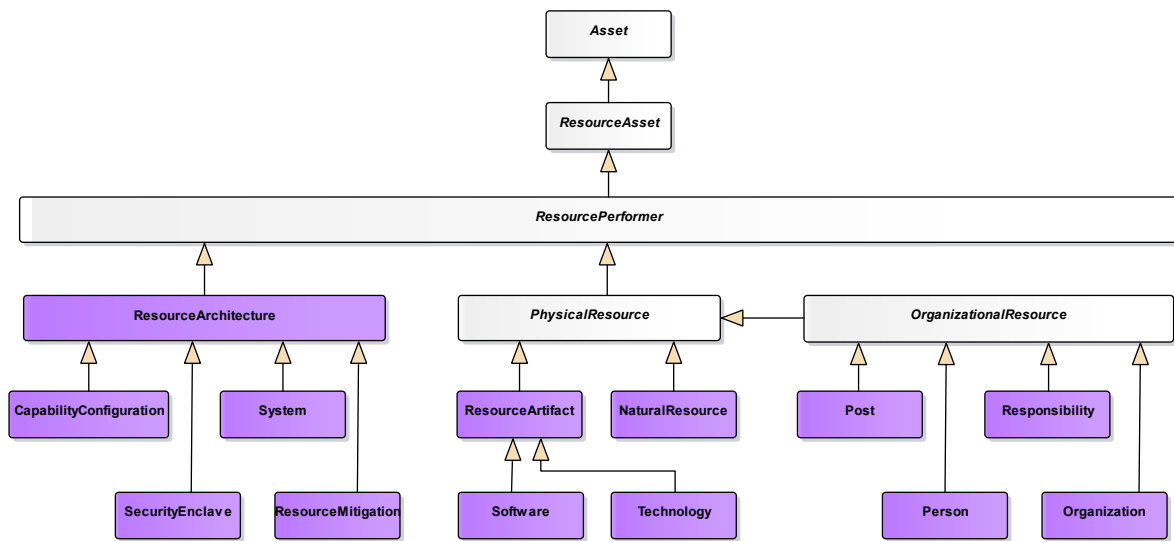
4.11 Elements

NAME	DEFINITION
ActualEnvironment	The ActualState that describes a physical location, for example using text to provide an address, Geo-coordinates, etc.
ActualEnterprisePhase	The ActualState that describes the phase of an Enterprise endeavour.
ActualEnvironment	The ActualState that describes a physical location, for example using text to provide an address, Geo-coordinates, etc.
ActualLocation	The ActualState that describes the circumstances of an Environment.
ActualMeasurementSet	A set of ActualMeasurements.
ActualProject	A time-limited endeavour to provide a specific set of ActualResource that meet specific Capability needs.
ActualProjectMilestone	An event with a start date in an ActualProject from which progress is measured.
ActualResource	Role in an Organization, where the role carries the authority to undertake a function - through the ActualOrganizationalResource given the role has the responsibility.
ActualService	An individual ServiceSpecification.
Asset	Asset as applied to Security views, an abstract type that indicates the types of elements that can be considered as a subject for security analysis.
CapableElement	An abstract type that represents a structural element that can perform behaviours (i.e. OperationalActivity).
Capability	A high-level specification of the enterprise's ability to execute a specified course of action.
CapabilityConfiguration	A composite structure representing the physical and human resources (and their interactions) in an enterprise, assembled to meet a capability).
Consumes	A tuple that asserts that a service in someway contributes or assists in the execution of an OperationalActivity.
DataElement	A formalized representation of data that is managed by or exchanged between resources.
DataModel	A structural specification of data types, showing relationships between them that is devoid of implementation detail. The type of data captured in the DataModel is described using the enumeration DataModelKind (Conceptual, Logical and Physical).
Exchange	Abstract tuple, grouping OperationalExchanges and ResourceExchanges that exchange Resources.
Exhibits	A tuple that exists between a CapableElement and a Capability that it meets under specific environmental conditions.
FieldedCapability	An individual, fully-realized capability.
GeoPoliticalExtentType	A geospatial extent whose boundaries are defined by declaration or agreement by political parties.
Implements	A tuple that defines how an element in the upper layer of abstraction is implemented by a semantically equivalent element (i.e. tracing the OperationalActivities to the Functions that implement them) in the lower level of abstraction.
InformationElement	An item of information that flows between OperationalPerformers and is produced and consumed by the OperationalActivities that the OperationalPerformers are capable to perform (see IsCapableToPerform).
InformationRole	A usage of InformationElement that exists in the context of an OperationalAsset. It also allows the representation of the whole-part aggregation of InformationElements.

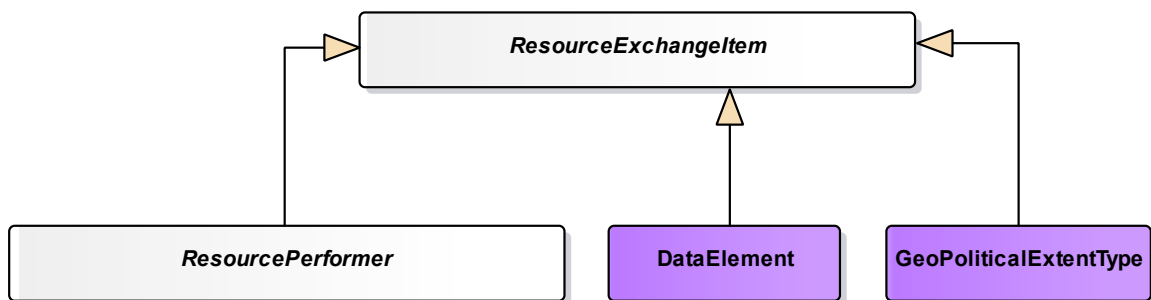
InteractionMessage	An abstract type that represents a behaviour or process (i.e. a Function or OperationalActivity) that can be performed by a Performer.
InteractionScenario	An abstract type that represents a behaviour or process (i.e. a Function or OperationalActivity) that can be performed by a Performer.
IsCapableToPerform	A tuple defining the traceability between the CapableElements to the Activities that they can perform.
KnownResource	Asserts that a known ResourcePerformer constrains the implementation of the OperationalPerformer that plays the role in the LogicalArchitecture.
LocationHolder	Abstract type, used to group elements that are allowed to be associated with a Location.
MeasurableElement	Abstract type, grouping elements that can be measured by applying MeasurementSets to them.
Measurement	A property of an element representing something in the physical world, expressed in amounts of a unit of measure.
MeasurementSet	A collection of Measurements.
MilestoneDependency	A tuple between two ActualProjectMilestones that denotes one ActualProjectMilestone follows from another.
NaturalResource	Type of physical resource that occurs in nature such as oil, water, gas or coal.
OperationalActivity	An Activity that captures a logical process, specified independently of how the process is carried out.
OperationalActivityAction	A call of an OperationalActivity in the context of another OperationalActivity.
OperationalActivityEdge	A tuple that shows the flow of Resources (objects/information) between OperationalActivityActions.
OperationalAgent	An abstract type grouping LogicalArchitecture and OperationalPerformer.
OperationalArchitecture	A type used to denote a model of the Architecture, described from the Operational perspective.
OperationalAsset	No definition.
OperationalConnector	A Connector that goes between OperationalRoles representing a need to exchange Resources. It can carry a number of OperationalExchanges.
OperationalConstraint	A Rule governing a logical architectural element i.e. OperationalPerformer, OperationalActivity, InformationElement etc.
OperationalExchange	Asserts that a flow can exist between OperationalPerformers (i.e. flows of information, people, materiel, or energy).
OperationalExchangeItem	An abstract grouping for elements that defines the types of elements that can be exchanged between OperationalPerformers and conveyed by an OperationalExchange.
OperationalInteractionScenario	A specification of the interactions between OperationalPerformers in an OperationalArchitecture.
OperationalInterface	A declaration that specifies a contract between the OperationalPerformer it is related to, and any other OperationalPerformers it can interact with.
OperationalMessage	Message for use in an Operational Event-Trace which carries any of the subtypes of OperationalExchange.
OperationalMethod	Behavioural feature of a OperationalPerformer whose behaviour is specified in an OperationalActivity.
OperationalParameter	A type that represents inputs and outputs of an OperationalActivity. It is typed by an OperationalExchangeItem.
OperationalPerformer	The specification of a set of functionalities provided one element for the use of others.
OperationalPort	Usage of an OperationalPerformer or LogicalArchitecture in the context of another OperationalPerformer or LogicalArchitecture. Creates a whole-part relationship.

OperationalRole	Usage of an OperationalPerformer or OperationalArchitecture in the context of another OperationalPerformer or OperationalArchitecture. Creates a whole-part relationship.
OperationalSignal	An item of information that flows between OperationalPerformers and is produced and consumed by the OperationalActivities that the OperationalPerformers are capable of performing (see IsCapableToPerform).
OperationalStateDescription	A state machine describing the behaviour of an OperationalPerformer, depicting how the OperationalPerformer responds to various events and the actions.
OrganizationalResource	An abstract type for Organization, Person Post and Responsibility.
PerformsInContext	A tuple that relates an OperationalAction to an OperationalRole, or a FunctionAction to a ResourceRole. It indicates that the action can be carried out by the role when used in a specific context or configuration.
ProblemDomain	A property associated with a logical architecture, used to specify the scope of the problem.
ProcessEdge	An abstract type that represents a behaviour or process (i.e. a Function or OperationalActivity) that can be performed by a Performer.
Project	A type that describes types of time-limited endeavours that are required to meet one or more Capability needs.
ProjectMilestone	A type of event in a Project by which progress is measured.
ProjectMilestoneRole	The role played by a ProjectMilestone in the context of a Project.
ProjectSequence	A tuple between two ActualProjects that denotes one ActualProject cannot start before the previous ActualProject is finished.
ProjectStatus	The status (i.e. level of progress) of a ProjectTheme for an ActualProject at the time of the ActualProjectMilestone.
ProjectTheme	A property of a ProjectMilestone that captures an aspect by which the progress of ActualProjects may be measured.
PropertySet	An abstract type grouping architectural elements that can own Measurements.
PropertySetGeneralization	A PropertySetGeneralization is a taxonomic relationship between a more general PropertySet and a more specific PropertySet.
PhysicalType	An abstract type defining physical resources (i.e. OrganizationalResource, ResourceArtifact and NaturalResource).
RequiredServiceLevel	A sub type of ActualService that details a specific service level required of the provider.
ResourceArchitecture	A type used to denote a model of the Architecture, described from the ResourcePerformer perspective.
ResourceArtifact	A type of man-made object that contains no human beings (i.e. satellite, radio, petrol, gasoline, etc.).
ResourceAsset	No definition
ResourcePerformer	An abstract type grouping element that can be the subject of a SecurityConstraint.
Rule	An abstract type for all types of constraint (i.e. an OperationalConstraint could detail the rules of accountancy best practice).
ServiceSpecification	The specification of a set of functionalities provided one element for the use of others.
StateDescription	An abstract type that represents a state machine (i.e. an OperationalStateDescription or ResourceStateDescription), depicting how the Asset responds to various events and the actions.
SubjectOfOperationalConstraint	An abstract type grouping element that can be the subject of an OperationalConstraint.
VersionedElement	An abstract type grouping ResourcePerformer and ServiceSpecification that allows VersionOfConfiguration to be related to ActualProjectMilestones.

UAF DMM extract ResourcePerformer:



UAF DMM extract ResourceExchangeItem:



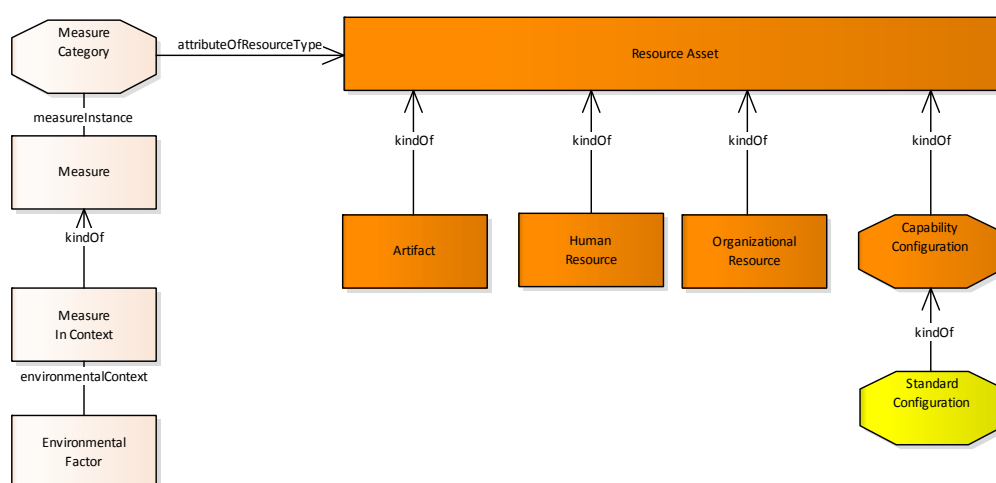
5.1 P1 – Resource Types

The P1 Viewpoint collects together all the Resource Types in the architecture together with a depiction of their performance characteristics. P1 also provides a summary of the technologies and competencies that impact on the Resources that constitute the architecture.

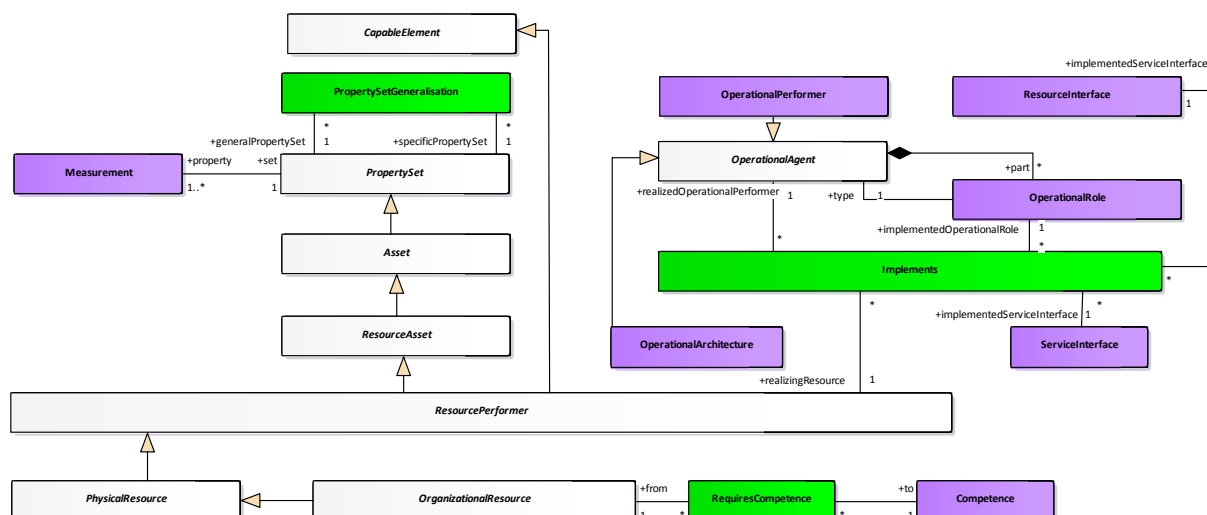
5.1.1 Concerns

- inclusion of all Resource Types relevant for the architecture,
- taxonomy of the technologies and competences,
- (optional) specification of hardware and software, and
- Organization of resources into a specialization hierarchy.

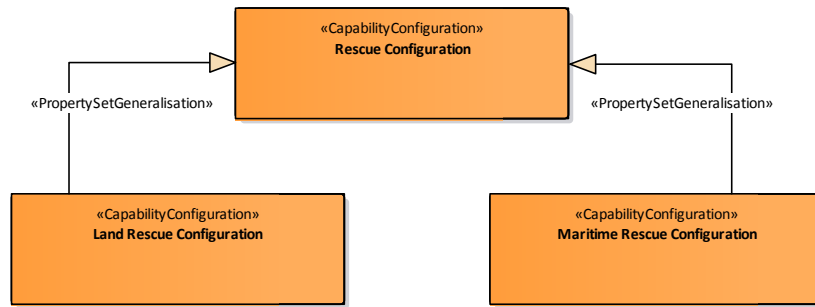
5.1.2 NAFv4 SMM



5.1.3 UAF DMM Extract



5.1.4 Example



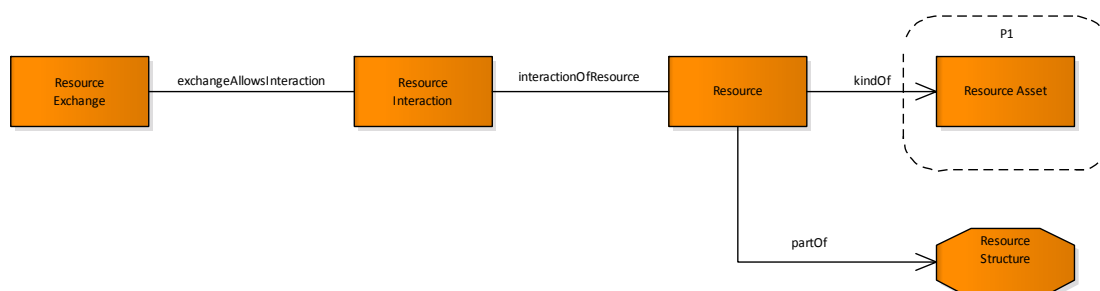
5.2 P2 – Resource Structure

The P2 Viewpoint links together the operational and physical architecture viewpoints by depicting how types of Resource are structured and interact to realize the logical architecture specified in L2, Logical Scenario. The P2 Viewpoint may represent the realization of a requirement specified in a L2 (i.e. in a “to-be” architecture) and so there may be many alternative Resource viewpoint suites that could realize the operational requirement. Alternatively, in an “as-is” architecture, a L2 may just be a simplified, logical representation of the P2 Viewpoint to allow communication of key information flows to non-technical stakeholders.

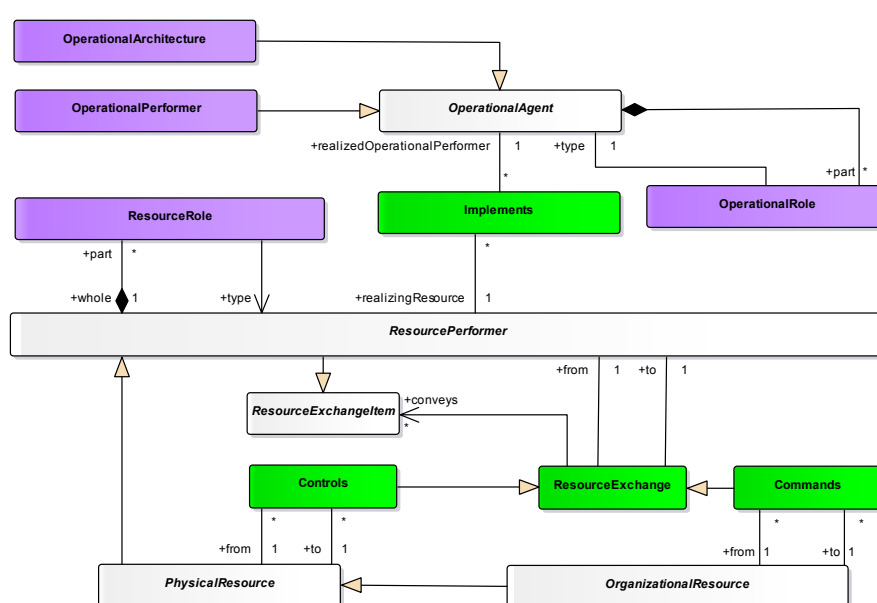
5.2.1 Concerns

- identification of how Resource Types structure and interact to realize the logical architecture specified in L2 – Logical Scenario,
- description of the structure of resources types,
- (optional) representation of the realization of a requirement specified in L2, and
- (optional) specification of typical organizational structures and how human resources interact with each other and with systems.

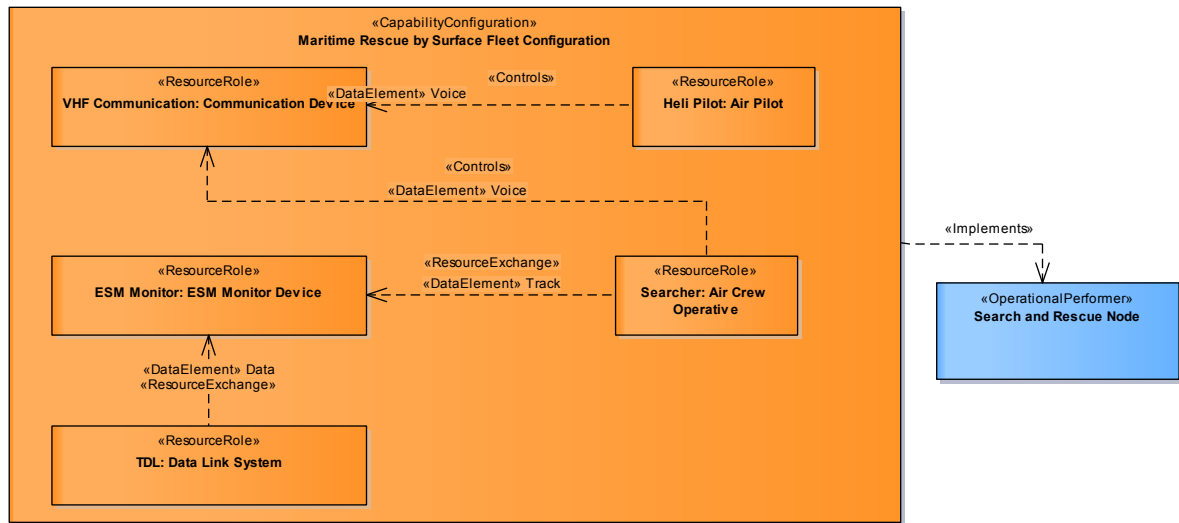
5.2.2 NAFv4 SMM



5.2.3 UAF DMM Extract



5.2.4 Example



5.3 P3 – Resource Connectivity

The networks and pathways documented in a P3 Viewpoint represent the physical implementation of the logical flows identified in a L2, Logical Scenario, or L3, Node Interactions View.

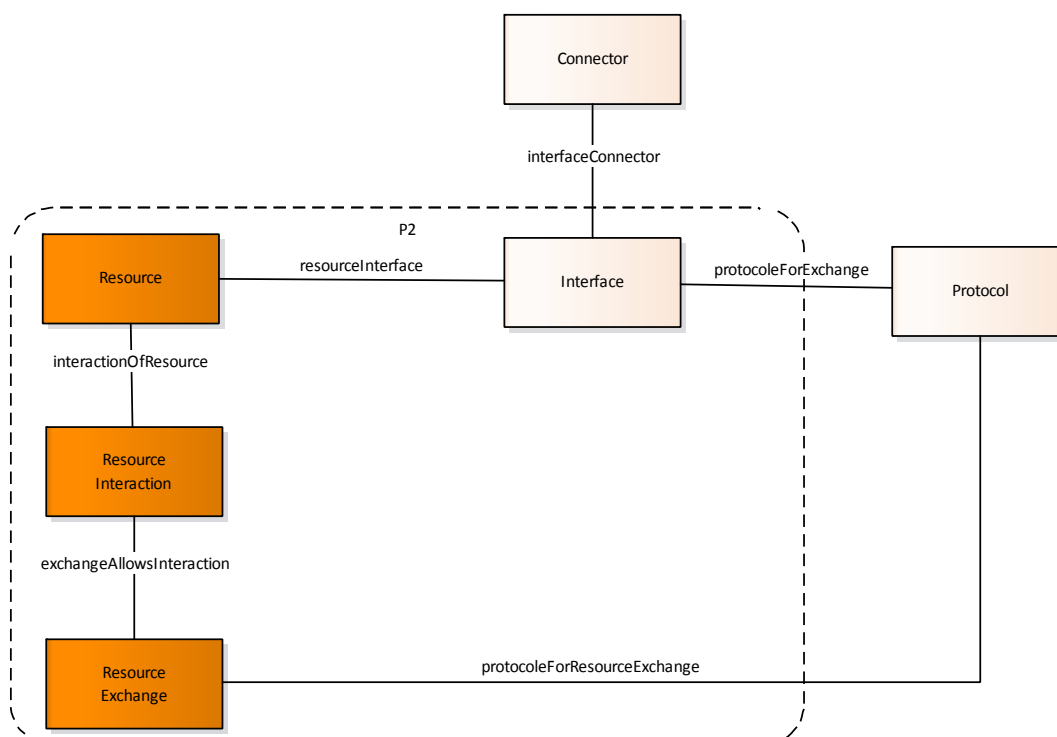
The P3 Viewpoint focuses on the physical characteristics of each link by specifying attributes. An example of this might be the geographic location and layout of network components such as routers, switches, amplifiers and repeaters. Attributes such as capacities (e.g. bandwidth, throughput), frequencies used, security encryption methods used and other descriptive information are also included. The P3 View encompasses the NSV-2b and 2c views from NAFv3.1, which define the communications links between systems.

The P3 Viewpoint differs from that of P2 in that it only features physical architectures, software and artefacts (as systems) – i.e. P3 does not feature any organizational resources. P3 also provides a great deal more technical detail than P2, specifying the protocols implemented by systems and used by the connections between those systems.

5.3.1 Concerns

- physical implementation of logical flows by specifying how systems are connected,
- focus on the physical characteristics of each link by specifying attributes, and
- inclusion of capacities, frequencies/protocols used, security encryption methods used and other descriptive information as attributes.

5.3.2 NAFv4 SMM





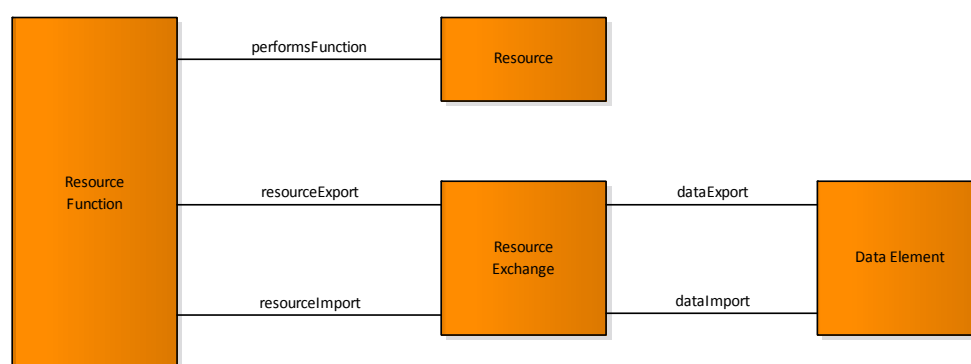
5.4 P4 – Resource Functions

The Functionality Description provides detailed information regarding the allocation of functions to resources, and flows between Resource Functions. The P4 Viewpoint is the Physical Resource counterpart to the L4 Logical Activities Viewpoint.

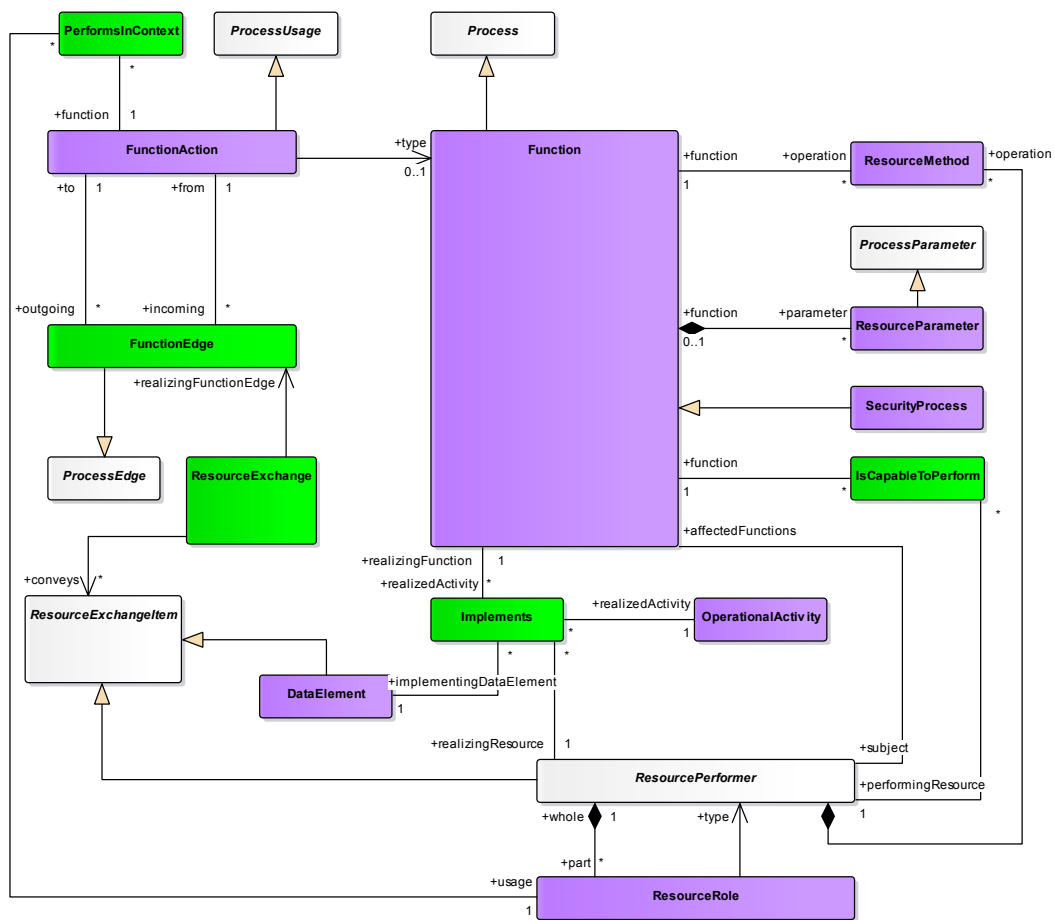
5.4.1 Concerns

- specification of the functionality of resources,
- inclusion of data flow between Resource Function (counterpart to the L4),
- description of implementation-specific realization of the operational activities, and
- inclusion of complete functional connectivity.

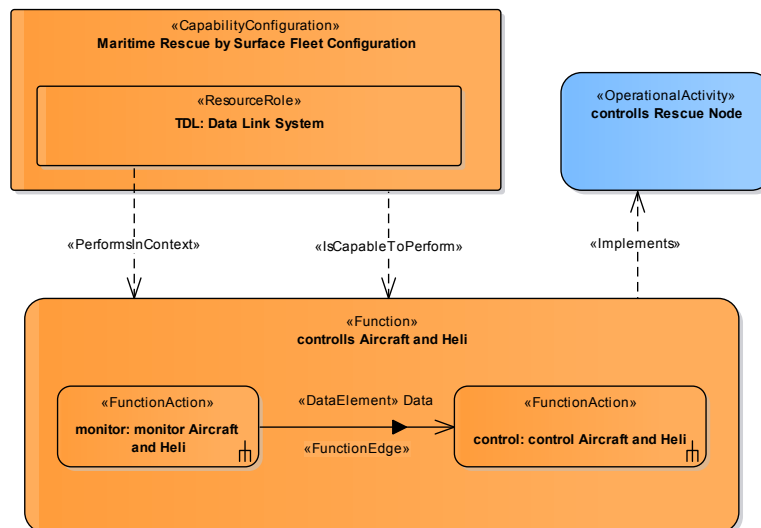
5.4.2 NAFv4 SMM



5.4.3 UAF DMM Extract



5.4.4 Example



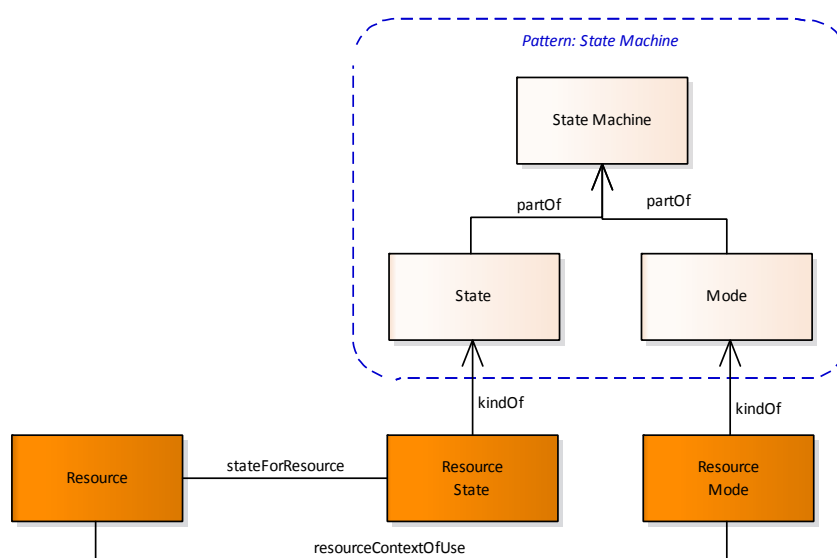
5.5 P5 – Resource States

The P5 Viewpoint identifies the states a Resource can be, the allowable changes between those states, and the stimuli (e.g. triggers and events) that cause the state changes.

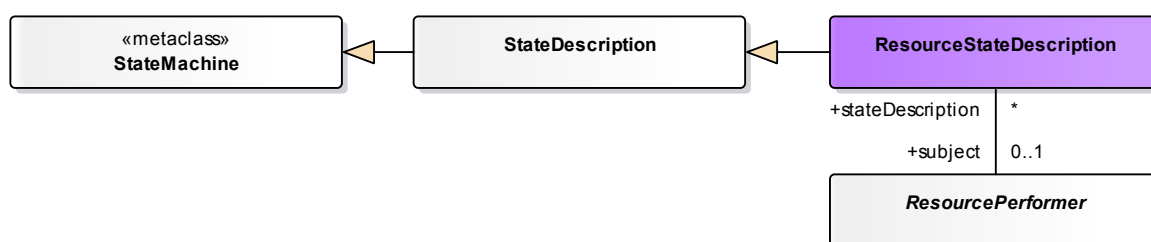
5.5.1 Concerns

- identification of states a resource can be, the allowable changes between those states, and the stimuli that cause the state changes,
- association of stimuli with resource states and describe the transition from one state to another from a resource perspective, with a focus on how the resource responds to stimuli, and
- (optional) describe different responses depending upon the rule set or conditions that apply, as well as the resource state at the time the stimuli are received.

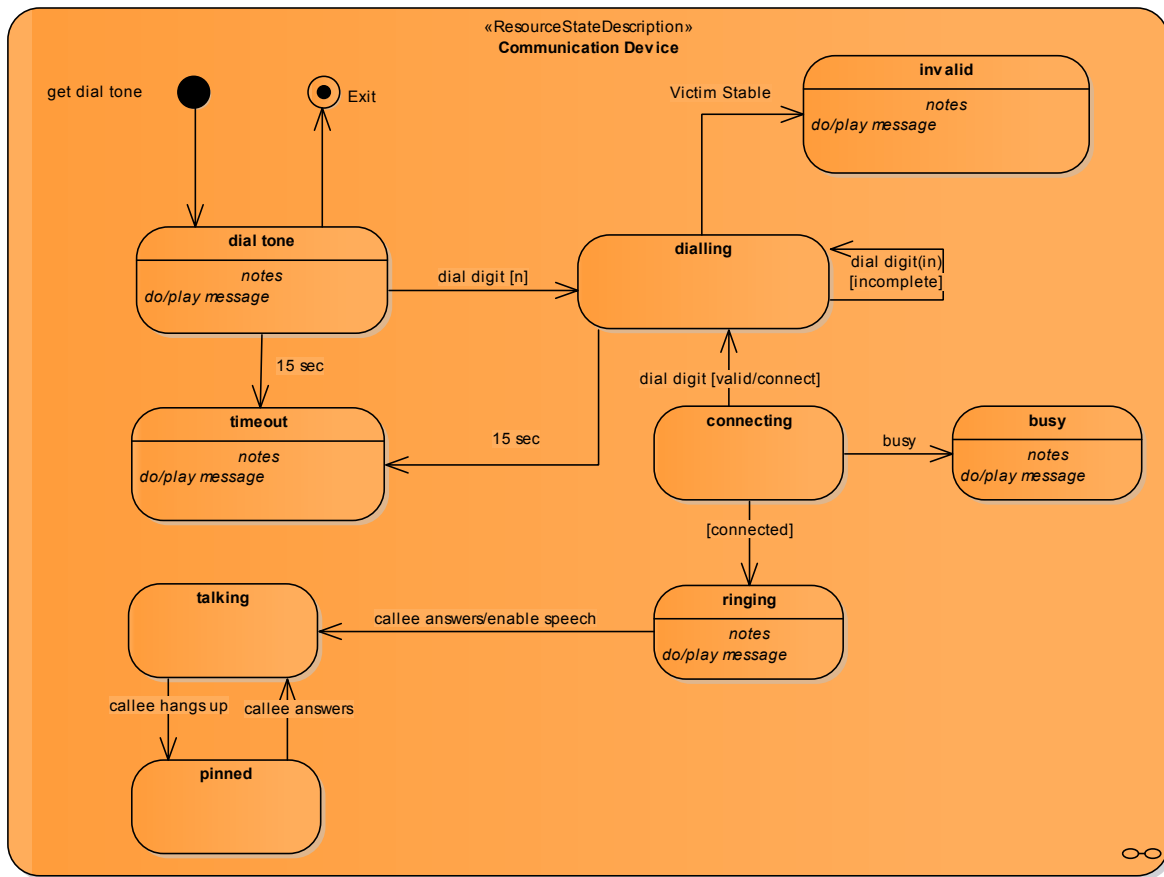
5.5.2 NAFv4 SMM



5.5.3 UAF DMM Extract



5.5.4 Example



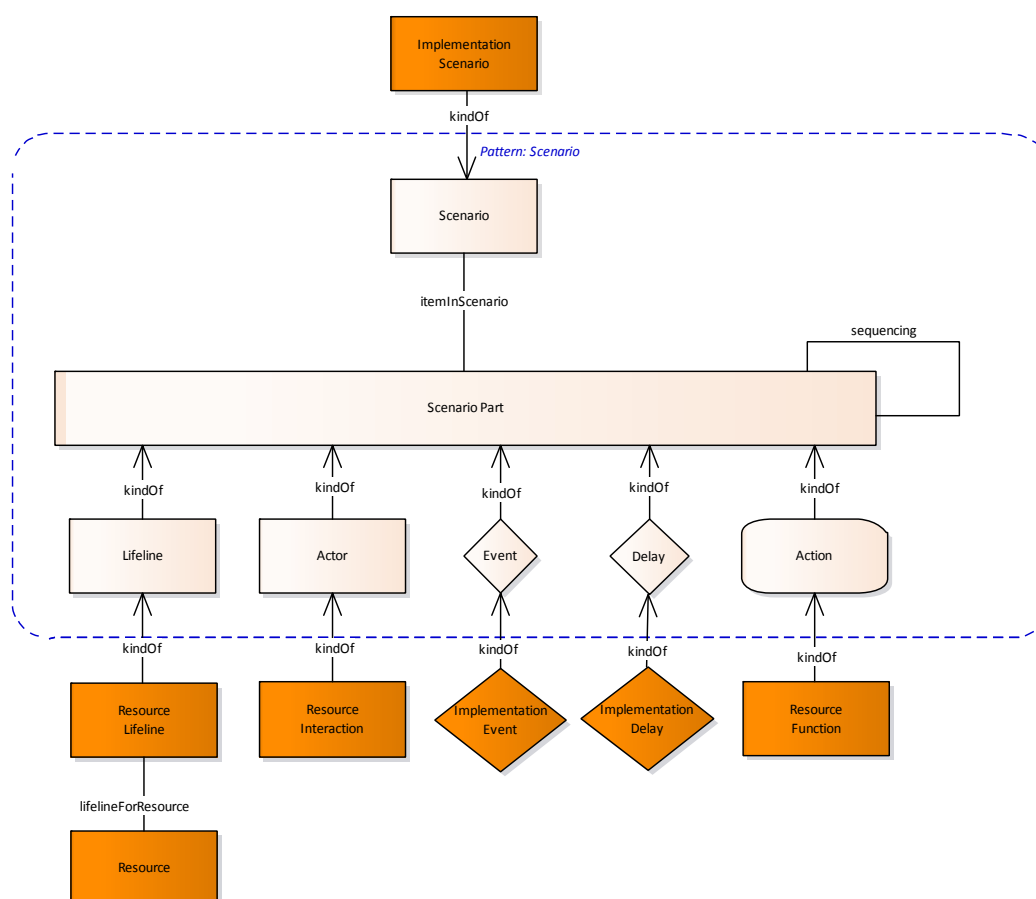
5.6 P6 – Resource Sequence

The P6 Viewpoint is valuable for moving to the next level of detail from the initial solution design, to help define a sequence of functions and Resource Interactions, and to ensure that each participating Resource or Port has the necessary information, at the right time, in order to perform its assigned functionality.

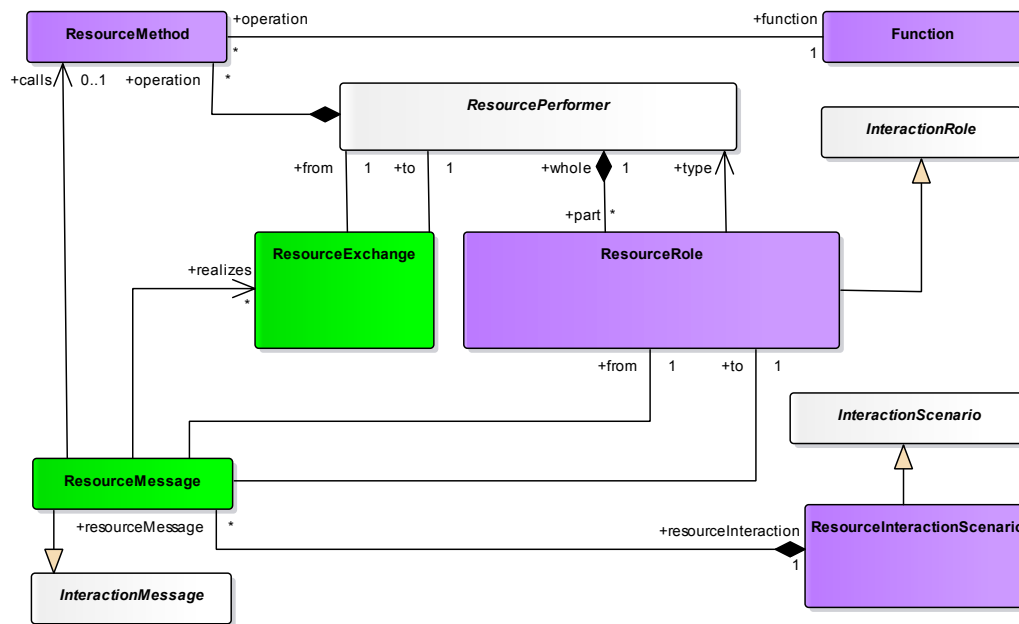
5.6.1 Concerns

- Specification of sequences in which data elements are exchanged in context of a Resource Type or Port,
- Inclusion of a time-ordered representation of the data elements exchanged between participating Resource Type or Ports,
- (optional) representation of materiel flows, human resources or energy as interactions, and
- (optional) inclusion of nonfunctional properties, such as security, performance, etc.

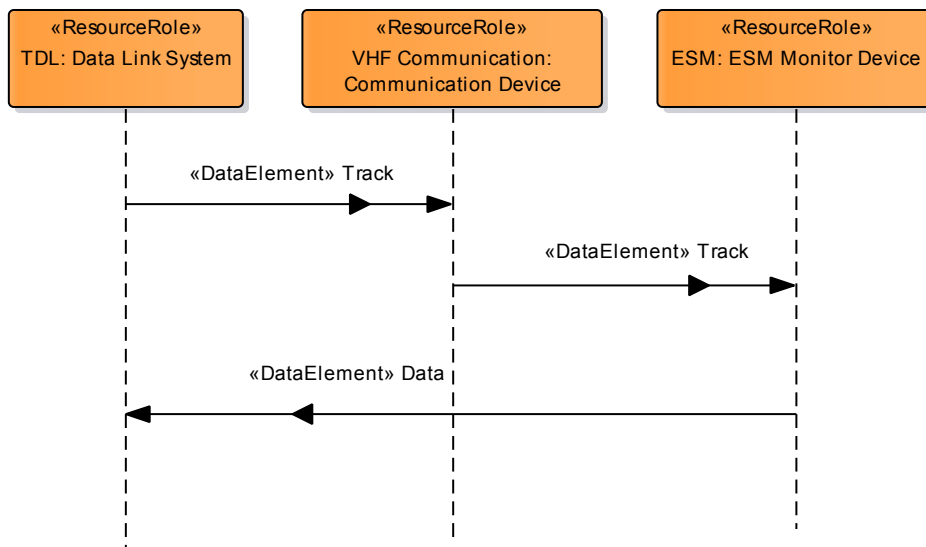
5.6.2 NAFv4 SMM



5.6.3 UAF DMM Extract



5.6.4 Example



5.7 P7 – Physical Data Model

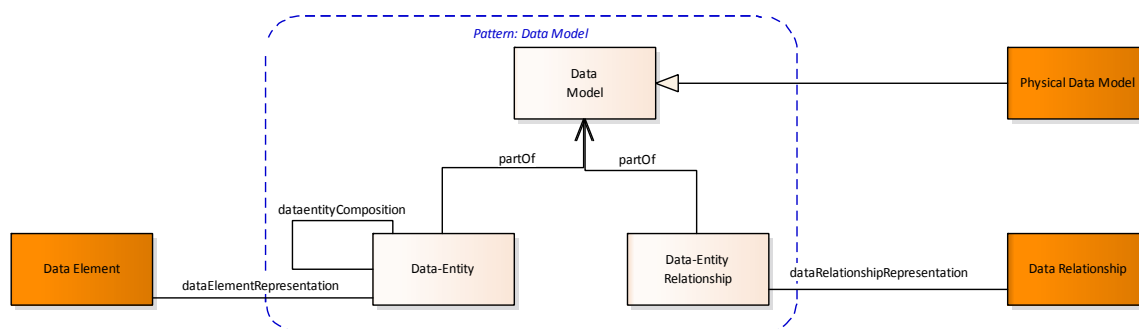
The P7 Viewpoint is one of the architectural products closest to actual system design in the NAF. It is used to describe how the information represented in the L7 Logical Data Model Viewpoint is implemented.

While the mapping between the logical and physical data models is relatively straightforward, the relationship between the components of each model (e.g. entity types in the logical model versus relational tables in the physical model) is frequently one-to-many or many-to-many.

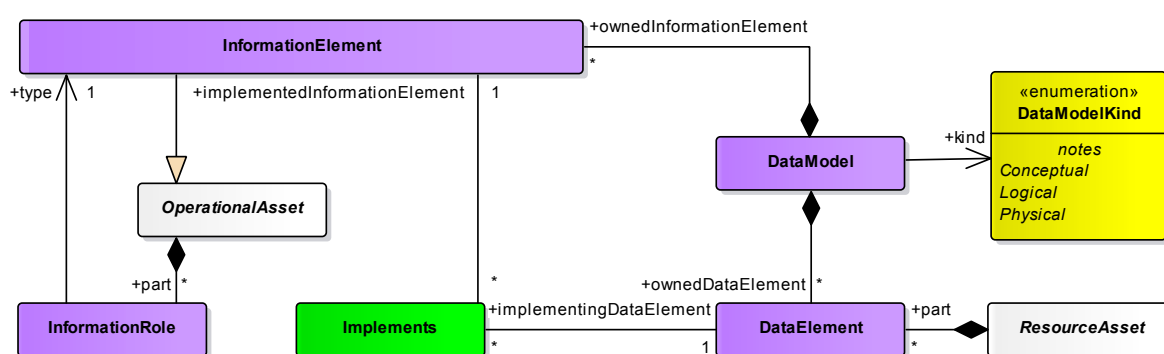
5.7.1 Concerns

- identification of data structure used by the resource types in the architecture, and
- (optional) mapping of a given logical data model (L7) to the physical data model (P7) if both models are used.

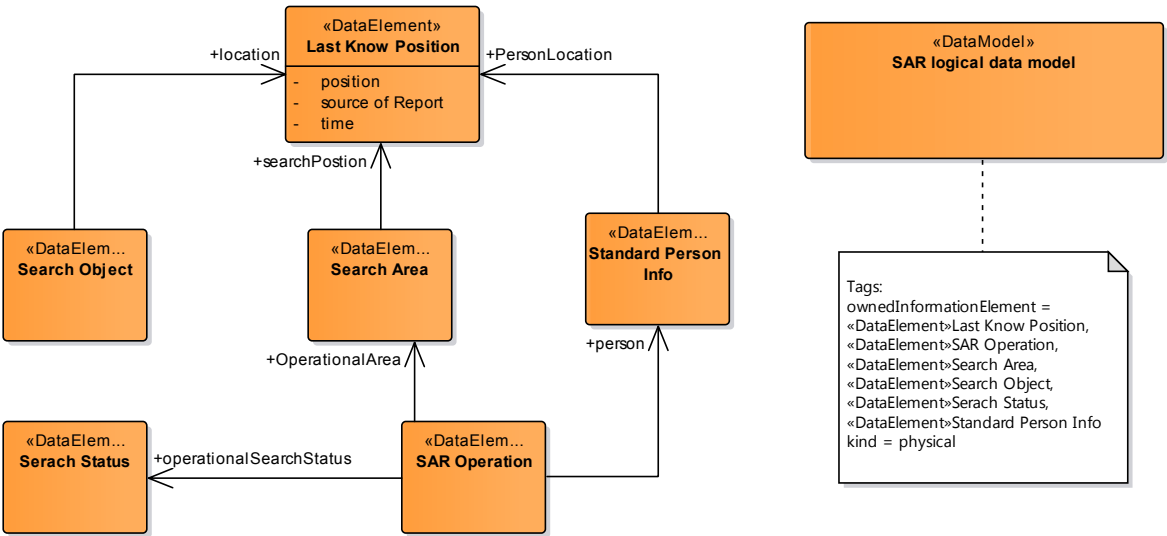
5.7.2 NAFv4 SMM



5.7.3 UAF DMM Extract



5.7.4 Example



5.9 P8 – Resource Constraints

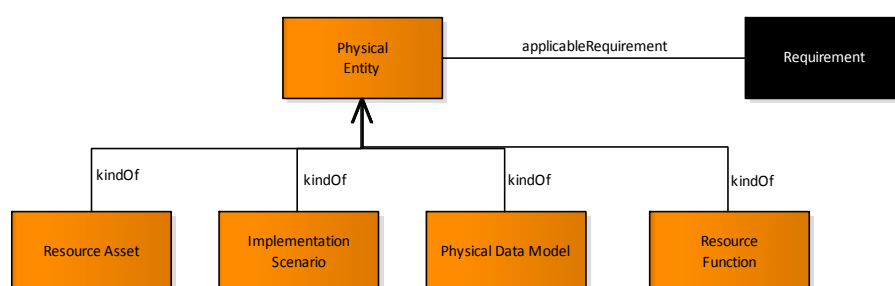
The P8 Viewpoint describes constraints on the Resources, Resource Functions, data and communications that make up a Physical Architecture.

The P8 Viewpoint describes the rules that control, constrain or otherwise guide the implementation aspects of the architecture. Resource constraints are statements that define or constrain some aspect of the technology or business, and may be applied to resources, functions, system ports or data elements.

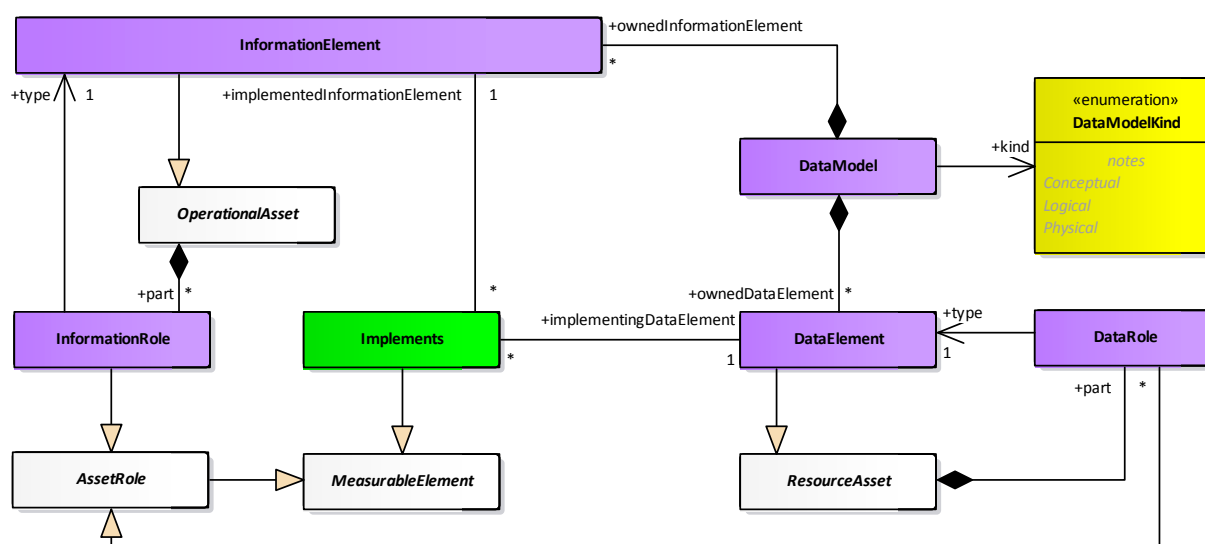
5.9.1 Concerns

- identification of functional and non-functional constraints on the implementation aspects on resource types, resource functions, data and ports, and
- identification of rules which control, constrain or otherwise guide the implementation aspects of the architecture.

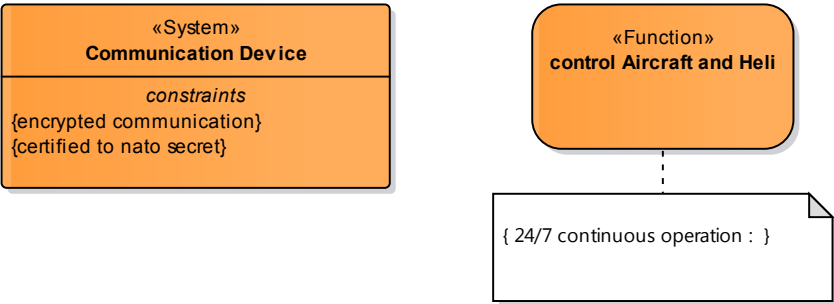
5.9.2 NAFv4 SMM



5.9.3 UAF DMM Extract



5.9.4 Example



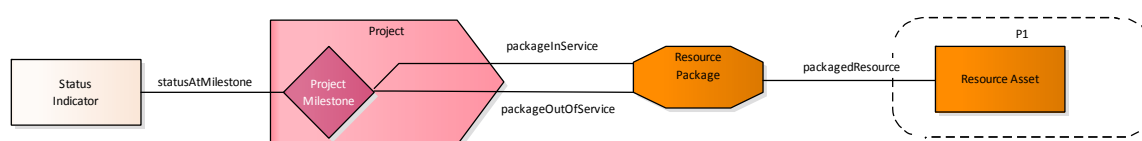
5.10 Pr – Configuration Management

The Pr Viewpoint provides an overview of how Resource Assets change over time (note that NAFv3.1 only allowed for Capability Configurations whereas now this is opened up to all Resource Types). It shows the structure of different versions of Resource Assets (usually Capability Configurations or Service Implementations) mapped against a timeline.

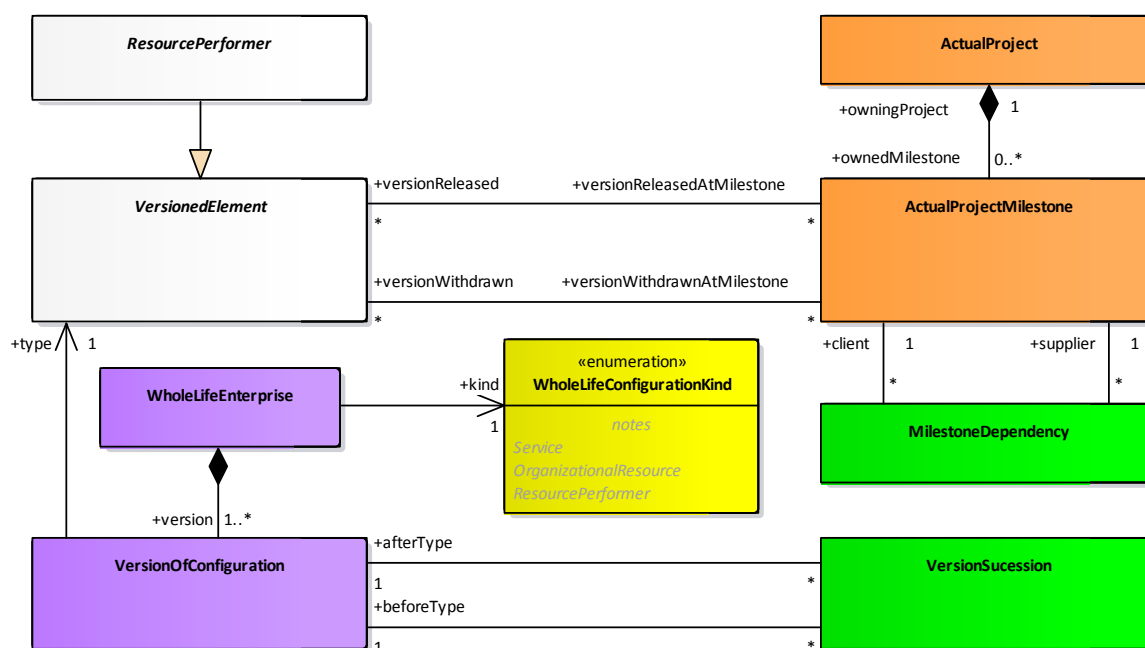
5.10.1 Concerns

- description of an overview of how a Resource Type structure changes over time, and
- inclusion of the structure of different versions of Resource Type (usually Capability Configurations or Service Implementations) mapped against a timeline.

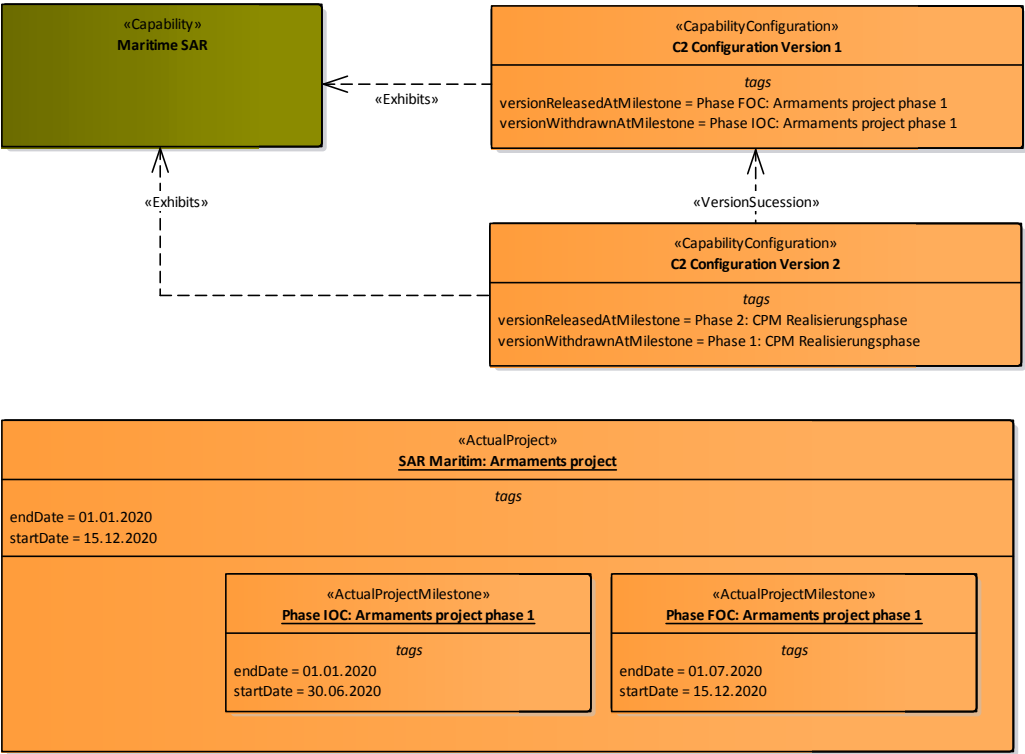
5.10.2 NAFv4 SMM



5.10.3 UAF DMM Extract



5.10.4 Example



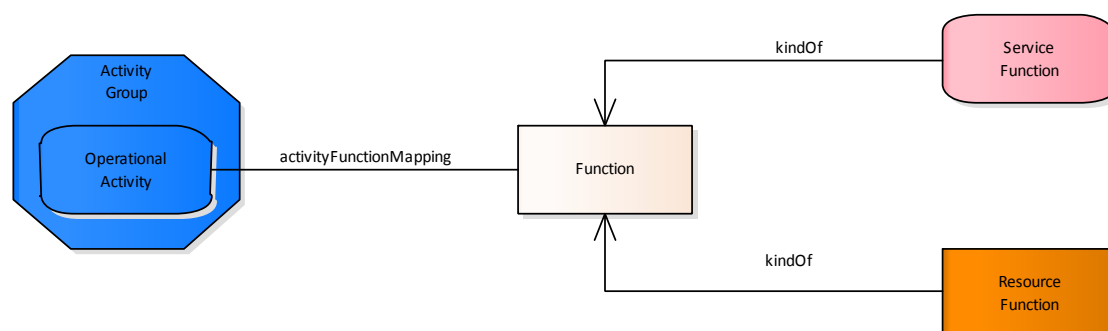
5.11 L4-P4 – Activity to Function Mapping

The L4-P4 Viewpoint depicts the mapping of Resource Functions (and optionally, the resources that provide them) to operational activities and/or service functions. For operational activities it thus identifies the transformation of an operational need into a purposeful action performed by a system or solution. For service functions it provides the link between the services used at the operational level and the specific Resource Functions provided by the resources supporting the services.

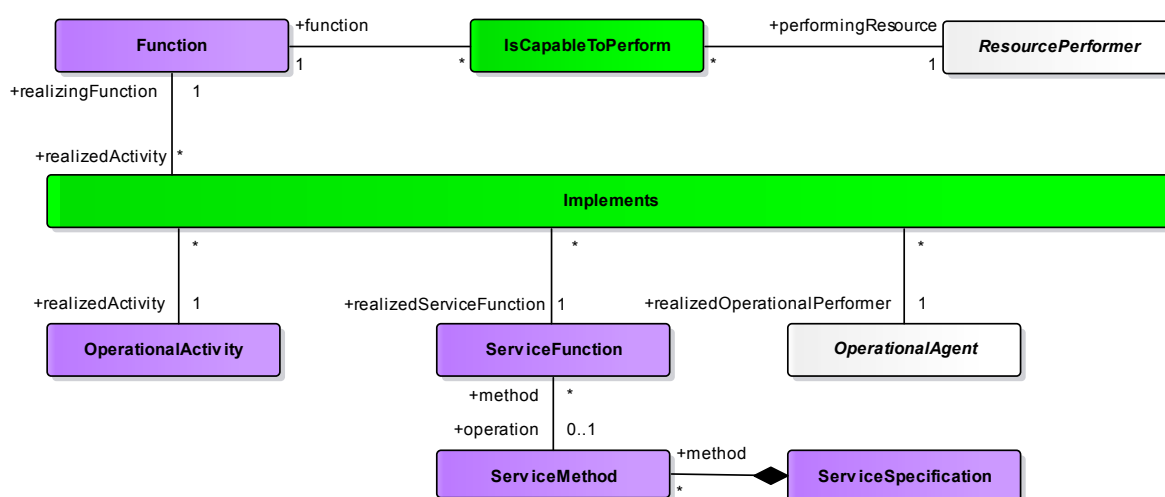
5.11.1 Concerns

- mapping of Resource Functions to operational activities or service functions,
- provision of the link between the services used at the operational level, and
- provision of the link between specific Resource Functions provided by the resources supporting the services.

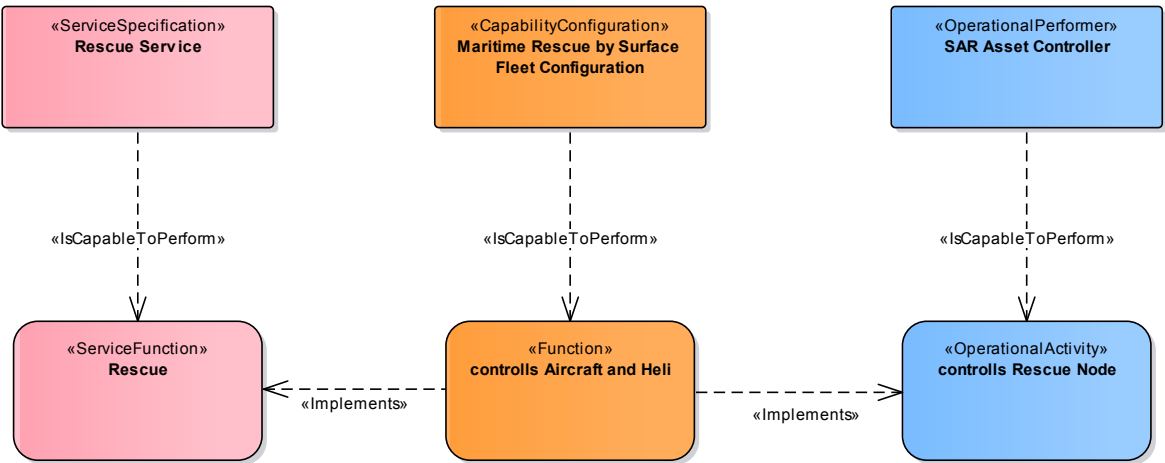
5.11.2 NAFv4 SMM



5.11.3 UAF DMM Extract



5.11.4 Example



5.12 Elements

NAME	DEFINITION
ActualEnterprisePhase	The ActualState that describes the phase of an Enterprise endeavour.
ActualMeasurement	An actual value that is applied to a Measurement.
ActualMeasurementSet	A set of ActualMeasurements.
ActualProject	A time-limited endeavour to provide a specific set of ActualResource that meet specific Capability needs.
ActualProjectMilestone	An event with a start date in an ActualProject from which progress is measured.
ActualResource	Role in an Organization, where the role carries the authority to undertake a function - through the ActualOrganizationalResource given the role has the responsibility.
ActualService	An individual ServiceSpecification.
Asset	Asset as applied to Security views, an abstract type that indicates the types of elements that can be considered as a subject for security analysis.
CapableElement	An abstract type that represents a structural element that can perform behaviours (i.e. OperationalActivity).
Capability	A high level specification of the enterprise's ability to execute a specified course of action.
CapabilityConfiguration	A composite structure representing the physical and human resources (and their interactions) in an enterprise, assembled to meet a capability).
Competence	A specific set of abilities defined by knowledge, skills and aptitude.
DataElement	A formalized representation of data that is managed by or exchanged between resources.
DataModel	A structural specification of data types, showing relationships between them that is devoid of implementation detail. The type of data captured in the DataModel is described using the enumeration DataModelKind (Conceptual, Logical and Physical).
Exhibits	A tuple that exists between a CapableElement and a Capability that it meets under specific environmental conditions.
Forecast	A tuple that specifies a transition from one Asset, Standard, Competence to another future one. It is related to an ActualEnterprisePhase to give it a temporal context.
Function	An Activity which is specified in the context to the ResourcePerformer (human or machine) that IsCapableToPerform it.
FunctionAction	A call of a Function indicating that the Function is performed by a ResourceRole in a specific context.
FunctionEdge	A tuple that shows the flow of Resources (objects/data) between FunctionActions.
GeoPoliticalExtentType	A geospatial extent whose boundaries are defined by declaration or agreement by political parties.
Implements	A tuple that defines how an element in the upper layer of abstraction is implemented by a semantically equivalent element (i.e. tracing the OperationalActivities to the Functions that implement them) in the lower level of abstraction.
InformationElement	An item of information that flows between OperationalPerformers and is produced and consumed by the OperationalActivities that the OperationalPerformers are capable to perform (see IsCapableToPerform).
InformationRole	A usage of InformationElement that exists in the context of an OperationalAsset. It also allows the representation of the whole-part aggregation of InformationElements.

InteractionMessage	An abstract type that represents a behaviour or process (i.e. a Function or OperationalActivity) that can be performed by a Performer.
InteractionRole	No definition.
InteractionScenario	An abstract type that represents a behaviour or process (i.e. a Function or OperationalActivity) that can be performed by a Performer.
IsCapableToPerform	A tuple defining the traceability between the CapableElements to the Activities that they can perform.
MeasurableElement	Abstract type, grouping elements that can be measured by applying MeasurementSets to them.
Measurement	A property of an element representing something in the physical world, expressed in amounts of a unit of measure.
MilestoneDependency	A tuple between two ActualProjectMilestones that denotes one ActualProjectMilestone follows from another.
OperationalActivity	An Activity that captures a logical process, specified independently of how the process is carried out.
OperationalAgent	An abstract type grouping LogicalArchitecture and OperationalPerformer.
OperationalArchitecture	A type used to denote a model of the Architecture, described from the Operational perspective.
OperationalAsset	No definition.
OperationalPerformer	A logical entity that IsCapableToPerform OperationalActivities which produce, consume and process Resources.
Organization	A group of OrganizationalResources (Persons, Posts, Organizations and Responsibilities) associated for a particular purpose.
OrganizationalResource	An abstract type for Organization, Person Post and Responsibility.
OperationalRole	Usage of an OperationalPerformer or OperationalArchitecture in the context of another OperationalPerformer or OperationalArchitecture. Creates a whole-part relationship.
PerformsInContext	A tuple that relates an OperationalAction to an OperationalRole, or a FunctionAction to a ResourceRole. It indicates that the action can be carried out by the role when used in a specific context or configuration.
Person	A type of a human being used to define the characteristics that need to be described for ActualPersons (e.g. properties such as address, telephone number, nationality, etc).
PhysicalResource	An abstract type defining physical resources (i.e. OrganizationalResource, ResourceArtifact and NaturalResource).
Post	A type of job title or position that a person can fill (e.g. Lawyer, Solution Architect, Machine Operator or Chief Executive Officer).
Process	An abstract type that represents a behaviour or process (i.e. a Function or OperationalActivity) that can be performed by a Performer.
ProcessEdge	An abstract type that represents a behaviour or process (i.e. a Function or OperationalActivity) that can be performed by a Performer.
ProcessParameter	An abstract type that represents a behaviour or process (i.e. a Function or OperationalActivity) that can be performed by a Performer.
ProcessUsage	An abstract type that represents a behaviour or process (i.e. a Function or OperationalActivity) that can be performed by a Performer.
PropertySet	An abstract type grouping architectural elements that can own Measurements.
PropertySetGeneralization	A PropertySetGeneralization is a taxonomic relationship between a more general PropertySet and a more specific PropertySet.
Protocol	A Standard for communication over a network. Protocols may be composite, represented as a ProtocolStack made up of ProtocolLayers.
ProtocolImplementation	An abstract type grouping architectural elements that can implement Protocols.
ProtocolLayer	Usage of a Protocol in the context of another Protocol. Creates a whole-part relationship.

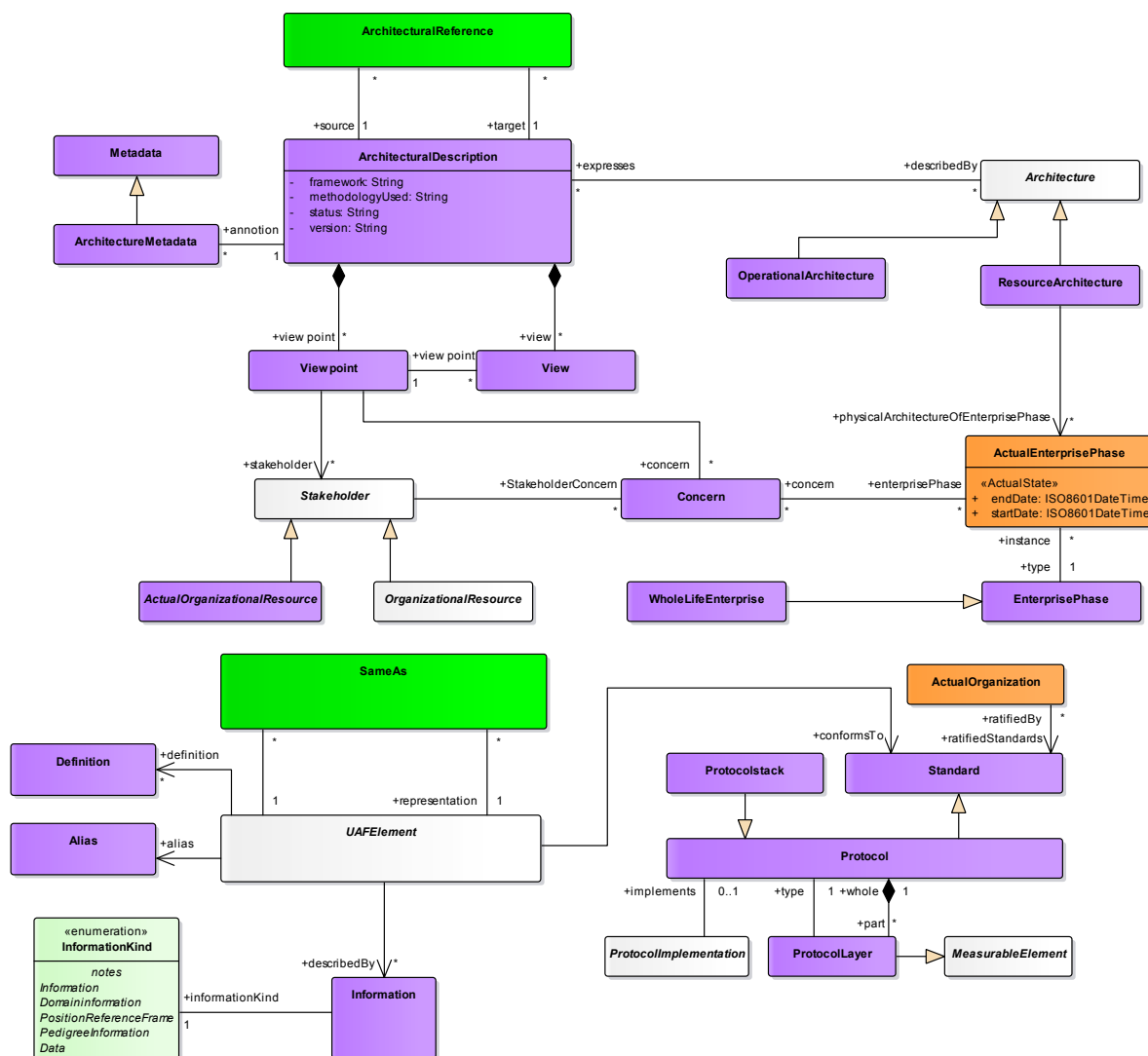
Protocolstack	A sub type of Protocol that contains the ProtocolLayers, defining a complete stack.
ProvidedServiceLevel	A sub type of ActualService that details a specific service level delivered by the provider.
RequiredServiceLevel	A sub type of ActualService that details a specific service level required of the provider.
RequiresCompetence	A tuple that asserts that an ActualOrganizationalResource is required to have a specific set of Competencies.
ResourceArchitecture	A type used to denote a model of the Architecture, described from the ResourcePerformer perspective.
ResourceAsset	No definition.
ResourceArtifact	A type of man-made object that contains no human beings (i.e. satellite, radio, petrol, gasoline, etc.).
ResourceConnetcor	A channel for exchange between two ResourceRoles.
ResourceConstraint	A rule governing the structural or functional aspects of an implementation.
ResourceExchange	Asserts that a flow can exist between ResourcePerformers (i.e. flows of data, people, material, or energy).
ResourceExchangItem	An abstract type grouping elements that defines the types of elements that can be exchanged between ResourcePerformers and conveyed by a ResourceExchange.
ResourceInteractionScenario	No definition.
ResourceInterface	A declaration that specifies a contract between the ResourcePerformers it is related to and any other ResourcePerformers it can interact with. It is also intended to be an implementation of a specification of an Interface in the Business and/or Service layer.
ResourceMessage	Message for use in a Resource Event-Trace which carries any of the subtypes of ResourceExchange.
ResourceMethod	A behavioural feature of a ResourcePerformer whose behaviour is specified in a Function.
ResourceMitigation	A set of measures intended to implement an OperationalMitigation. Comprises a subset of activities that are performed in mitigation of the risk to protect the asset that is the subject of risk (ResourceRole) at the physical level. In the case of a Risk applicable to security, the form of activity is a SecurityControl or an EnhancedSecurityControl, otherwise it is a Function.
ResourceParameter	A type that represents inputs and outputs of a Function. It is typed by a ResourceInteractionItem.
ResourcePerformer	An abstract type grouping elements that can be the subject of a SecurityConstraint.
ResourcePort	An interaction point for a ResourcePerformer through which it can interact with the outside environment and which is defined by a ResourceInterface.
ResourceRole	Usage of a ResourcePerformer in the context of another ResourcePerformer. Creates a whole-part relationship.
ResourceStateDescription	A state machine describing the behaviour of a ResourcePerformer, depicting how the ResourcePerformer responds to various events and the actions.
Responsibility	The type of duty required of a Person or Organization.
Rule	An abstract type for all types of constraint (i.e. an OperationalConstraint could detail the rules of accountancy best practice).
SecurityEnclave	Collection of information systems connected by one or more internal networks under the control of a single authority and security policy. The systems may be structured by physical proximity or by function, independent of location.
SecurityProcess	The security-related procedure that satisfies the security control requirement.

ServiceFunction	An Activity that describes the abstract behaviour of ServiceSpecifications, regardless of the actual implementation.
ServiceInterface	A contract that defines the ServiceMethods and ServiceMessageHandlers that the ServiceSpecification realizes.
ServiceMethod	A behavioural feature of a ServiceSpecification whose behaviour is specified in a ServiceFunction.
ServiceSpecification	The specification of a set of functionality provided one element for the use of others.
StateDescription	An abstract type that represents a state machine (i.e. an OperationalStateDescription or ResourceStateDescription), depicting how the Asset responds to various events and the actions.
Software	A sub-type of ResourceArtifact that specifies an executable computer programme.
SubjectOfForecast	An abstract type grouping elements that can be the subject of a Forecast.
SubjectOf ResourceConstraint	An abstract type grouping elements that can be the subject of a ResourceConstraint.
System	An integrated set of elements, subsystems, or assemblies that accomplish a defined objective. These elements include products (hardware, software, and firmware), processes, people, information, techniques, facilities, services, and other support elements (INCOSE SE Handbook V4, 2015).
Technology	A sub type of ResourceArtifact that indicates a technology domain, i.e. nuclear, mechanical, electronic, mobile telephony etc.
VersionOfConfiguration	A property of a WholeLifeConfiguration, used in version control of a VersionedElement. It asserts that a VersionedElement is a version of a WholeLifeConfiguration.
VersionedElement	An abstract type grouping ResourcePerformer and ServiceSpecification that allows VersionOfConfiguration to be related to ActualProjectMilestones.
VersionSucession	A tuple between two VersionOfConfigurations that denotes that one VersionOfConfiguration follows from another.
WholeLifeEnterprise	A WholeLifeEnterprise is a purposeful endeavour of any size involving people, organizations and supporting systems. It is made up of TemporalParts and StructuralParts.

6 ARCHITECTURE META-DATA VIEWPOINTS

	Taxonomy	Structure	Connectivity	Processes	States	Sequences	Information	Constraints	Roadmap
	A1	A2	A3	A4	A5	A6	A7	A8	Ar
Architecture Meta-Data	Meta-Data Definitions NAV-3	Architecture Products	Architecture Correspondence ISO42010	Methodology Used NAF Ch2	Architecture Status NAV-1	Architecture Versions NAV-1	Architecture Meta-Data NAV-1/3	Standards NTV-1/2	Architecture Roadmap

Viewpoints in the Architecture Meta-Data row of the NAF grid support the administrative aspects of the architecture, such as who created it, for whom and when.



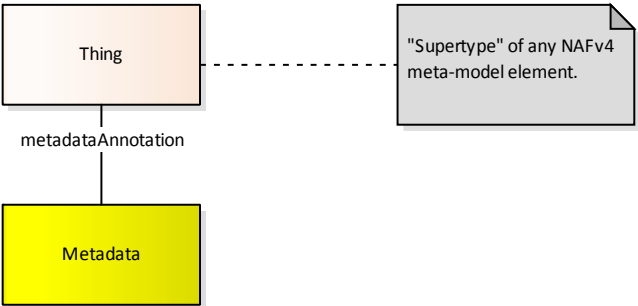
6.1 A1 – Meta-Data Definitions

Architectures, particularly large ones, usually require meta-data tags to aid with searching and discovery.

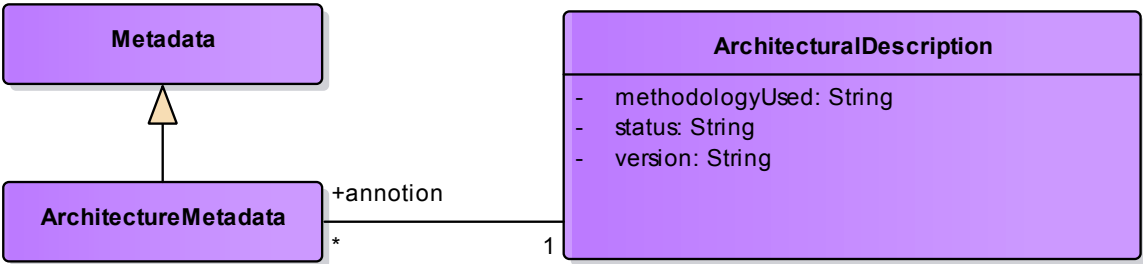
6.1.1 Concerns

- listing of all the meta-data tags used throughout the architecture,
- inclusion required meta-data tags to aid with searching and discovery, and
- (optional) list the architectural elements that are tagged.

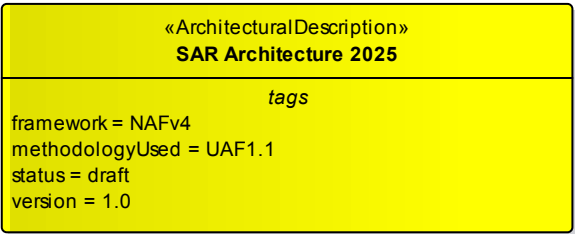
6.1.2 NAFv4 SMM



6.1.3 UAF DMM Extract



6.1.4 Example



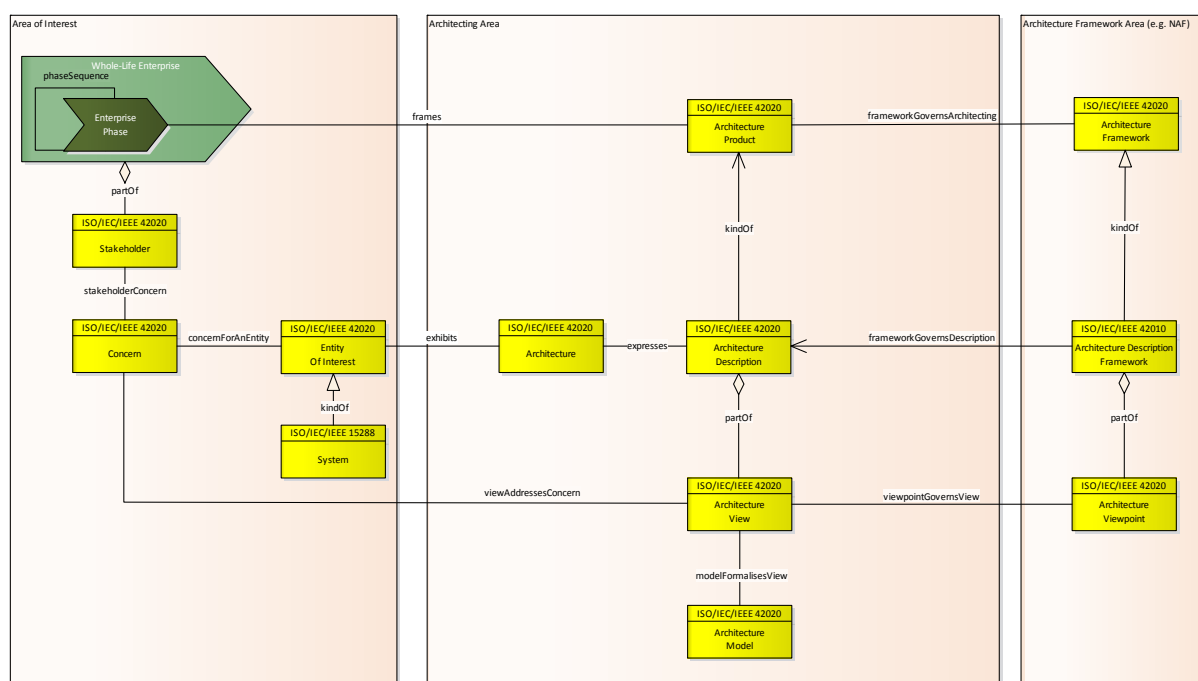
6.2 A2 – Architecture Products

The A2 Viewpoint specifies the structure of an architecture, and the products that describe the architecture. Each product may correspond to an architecture view. This viewpoint also traces the architectures onto the Enterprise Phases they correspond to (see also C2 – Enterprise Vision) and identifies the key stakeholders, their concerns and the products that address those concerns (from ISO/IEC/IEEE Standards).

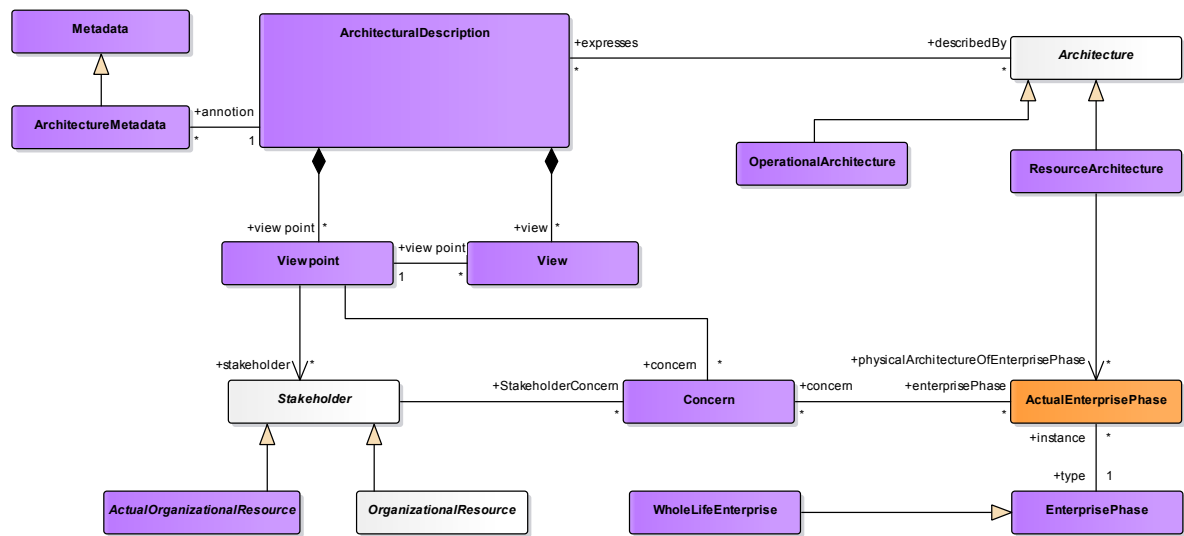
6.2.1 Concerns

- listing of Views that make up the Architecture Description and which Viewpoints those Views conform to, and
- tracing the architectures onto the Enterprise Phases they correspond to and identify the key stakeholders, their concerns and the products that address those concerns.

6.2.2 NAFv4 SMM



6.2.3 UAF DMM Extract



6.2.4 Example

«ArchitecturalDescription» SAR Architecture 2025
tags
framework = NAFv4 methodologyUsed = UAF1.1 status = final version = 1.0 view = P1 - Communication System viewpoint = P1

«View_» P1 - Communication System
tags
viewpoint = P1

«Viewpoint» P1
tags
concern = Interoperability stakeholder = ACT

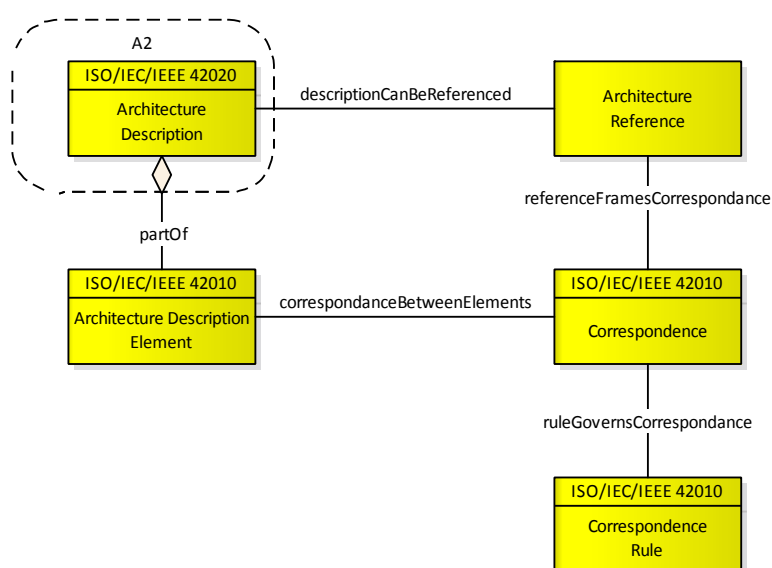
6.3 A3 – Architecture Correspondence

ISO/IEC/IEEE42010 introduces the idea of architecture correspondence and correspondence rules. Quoting from ISO/IEC/IEEE 42010; “A correspondence defines a relation between AD (Architecture Description) elements. Correspondences are used to express architecture relations of interest within an architecture description (or between architecture descriptions). Correspondences can be governed by correspondence rules. Correspondence rules are used to enforce relations within an architecture description (or between architecture descriptions).”

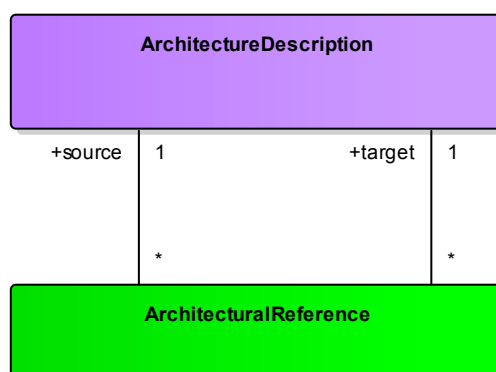
6.3.1 Concerns

- Inclusion of all relevant dependencies between architectures.

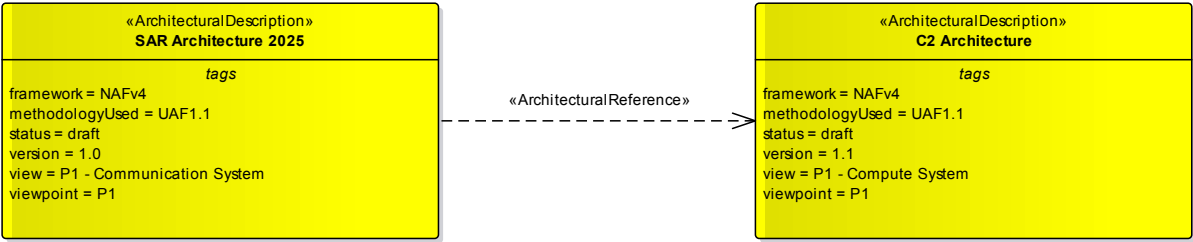
6.3.2 NAFv4 SMM



6.3.3 UAF DMM Extract



6.3.4 Example



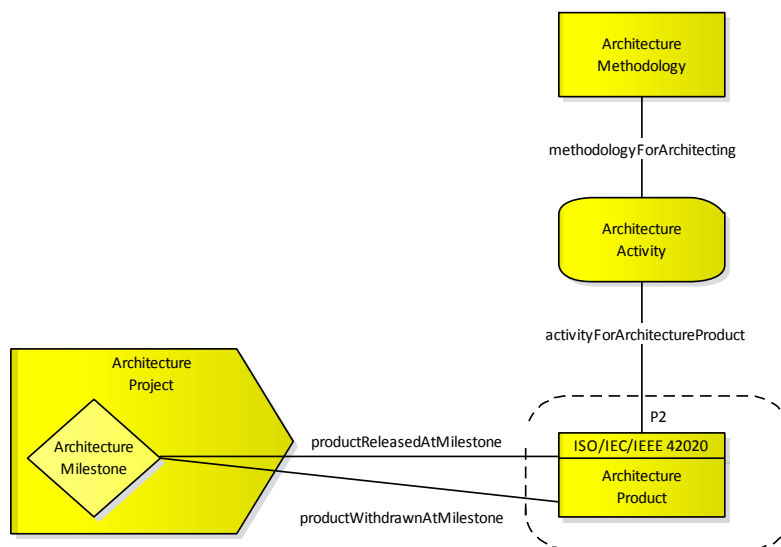
6.4 A4 – Methodology Used

The A4 Viewpoint specifies the methodology used for the architecting activities. This methodology is described in NAFv4 Chapter 2.

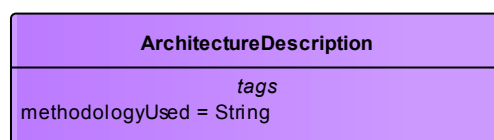
6.4.1 Concerns

- specification of the methodology used.

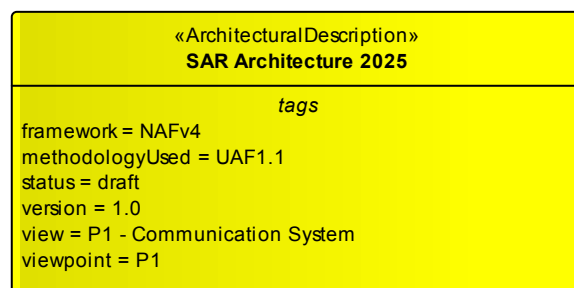
6.4.2 NAFv4 SMM



6.4.3 UAF DMM Extract



6.4.4 Example



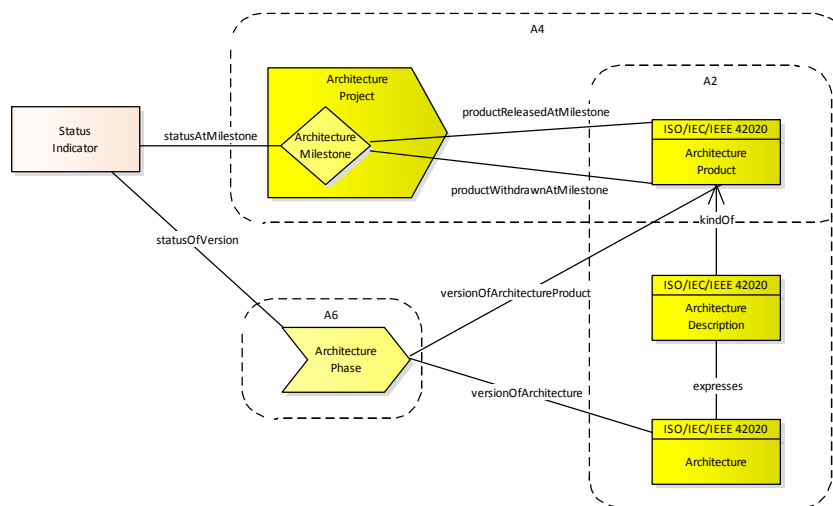
6.5 A5 – Architecture Status

Since NAFv3.0, it has been possible to assign version numbers to views and specify their approval dates. In NAFv4.0, this is integrated with the meta-model's generic project & milestone approach so that architecture releases are tracked against the architecture delivery project.

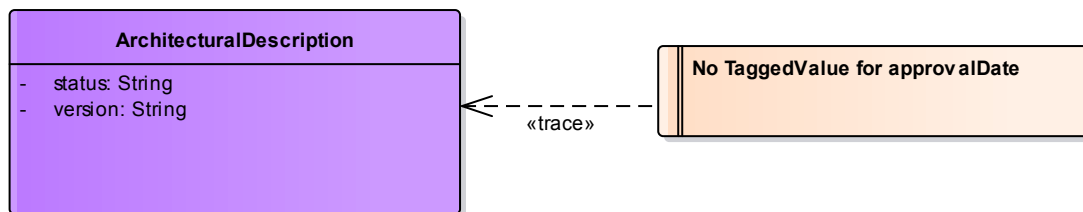
6.5.1 Concerns

- identification of version number and approval status of the architecture.

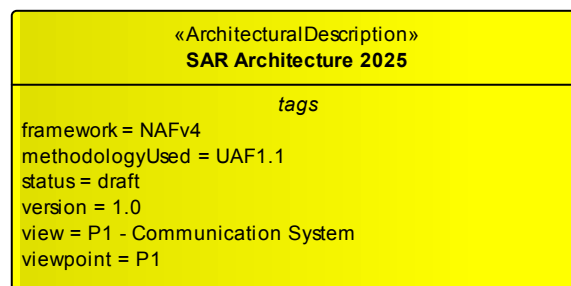
6.5.2 NAFv4 SMM



6.5.3 UAF DMM Extract



6.5.4 Example



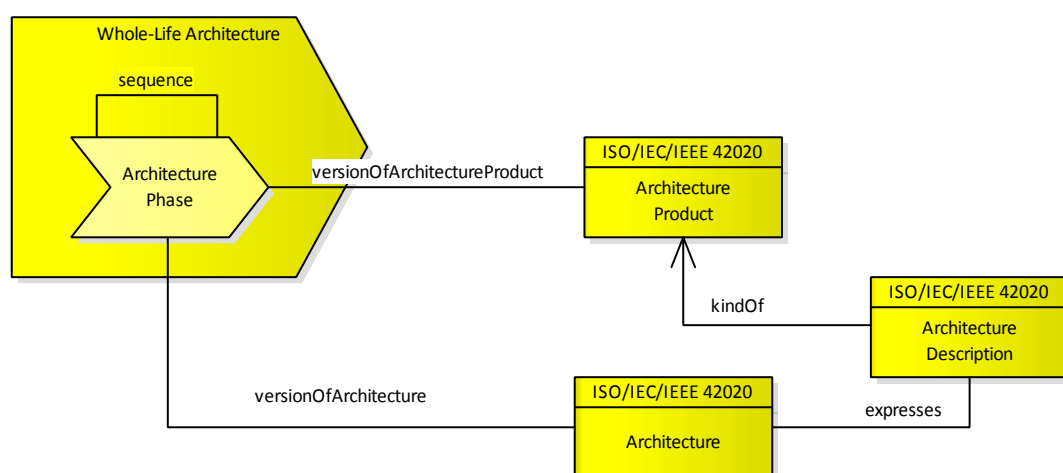
6.6 A6 – Architecture Versions

Where the A5 Viewpoint shows the current version number and approval status for the architecture, the A6 Viewpoint shows the complete history of the architecture.

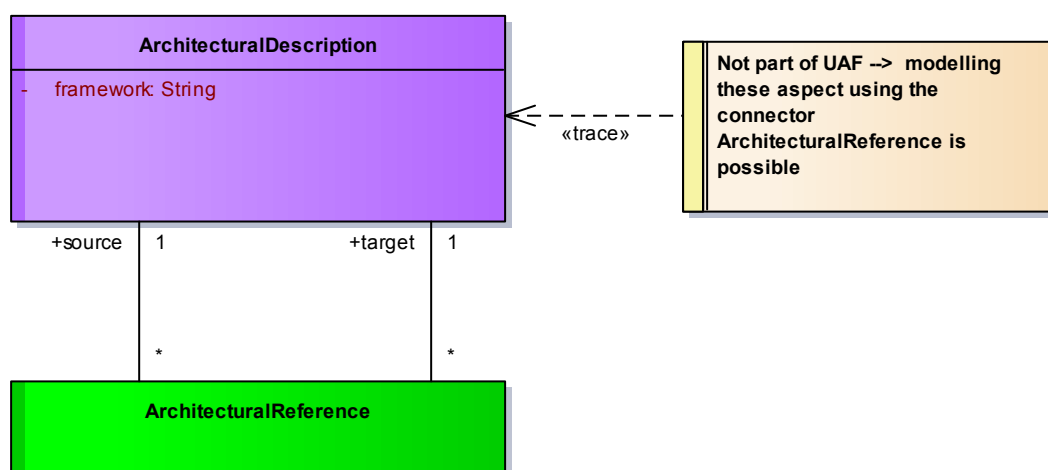
6.6.1 Concerns

- expansion of the information in the A5 by representing the catalogue of previous architecture versions, and
- reference of the framework used to create the architecture.

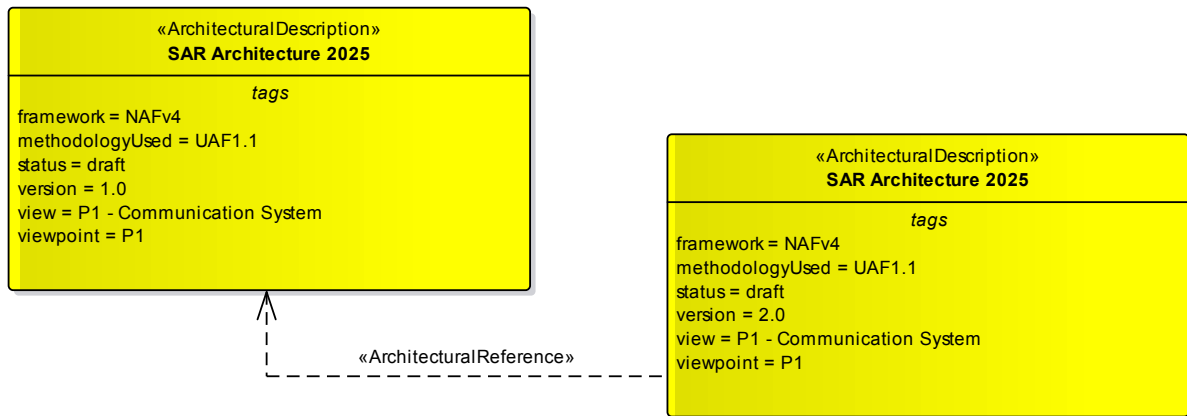
6.6.2 NAFv4 SMM



6.6.3 UAF DMM Extract



6.6.4 Example



6.7 A7 – Architecture Meta-Data

The A7 Viewpoint gives the overall specification of architecture meta-data.

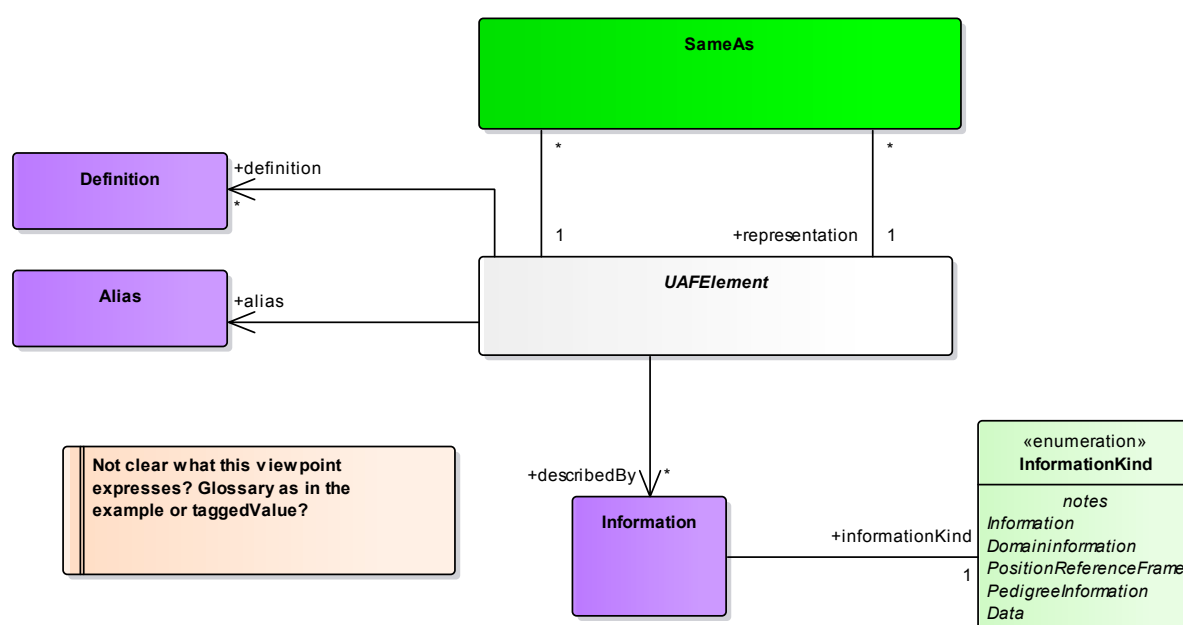
6.7.1 Concerns

- Tagging of architecture elements and Views with meta-data, and
- (optional) inclusion of definitions, assumptions, findings, recommendations, or references.

6.7.2 NAFv4 SMM

The A7 is not a view modeled as a diagram. An example for representation would be a search index or generated table (cf. Annex C).

6.7.3 UAF DMM Extract



6.7.4 Example

The A7 is not a view modeled as a diagram. An example for representation would be a search index or generated table (cf. Annex C).

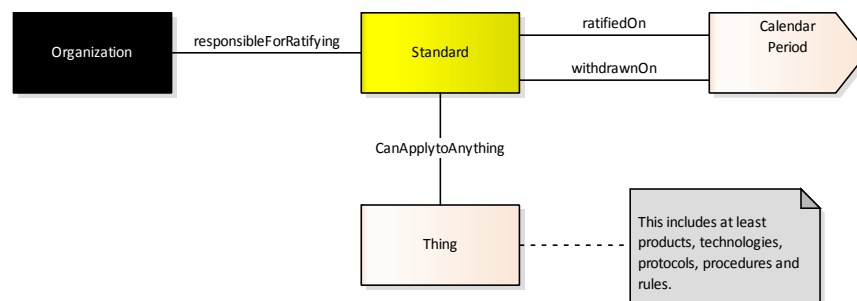
6.8 A8 – Standards

The A8 Viewpoint encompasses both technical and non-technical standards. The standards specified in the A8 view can be applied across the architecture to a variety of structural and behavioural elements. Standards are essential to the coherent running of businesses and to the delivery of reliable, interoperable systems.

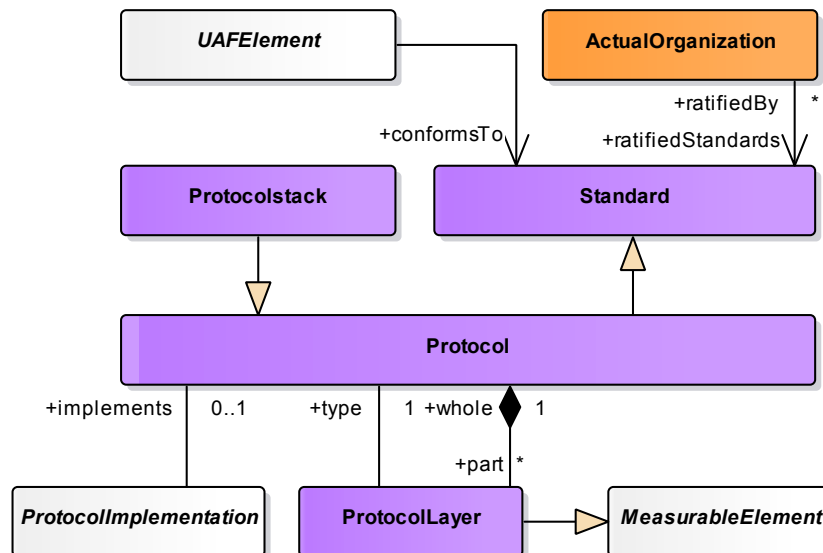
6.8.1 Concerns

- identification of all standards and protocols are used in the architecture,
- (optional) mapping of standards and protocols to the ratifying actual organization, and
- (optional) identification of the structure of protocols.

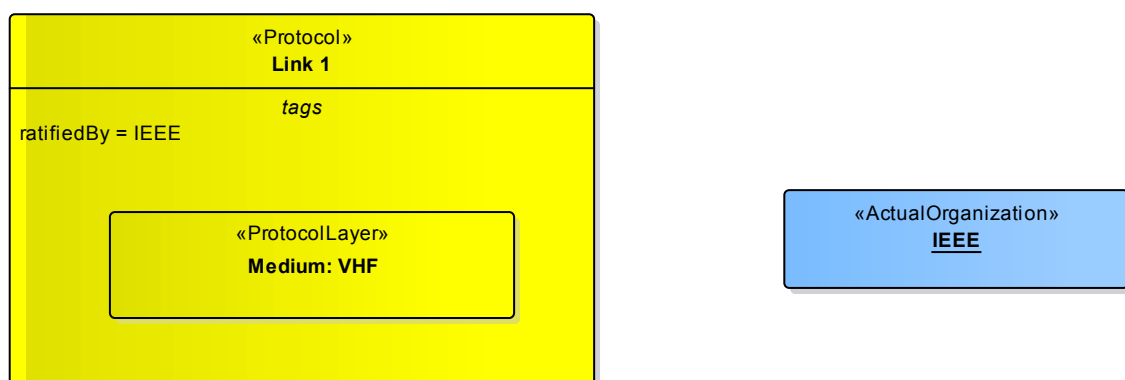
6.8.2 NAFv4 SMM



6.8.3 UAF DMM Extract



6.8.4 Example



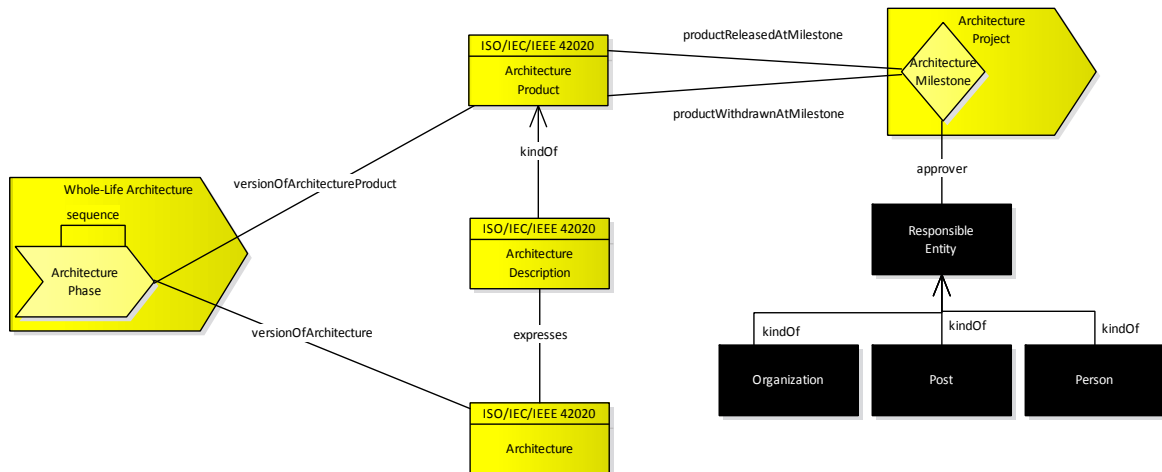
6.9 Ar – Architecture Roadmap

The Ar Viewpoint shows the history of the architecture project as well as its future direction.

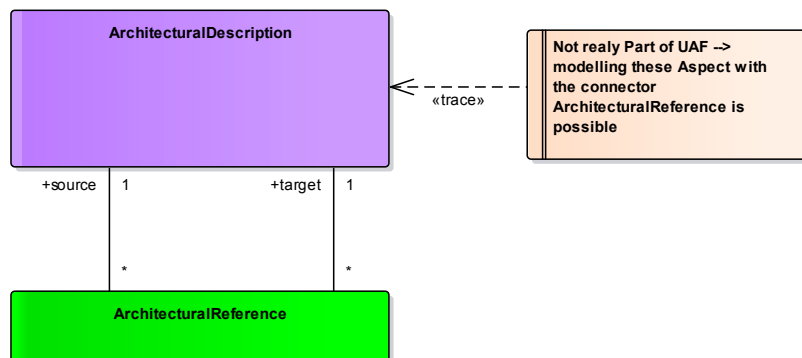
6.9.1 Concerns

- provision of detailed information regarding the architecture project from a timeline perspective.

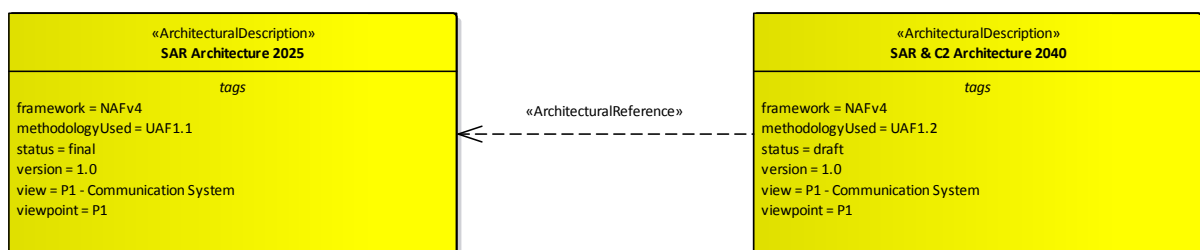
6.9.2 NAFv4 SMM



6.9.3 UAF DMM Extract



6.9.4 Example



6.10 Elements

NAME	DEFINITION
ActualEnterprisePhase	The ActualState that describes the phase of an Enterprise endeavour.
ActualOrganization	An actual formal or informal organizational unit, e.g. "Driving and Vehicle Licensing Agency", "UAF team Alpha".
ActualOrganizationalResource	Abstract element for an ActualOrganization, ActualPerson or ActualPost.
Alias	A meta-model Artifact used to define an alternative name for an element.
ArchitecturalDescription	An Architecture Description is a work product used to express the Architecture of some System Of Interest. It provides executive-level summary information about the architecture description in a consistent form to allow quick reference and comparison between architecture descriptions -- It includes assumptions, constraints, and limitations that may affect high-level decisions relating to an architecture-based work programme.
ArchitecturalReference	A tuple that specifies that one architectural description refers to another.
Architecture	An abstract type that represents a generic architecture. Subtypes are LogicalArchitecture and PhysicalArchitecture.
ArchitectureMetadata	Information associated with an ArchitecturalDescription, that supplements the standard set of tags used to summarize the Architecture. It states things like what methodology was used, notation, etc.
Concerns	Interest in an EnterprisePhase (EnterprisePhase is synonym for System in ISO 42010) relevant to one or more of its stakeholders.
Definition	A comment containing a description of an element in the architecture.
EnterprisePhase	A current or future state of the wholeLifeEnterprise or another EnterprisePhase.
MeasurableElement	Abstract type, grouping elements that can be measured by applying MeasurementSets to them.
Metadata	A comment that can be applied to any element in the architecture. The attributes associated with this element details the relationship between the element and its related dublinCoreElement, metaDataScheme, category and name. This allows the element to be referenced using the Semantic Web.
OperationalArchitecture	A type used to denote a model of the Architecture, described from the Operational perspective.
OrganizationalResource	An abstract type for Organization, Person Post and Responsibility.
Protocol	A Standard for communication over a network. Protocols may be composite, represented as a ProtocolStack made up of ProtocolLayers.
ProtocolImplementation	An abstract type grouping architectural elements that can implement Protocols.
ProtocolLayer	Usage of a Protocol in the context of another Protocol. Creates a whole-part relationship.
Protocolstack	A sub type of Protocol that contains the ProtocolLayers, defining a complete stack.
ResourceArchitecture	A type used to denote a model of the Architecture, described from the ResourcePerformer perspective.
SameAs	A tuple that asserts that two elements refer to the same real-world thing.

Standard	A ratified and peer-reviewed specification that is used to guide or constrain the architecture. A Standard may be applied to any element in the architecture.
Stakeholder	Individual, team, organization, or classes thereof, having an interest in an EnterprisePhase [ISO/IEC/IEEE 42010:2011].
UAFElement	Abstract super type for all of the UAF elements. It provides a way for all of the UAF elements to have a common set of properties.
View	An architecture view expresses the architecture of the system-of-interest in accordance with an architecture viewpoint (or simply, viewpoint). [ISO/IEC/IEEE 42010:2011(E)].
Viewpoint	An architecture viewpoint frames (to formulate or construct in a particular style or language) one or more concerns. A concern can be framed by more than one viewpoint. [ISO/IEC/IEEE 42010:2011(E)].
WholeLifeEnterprise	A WholeLifeEnterprise is a purposeful endeavour of any size involving people, organizations and supporting systems. It is made up of TemporalParts and StructuralParts.

7 BIBLIOGRAPHY

NAFv4 Main Document	https://www.nato.int/cps/en/natohq/topics_157575.htm
UAF (current version)	https://www.omg.org/spec/UAF/

Annex A

Mapping of the NAF 3.1 Elements to the UAF DMM Elements

The following table should be of assistance to architects who previously worked with the NAFv3.1. It contains concepts from NAFv3.1, their explanation and the corresponding concepts in the UAF DMM. Some cells in the UAF 1.1 column are left blank intentionally because the mapping is a work in progress and has not been finished until the release date of this document.

NAFv3.1	NOTE (NAFv3.1)	UAF 1.1
ActivityComposition	An assertion that the parent activity has the child as a part - i.e. the child activity is conducted as part of conducting the parent activity. Note: Unfortunately, UML offers two ways to do this; by composite class properties (i.e. this stereotype) and by UML::CallBehaviourAction. To prevent ambiguity, this meta-model forces both approaches to be used in parallel (SysML takes the same approach). Any ActivityComposition must be accompanied by a corresponding OperationalActivityAction. Hopefully, a future version of UML may be more coherent in this department, and this duplication can be removed.	OperationalActivityAction
ActivityMapsToCapability	Asserts that a StandardOperationalActivity is in some way part of a capability. The nature of the mapping should be specified in the name of the dependency.	MapsToCapability
ActivitySubject	Anything that is acted upon by an OperationalActivity.	Performer, Role
ActivityToFunctionMapping	Asserts that a Function (at least in part) performs or assists in the conducting of an OperationalActivity.	Implements
ActsUpon	Asserts that something (subject) is acted upon by an OperationalActivity (activity).	IsCapableToPerform, PerformsInContext
ActualCompetence	Asserts that an ActualOrganisationalResource actually has a Competence.	
ActualLocation	A location anywhere on the earth. The means of describing the location is a string (locationDescription). The information contained in that string is governed by the taxonomy reference - e.g. if the ActualLocation is a "GPS reference", the string will contain the GPS coordinates.	ActualLocation
ActualOrganisation	An actual specific organization, an instance of an organization class - e.g. "The US Department of Defense".	ActualOrganization
ActualOrganisationRelationship	A relationship between two actual specific organizations or parts of an organization. Note: the TypicalOrganisationRelationship which is realized by the ActualOrganisationRelationship is referred to via the typicalRelationship attribute.	
ActualOrganisationalResource	An instance of either an actual organization or an actual post. [ABSTRACT].	ActualResourceRelationship ActualOrganizationalResource

ActualOrganizationComposition	Relates an actual specific organization to an actual specific organizational resource that fulfils a role in that organization.	No Element
ActualPost	An actual, specific post, an instance of a PostType class - e.g. "President of the United States of America".	ActualPost
Alias	An alternative name for an element.	Alias
ArbitraryConnection	Represents a visual indication of connection used in high level operational concept diagrams. The connections are purely indicative and cannot be related to any architectural semantics.	
ArchitecturalDescription	A formal description of a system, or a detailed plan of the system at component level, to guide its implementation.	ArchitecturalDescription
ArchitecturalFramework	A tool for assisting in the description of architectures. It consists of specific views, reference models and a list of standards, architectural products and their interrelationships.	
ArchitecturalProduct	A connected and coherent set of Architectural Elements which conform to a View.	
ArchitecturalReference	Asserts that one architectural description (referrer) refers to another (referred).	ArchitecturalReference
Architecture	The structure of components, their inter-relationships, and the principles and guidelines governing their design and evolution over time.	Architecture
ArchitectureComplianceStatement	This extension of comment is intended to be used to state whether an architectural entity is compliant or not. It can be attached to any entity within the model that can accomodate a comment.	
ArchitectureMetaData	A Metadata element that applies to the whole architecture.	ArchitectureMetadata
Artefact	A type of man-made object. Examples are "car", "radio", "fuel", etc.	ResourceArtifact
AssignedProperty	A property with a value assigned.	
AsynchronousMessage	A signal which is transmitted irregularly with respect to time. Note: An asynchronous message is not guaranteed to arrive in a specific time following a request.	ServiceMessage
Attribute	A defined property of an Entity.	
Capability	A high level specification of the enterprise's ability.	Capability
CapabilityComposition	A parent-child relationship between two capabilities - i.e. the relationship indicates one capability (child) is a sub-capability of the other (parent).	
CapabilityConfiguration	A composite structure representing the physical and human resources (and their interactions) in an enterprise. A CapabilityConfiguration is a set of artefacts or an organization configured to provide a capability, and should be guided by [doctrine] which may take the form of Standard or OperationalConstraint stereotypes.	CapabilityConfiguration

CapabilityDependency	A relationship which asserts that a capability (toCapability) is dependent on another (fromCapability) capability in the context of an overall capability. Note: this dependency relates parts (i.e. properties) in a composite class diagram, therefore each dependency is in context of the parent composite class.	CapabilityDependency
CapabilityForNode	An assertion that a Node is required to have a Capability.	Exhibits
CapabilityIncrement	A ProjectMilestone that indicates the point in time at which a project is predicted to deliver or has delivered a Capability. Example: When a project reaches Initial Operating Capability (IOC) it may deliver a Capability with a given set of metrics then deliver a second Capability corresponding to the same Capability when it reaches Full Operational Capability (FOC). Both the IOC and FOC milestones would be instances of CapabilityIncrement.	
CapabilityRealisation	Asserts that a CapabilityConfiguration is capable of achieving a Capability.	Exhibits
CapabilitySpecialisation	Asserts that one Capability is a special case of the other.	
Climate	A type of weather condition, or combination of weather conditions (e.g. high temperature & dry).	
Commands	Asserts that one OrganisationalResource (source) commands another (target).	Commands
Competence	A specific set of abilities defined by knowledge, skills and attitude.	Competence
CompetenceForRole	Asserts that a Role requires a Competence.	RequiresCompetence
ConceptDescription	A textual representation of a HighLevelOperationalConcept.	CompetenceForRole
Concern	An interest in a subject held by one or more stakeholder OrganisationalResource.	Concern
ConfigurationDeployed	Asserts that an ActualOrganisationResource started to use, or is slated to start using a CapabilityConfiguration from a specific point in time. This is used to describe capabilities going into service with specific organizations or posts.	
ConfigurationNoLongerUsed	Asserts that an ActualOrganisationResource ceased to use or is slated to cease using a CapabilityConfiguration from a specific point in time. This is used to describe capabilities going out of service with specific organizations or posts.	
ConformsTo	Asserts that an element in the architecture conforms to a Standard.	
ConsumedBy	Asserts that a service is consumed by a node. It is not required to know what provides the service.	Consumes
Controls	A type of ResourceInteraction where one Resource (source) controls another (target) Examples - the driver of a tank, one organization having operational control of another, a fire control system controlling a weapons system.	Controls
Copy	No description.	
DataElement	A formalized representation of data which is managed by or exchanged between systems.	DataElement

DataModel	A structural specification of data, showing classifications of data elements and relationships between them. [ABSTRACT].	DataModel
Definition	A definition of an element in the architecture. Note - every element added by an architect must have a definition.	Definition
DeriveReq	No description.	
Dimension	No description.	
EnduringTask	A type of behaviour recognized by an enterprise as being essential to achieving its goals - i.e. a strategic specification of what the enterprise does.	EnduringTask
Energy	A representation of any kind of energy.	
EnergyFlow	No description.	OperationalConnector
EnterpriseGoal	A specific, required objective of the enterprise that the architecture represents. Note: Benefits of achieving the goal are presented as a list of textual items.	EnterpriseGoal
EnterprisePhase	A current or future state of a WholeLifeEnterprise or another EnterprisePhase.	EnterprisePhase
EnterpriseStructure	Asserts that one EnterprisePhase is a spatial part of another. Note: This is a topological structuring relationship, hence the EnterprisePhase may be physically disjoint.	
EnterpriseTemporalPart	Asserts that one EnterprisePhase is a temporal part of another. Note: This means that both EnterprisePhases have the same spatial extent - i.e. this is only a temporal structure.	
EnterpriseVision	The overall aims of an enterprise over a given period of time.	EnterpriseVision
EntityNAF	A definition (type) of an item of interest.	
EntityRelationship	Asserts that there is a relationship between two entities.	
Environment	A definition of the conditions in which something exists or functions. An Environment may be specified in terms of LocationType (e.g. terrain), Climate (e.g. tropical), and LightCondition (e.g. dark, light, dusk, etc.).	Environment
EnvironmentalConditions	Asserts that a Capability's capabilityMetric (MeasureableProperty) is valid for a particular environment. Example - a capability with a rate of advance of 40 km per day must be qualified by the environment for which this is specified - e.g. desert conditions.	Condition
EnvironmentalProperty	Asserts that an Environment has one or more properties. These may be Climate, LocationType, or LightCondition.	EnvironmentalProperty
EnvironmentalPropertyType	No description.	
ExternalIndividual	An individual (i.e. something which has spatial and temporal extent) defined by an external ontology.	
ExternalType	A type defined by an external ontology. Note: this may be higher-order - i.e. a type of a type.	

FieldedCapability	An actual, fully-realized capability. A FieldedCapability must indicate its configuration CapabilityConfiguration. Example: "HMS Iron Duke, configured and crewed, operating under the appropriate doctrine". Note - the CapabilityConfiguration that this realizes would specify a UK Type 23 Frigate, the crew, the weapons systems, etc.	FieldedCapability
Forecast	A statement about the future state of one or more types of system or standard. Note, this is an EffectivityConstrainedItem - i.e. the forecast is effective for a given period.	Forecast
FrequencyRange	A MeasureableProperty that specifies maximum and minimum frequencies, measured in Hertz as real numbers.	
Function	An activity which is specified in context of the resource (human or machine) that performs it. Note 1: Contrast with OperationalActivity, where the actor performing the activity is not known (i.e. it is just a logical node). A Function is implementation-specific. Note 2: Should the Function be specific to one usage of a type of system, then the usageContext is specified by a reference to the composite structure property ResourceComposition typed by the system.	Function
FunctionAction	Used to relate a Function to its sub-functions. Note 1: A FunctionAction will be created for every Function to provide a way to manage sub-functions, and to allow flows between functions. Note 2: See also FunctionComposition. Note 3: Also provides a means for attaching information (properties) to a Function.	FunctionAction
FunctionAtomicAction	No description.	FunctionAction
FunctionComposition	An assertion that the parent function has the child as a part - i.e. the child function is conducted as part of conducting the parent function. Note: Unfortunately, UML offers two ways to do this; by composite class properties (i.e. this stereotype) and by UML::CallBehaviourAction. To prevent ambiguity, this meta-model forces both approaches to be used in parallel (SysML takes the same approach). Any FunctionComposition must be accompanied by a corresponding FunctionAction. Hopefully, a future version of UML may be more coherent in this department, and this duplication can be removed.	FunctionAction
FunctionFlow	An UML::ObjectFlow between Functions.	FunctionEdge
FunctionInputPin	A port for flows that feed into a function.	
FunctionOutputPin	A port for flows that leave a function.	
FunctionProvision	Asserts that a FunctionalResource provides a Function.	IsCapableToPerform, ResourceMethod
FunctionsUpon	Asserts that a Function has some effect on a DataElement.	

HighLevelOperationalConcept	A generalized model for operations. Note: a background image may be associated with the HLOC, which is referred to by the backgroundImageURL attribute. Scaling information is also provided about the image, so that when an ItemInConcept is shown in the diagram, it can be properly located and scaled. No units are specified, but the same length unit shall be used throughout a single product.	HighlevelOperationalConcept
HostedSoftware	Asserts that Software is hosted on an Artefact (which means the artefact is some kind of computer system).	ResourceRole
HumanResource	The role of a person or organization in a PhysicalArchitecture.	ResourceRole
ISO8601DateTime	A date and time specified in the ISO8601 date-time format including timezone designator (TZD): YYYY-MM-DDThh:mm:ssTZD So, 7:20pm and 30 seconds on 30th July 2005 in the CET timezone would be represented as "2005-07-30T19:20:30+01:00". The date time string is represented by the value attribute of UML::LiteralString.	
ImplementsDataModel	No description.	Implements
ImplementsProtocol	An assertion that a ProtocolImplementation implements a Protocol.	Implements
InformationElement	A formalized representation of information, subject to an operational process.	InformationElement
InformationExchange	The collection of information elements that are exchanged between two nodes.	
InformationExchangeMessage	A message representing the exchange of information defined by an InformationExchange.	
InformationModel	An ArchitecturalProduct that describes information elements and their relationships in a coherent manner.	
InstanceInConcept	Asserts that a FieldedCapability is in a HighLevelOperationalConcept. Note: The position and URL tagged values are inherited. Note: The relationship to ConceptItem should not be set.	
ItemInConcept	A relationship which asserts that a ConceptItem forms part of the high level operational concept.	
KnownResource	Asserts that a known Resource plays a part in the architecture. Note: An OV-2 is meant to show logical interactions between nodes. However, sometimes it is known.	KnownResource
LightCondition	A specification of environmental lighting conditions. Examples would be daylight, dusk, night, moonlight, artificial	
LocationType	A general specification of the surroundings / scenario in which an operation may take place. Examples would be: "desert", "arctic", "at sea", etc.	Location
LogicalArchitecture	A CompositeStructureModel whose parts are either Nodes or ProblemDomainss.	OperationalArchitecture
LogicalDataModel	A LogicalDataModel is a specification of business information requirements as a formal data structure, where relationships and classes (entities) are used to specify the logic which underpins the information.	DataModel

LogicalFlow	Asserts that a flow exists or is required between Nodes (e.g. flows of information, people, materiel, or energy). A LogicalFlow itself may be part of a ProblemDomain – i.e. there may be options for implementing the LogicalFlow which could be outlined in alternative SV suites. If so, this is asserted using the inDomain tagged value.	OperationalConnector
MaterielFlow	No description.	OperationalConnector
Matrix	An ArchitecturalProduct that presents information in a tabular form.	
MeasurableProperty	An AssignedProperty of something in the physical world, expressed in amounts of a unit of measure. The property may have a required value - either specified by the [defaultValue] from UML::property attribute, or the [minValue] and [maxValue] to specify a required range.	
MessageHandler	An aspect of a ServiceInterfaceDefinition which receives incoming AsynchronousMessages.	
MetaData	Annotation that can be applied to any element in the architecture. Note: wherever possible, standard Meta-Data types should be used - e.g. conforming to Dublin Core.	Metadata
MileStoneCapConfRelati onship	A relationship between ProjectMilestone and CapabilityConfiguration.	
MilestoneInProject	Asserts that a ProjectMilestone belongs to a project. A milestone shall not belong to more than one project.	
MilestoneRelationship	A relationship between two milestones.	MilestoneDependency
Mission	A purpose to which a person, organization or autonomous system is tasked.	
MovementOfPeople	No description.	OperationalConnector
Needline	The collection of information exchanges that are bundled together to form a needline.	
NestedConnectorEnd	To maintain compatibility with the SysML standard (www.sysml.org), this meta-model re-uses as many of the SysML stereotypes as possible. [ABSTRACT].	
Network	A network is a specialization of system and is used for communication networks that have a geographical extension.	ResourceRole
Node	A representation of a NodeType within a specific context.	OperationalRole
NodeEnvironment	A specification of the Environment in which the node operates or is required to operate.	
NodeHasBehaviour	Asserts that an OperationalActivity is conducted at a Node. Should the same type of node be used in different contexts in the architecture, and the activity is conducted under only one of the contexts, that context is provided by the [nodeUsageContext] property.	1. IsCapableToPerform 2. OperationalMethod 3. PerformsInContext
NodeParent	The abstract supertype of all elements that can have child Nodes (LogicalArchitecture, ProblemDomain & Node).	
NodeRealisation	An assertion that a CapabilityConfiguration provides the functionality specified by an operational node.	Implements

NodeType	A logical entity that performs operational activities. Note: nodes are specified independently of any physical realisation.	OperationalPerformer
Ontology	Formal representation of a set of concepts and their relationships within a domain. Most of them describe individuals (instances), classes (concepts), attributes, and relations.	
OpActivityInputPin	A port for flows that feed into an activity.	
OpActivityOutputPin	A port for flows that leave an activity.	
OperationalActivity	A logical process, specified independently of how the process is carried out. Note: an OperationalActivity may only be carried out by a logical Node.	OperationalActivity
OperationalActivityAction	Used to relate an OperationalActivity to its sub-activities. Note 1: An OperationalActivityAction will be created for every OperationalActivity to provide a way to manage sub-activities, and to allow flows between activities. Note 2: See also ActivityComposition Note 3: Also provides a means for attaching information (properties) to an activity.	OperationalActivityAction
OperationalActivityFlow	A flow of information, energy or materiel from one activity to another.	OperationalActivityEdge
OperationalAtomicAction	An OperationalAtomicAction is a basic process step in an OperationalAction. The execution of an OperationalAtomicAction represents some transformation or processing in the modeled element, e.g. a Logical Node.	OperationalActivityAction
OperationalConstraint	A rule governing an operational behaviour or property.	OperationalConstraint
OperationalDeterminationNode	An OperationalDeterminationNode directs a incoming OperationActivityFlowItem (InformationElement, OrganisationalResource, ResourcesWithMaterialContent or Energy) to one or a set of outgoing edges. Which of the edges is actually traversed depends on the evaluation of the guards on the outgoing edge.	
OperationalExchangeMessage	A message representing the exchange of items defined by a LogicalFlow.	OperationalMessage
OperationalFinalNode	An OperationalFinalNode is a control node at which flows ends when the OperationalAction is finished.	
OperationalForkNode	An OperationalForkNode directs a incoming OperationActivityFlowItem (InformationElement, OrganisationalResource, ResourcesWithMaterialContent or Energy) to its outgoing edges.	
OperationalInitialNode	An OperationalInitialNode is a control node at which flows starts when the OperationalAction is invoked.	
OperationalInteractionSpecification	A specification of the interactions between nodes in an operational architecture.	OperationalInteractionScenario
OperationalJoinNode	An OperationalJoinNode synchronizes all its incoming OperationActivityFlowItems (InformationElement, OrganisationalResource, ResourcesWithMaterialContent or Energy) to one outgoing edge.	

OperationalMergeNode	An OperationalMergeNode directs one or a set of incoming OperationActivityFlowItems (InformationElement, OrganisationalResource, ResourcesWithMaterialContent or Energy) to a outgoing edge.	
OperationalNodeLifeline	A lifeline which represents a usage of a node in an operational architecture.	InteractionRole
OperationalStateDescription	A rule governing an operational behaviour or property.	OperationalStateDescription
OperationalSwimlane	A visual representation of nodes which conduct activities, shown as "swimlanes".	
OrgResourceReference	A reference to an ActualPost or ActualOrganisation.	
OrganisationProjectRelationship	A relationship between an ActualOrganisation and a Project. Example: ownership Example: supplier.	
OrganisationType	A group of persons, associated for a particular purpose.	Organization
OrganisationalResource	Either an organization, or a post. [ABSTRACT]	OrganizationalResource
OutOfService	A ProjectMilestone that indicates a project's deliverable is to go out of service.	
Part	Usage of an Artefact as a part of another Artefact.	ResourceRole
PhysicalArchitecture	A configuration of Resources for a purpose.	
PhysicalAsset	Usage of an Artefact as a component of a ResourceConfiguration.	ResourceAsset
PhysicalDataModel	A PhysicalDataModel is an implementable specification of a data structure. A PhysicalDataModel realizes a LogicalDataModel, taking into account implementation restrictions and performance issues whilst still enforcing the constraints, relationships and typing of the logical model.	DataModel
Platform	Usage of an Artefact as a platform (e.g. vessel, aircraft, etc.) in a particular ResourceConfiguration.	ResourceRole
Post	Asserts that a post exists in an OrganisationType of the type specified by the related PostType. Note: posts in organizations may or may not be filled.	PostRole
PostType	A type of point of contact or responsible person. Note that this is the type of post - e.g. Desk Officer, Commander Land Component, etc.	Post
ProblemDomain	The boundary containing those Nodes which may be realized by functional resources specified in SV-1. There may be more than one alternative solution for a given ProblemDomain specified as a set of SV suites. There may be only one ProblemDomain in a LogicalArchitecture.	ProblemDomain
ProcessOwner	Asserts that an OrganizationalResource has responsibility for an OperationalActivity. Note this does not imply the resource conducts the activity, merely that it has managerial responsibility for it.	
ProductOfView	Asserts that an ArchitecturalProduct conforms to a View specification.	
Project	A time-limited endeavour to create a specific set of products or services.	ActualProject

ProjectAggregationship	An aggregation between to projects.	
ProjectMilestone	An event in a Project by which progress is measured - modelled as a Project of zero duration. Note: in the case of an acquisition project, there are two key types of milestone which shall be represented using subtypes - CapabilityIncrement and OutOfService.	ProjectMilestone
ProjectOwnership	A type of OrganisationProjectRelationship where the organization is the party responsible for the project.	
ProjectRelationship	A relationship between two projects.	
ProjectSequence	Asserts that one Project follows from another - i.e. the target Project cannot start until the source Project has ended.	ProjectSequence
ProjectTheme	An aspect by which the progress of various Projects may be measured.	ProjectTheme
ProjectType	A category of Project Example: "Programme" Example: "Acquisition Project" Example: "Training Programme".	Project
ProjectWholePart	Relates a parent project (owningProject) to a sub-project (relatedProject).	
Protocol	A Standard for communication. Protocols may be composite (i.e. a stack).	Protocol
ProtocolImplementation	An element that can implement a Protocol	ProtocolImplementation
ProtocolStack	Asserts that a Protocol (upperLayer) uses another Protocol (lowerLayer).	Protocolstack
Provides	Asserts that a node provides a service.	
QualitativeProperty	An AssignedProperty whose value is a text literal (string).	
RadioFrequencyPortConnector	A SystemPortConnector that connects two ports which are typed as RadioFrequencyPort.	
RatificationBody	Asserts than an ActualOrganisation is responsible for the ratification of a standard.	
ReferredLocation	Either an actual location, or a type of location (i.e. environment) at/in which operations may be conducted. [ABSTRACT]	LocationHolder
RelatedProjectReference	A reference to a sub-project from a ProjectWholePart relationship.	
RequiredNodeLocation	Relates a node to a location to assert that the operational node is required to be situated at that location.	
Requirement	To provide some level of compatibility with SysML, this meta-model copies the Requirement stereotype definition.	Requirement
ResourceConfiguration	No description.	
ResourceConstraint	A rule governing the structural or functional aspects of an implementation - this may also include constraints on OrganisationalResources that are part of an implementation.	ResourceConstraint
ResourceExchangeMessage	A message representing the exchange of items defined by a ResourceInteraction.	ResourceMessage
ResourceInteraction	An assertion that two FunctionalResources interact. Examples: data exchange between systems, conversations between people, people using systems.	ResourceExchange

ResourceInteractionSpecification	A specification of the interactions between aspects of a Resources architecture.	ResourceInteractionScenario
ResourceLifeLine	A UML::Lifeline that represents a ResourceLifelineItem that interacts with another ResourceLifelineItem.	InteractionRole
ResourceLifelineItem	An element that may be represented as a ResourceLifeLine in a ResourceInteractionSpecification. [ABSTRACT].	
ResourcePartition	A swimlane representing a usage of a FunctionalResource.	
ResourceStateMachine	A state transition model which represents the behaviour of a Resource or Function.	ResourceStateDescription
ResourceType	A PhysicalAsset, OrganisationalResource or FunctionalResource that can contribute towards fulfilling a capability [ABSTRACT].	ResourcePerformer
ResourceUsage	A relationship between Resources that asserts one resource is part of the other (i.e. composition). The relationship is abstract, and one of its subtypes should be used to describe *how* one resource is part of another.	ResourceRole
ResourcesWithMaterialContent	An abstract element that contains all resource types that do not only contain organizational resources.	PhysicalResource
Role	No description.	
RoleType	An aspect of a person or organization that enables them to fulfil a particular function.	
SameAs	Asserts that two elements refer to the same real-world thing.	SameAs
Satisfy	No description.	
Service	A type of delivered functionality, specified independently of the capabilities that provide it. Note: A service may or may not have a physical effect on its environment OASIS Definition: A service is a mechanism to enable access to a set of one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description.	ServiceSpecification
ServiceAimsToAchieve	Asserts that a Service is intended to deliver a Capability Note: multiple instantiations of this element may be required, as it is likely that more than one service is required to achieve a capability.	Exhibits
ServiceAttribute	A property of Service. Example: availability.	
ServiceComposition	An assertion that a Service (parentService) calls upon another Service (childService) in order to deliver its stated functionality.	ServiceSpecificationRole
ServiceConnectorEnd	One of two ends of a ServiceNeedline	
ServiceConsumer	A UML::Actor representing an unknown service user.	
ServiceFunction	A type of activity describing the functionality of a service.	ServiceFunction
ServiceFunctionToFunctionMapping	Asserts that a ServiceFunction is implemented by a Function.	Implements
ServiceGeneralisation	An assertion that one Service class is a specialization of another.	ServiceSpecificationGeneralization

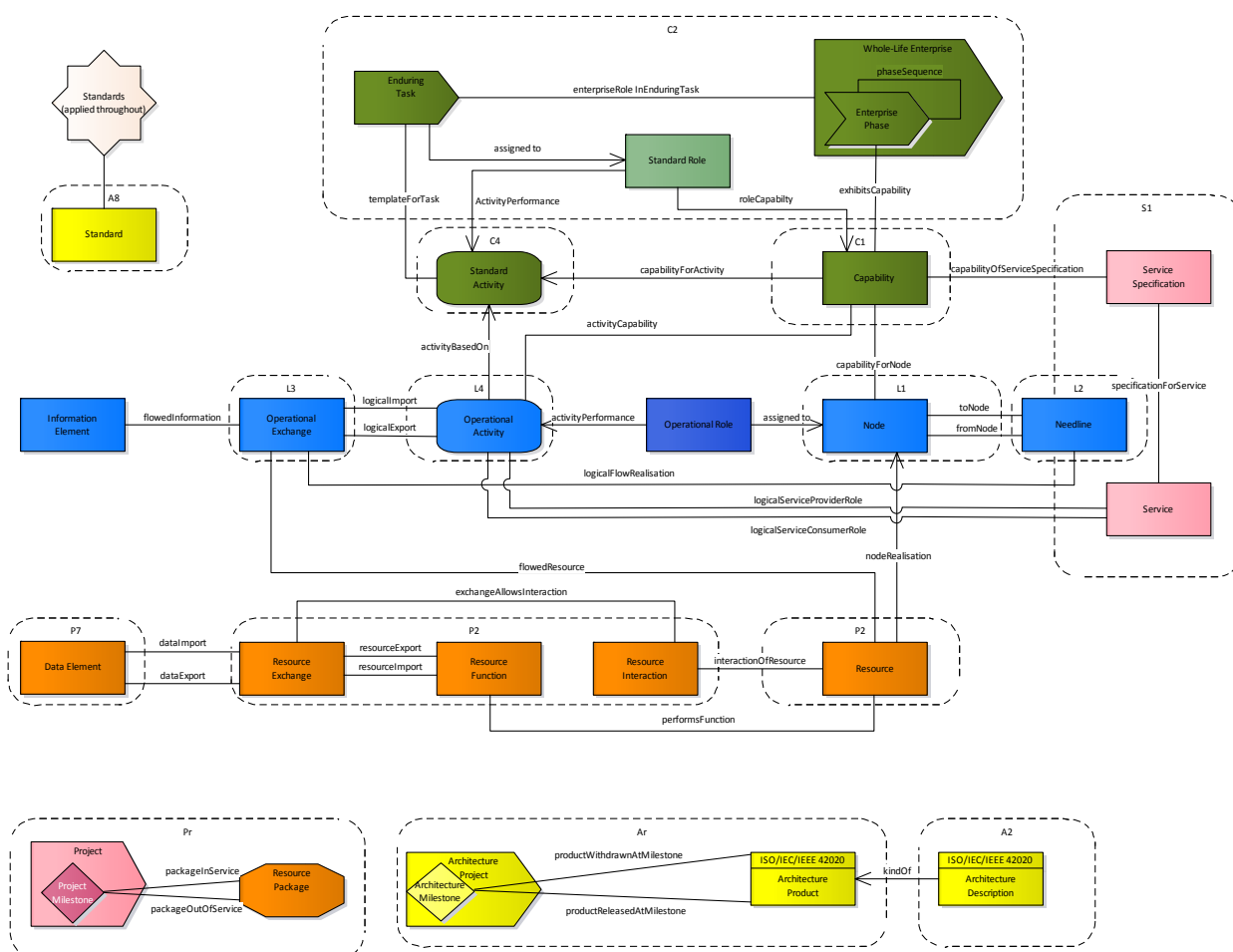
ServiceInteractionSpecification	A model representing how a set of Service classes interacts with one another.	ServiceInteractionSpecification
ServiceInterface	The mechanism by which a Service communicates. Note: a ServiceInterface specifies the ServiceInterfaceDefinition provided and required by the Service.	ServicePort
ServiceInterfaceDefinition	The specification of a provided or required communication method exposed by a ServiceInterface.	ServiceInterface
ServiceInterfaceOperation	A function or procedure which enables programmatic communication with a Service via a ServiceInterface.	ServiceMethod
ServiceInterfaceParameter	A constant or variable passed into or out of a ServiceInterface as part of the execution of a ServiceInterfaceOperation.	ServiceParameter
ServiceInterfaceSchema	An assertion that a PhysicalDataModel defines the data structure used by a ServiceInterface when communicating with a Service or client.	
ServiceLevel	A value specification for a set of ServiceAttributes indicating the level to which a Resource delivers a Service, in a particular environment. Example: A ServiceAttribute "availability" may be defined against a Service. A given Resource could have a corresponding ServiceLevel - e.g. "90%".	1.ActualService, 2.ProvidedServiceLevel, 3.RequiredServiceLevel
ServiceLifeLine	A part of a ServiceInteractionSpecification denoting the role of a ServiceInterface.	ServiceSpecificationRole
ServiceNeedline	An assertion that two Services need to communicate when assembled together under another Service.	
ServicePolicy	A constraint governing one or more Services.	ServicePolicy
ServiceProvision	An assertion that a Resource delivers a Service to a specified ServiceLevel.	
ServiceStateMachine	A model representing the changes of state which are possible for a Service.	ServiceStateDescription
ServiceSupportsActivity	An assertion that a Service in some way contributes or assists in the execution of an OperationalActivity.	Consumes
Software	An executable computer programme	Software
SoftwareComponent	Asserts that Software is a component of another Software Example - Java classes forming part of a Java applet Example - multiple Pascal units in a Pascal programme.	ResourceRole
SoftwarePort	An interface (logical or physical) provided by a System. A SystemPort may implement a PortType, though there is no requirement for SystemPorts to be typed.	ResourcePort
SpectrumAllocation	A Standard specifying a particular frequency range of the electromagnetic spectrum that is allotted to a particular usage.	
StakeholderHasConcern	An assertion that an OrganisationalResource has a Concern.	
Standard	A ratified and peer-reviewed specification that is used to guide or constrain the architecture. A Standard may be applied to any element in the architecture via the [constrainedItem] property of UML::Constraint.	Standard

StandardConfiguration	A UML::Comment that when attached to a CapabilityConfiguration indicates that it is a standard pattern for re-use in the architecture.	
StandardOperationalActivity	An OperationalActivity that is a standard procedure that is doctrinal. Note: This is equivalent to what some defence organizations call JETLs.	StandardOperationalActivity
Status	An allowable value for a StatusIndicator Example - 3 Example - amber	
StatusAtMilestone	A relationship between a Status and a milestone which asserts the status (i.e. level of progress) of a ProjectTheme for the project at the time of the milestone. For example, a procurement project may have workstreams corresponding to lines of development. The status of each of workstream is summarised on the milestone.	
StatusIndicators	An enumeration of the possible statuses for one or more ProjectThemes. Example - 1 to 5 Example - red, amber, green.	StatusIndicators
StatusLiteral	A literal value corresponding to a Status.	
StereotypeExtension	Defines an additional stereotype used in the architecture which is not defined in this meta-model. The body attribute contains the name of the new stereotype. The extendedStereotype tagged value shall contain the name of the meta-model stereotype which is extended. The ontologyReference tagged value shall be populated with a reference to the external ontology element represented by the new stereotype. Note: this is effectively a short-hand method for representing ontology items in the architecture. New stereotype names can be created at will by the architect, provided that they reference an element in a recognised external ontology. Note: Any stereotypes added by the architect which do not have a corresponding StereotypeExtension will be deemed non-compliant and ignored by tools importing data compliant to this meta-model.	
SubOrganisation	Asserts that one type of organization is typically the parent of another - e.g. a squadron may be part of a battalion.	SubOrganization
SubtypeRelationship	Asserts that one Entity (subtype) is a specialization of the other (supertype).	
System	The usage of an artefact as a System in a CapabilityConfiguration.	ResourceRole
SystemPort	An interface (logical or physical) provided by a System. A SystemPort may implement a PortType, though there is no requirement for SystemPorts to be typed.	ResourcePart
SystemPortConnector	Asserts that a connection exists between two ports belonging to parts in a system composite structure model.	ResourceConnector
SystemPortConnectorEnd	The end of a connector between system ports.	
SystemStructureModel	A CompositeStructureModel whose parts are Systems.	

TestCase	No description.	
TextProduct	An ArchitecturalProduct that is represented as text.	
Trace	No description.	
Unit	No description.	
UsedConfiguration	The usage of a CapabilityConfiguration in another CapabilityConfiguration.	ResourceRole
UsedPhysicalArchitecture	The usage of a PhysicalArchitecture in another PhysicalArchitecture or in a CapabilityConfiguration.	ResourceRole
UsesService	Asserts that a Resource requires a Service to be provided in order to function correctly.	
ValueType	No description.	
Verify	No description.	
VersionOfConfiguration	Asserts that a CapabilityConfiguration is a version of a WholeLifeConfiguration.	VersionOfConfiguration
View	A specification of a way to present an aspect of the architecture. Views are defined with one or more purposes in mind - e.g. showing the logical topology of the enterprise, describing a process model, defining a data model, etc.	View
VisionStatement	A high-level textual description of a EnterpriseVision. Note: VisionStatement is a stereotype of UML::Comment and the [body] of the comment shall be represented as XHTML. If plain text is required, then no HTML tags should be embedded.	
WholeLifeConfiguration	A set of versions of a Capability Configuration over time. WholeLifeConfiguration is used to collect together successive versions of CapabilityConfigurations from the first design to the last.	
WholeLifeEnterprise	An EnterprisePhase that represents the whole lifecycle of an enterprise.	WholeLifeEnterprise

NAFv4 SMM – One Pager

This diagram provides the key-concepts of the NAFv4 simplified meta-model.



Annex C

NAFv4 SMM Elements

	Element Name	Stereotype	Notes (Definitions)
1.	Achiever	«Entity»	Entity who performs with a high or specified level of success.
2.	Activity Group	«Entity»	A collection of activities held by an organization that is subject to governance and management by the organization as a whole.
3.	Architecting Area	«Area»	Environment where the Architecture related activities take place.
4.	Architecture	«Referenced Entity»	Fundamental concepts or properties of an entity in its environment and governing principles for the realization and evolution of this entity and its related life cycle processes [SOURCE: ISO/IEC/IEEE 42020.FDIS 2019].
5.	Architecture Activity	«Behaviour»	Any activity relating to using or providing an architecture.
6.	Architecture Description	«Referenced Entity»	A work product used to express an architecture [SOURCE: ISO/IEC/IEEE 42020.FDIS 2019].
7.	Architecture Methodology	«Entity»	A system of practices, techniques, procedures, and rules used by those who work on architecture (Adapted from PMBOK(R) Guide, Fifth Edition).
8.	Architecture Milestone	«Event»	A significant stage, phase or event in the life cycle of an architecture.
9.	Architecture Model	«Referenced Entity»	An abstracted representation of an entity or collection of entities that provides the ability to portray, understand or predict the properties or characteristics of the entity or collection under conditions or situations of interest [SOURCE: ISO/IEC/IEEE 42020.FDIS 2019].
10.	Architecture Product	«entity»	An artifact associated with the execution of an architecture activity [Adapted from ISO/IEC/IEEE 42020.FDIS 2019].
11.	Architecture Project	«Plan»	An endeavour with defined start and finish criteria undertaken to develop architecture products in accordance with specified resources and requirements [Adapted from ISO/IEC/IEEE 15288:2015].
12.	Architecture Reference	«Entity»	An expression that unambiguously identifies an architecture [Adapted from IEEE 1320.1-1998].
13.	Architecture View	«Referenced Entity»	Information item expressing the architecture from the perspective of specific stakeholders regarding specific aspects of the architecture entity and its environment [SOURCE: ISO/IEC/IEEE 42020.FDIS 2019].

14.	Architecture Viewpoint	«Referenced Entity»	Conventions for the construction, interpretation and use of architecture views to address specific concerns about the architecture entity [SOURCE: ISO/IEC/IEEE 42020.FDIS 2019].
15.	Architecture Framework	«entity»	Conventions, principles and practices for use by <i>architecture</i> -related activities that have been established within a specific domain of application or community of <i>stakeholders</i> [ISO/IEC/IEEE 42020.FDIS 2019].
16.	Architecture Phase	«Phase»	Period of time in the life cycle of an architecture during which activities are performed that enable achievement of objectives for that phase [Adapted from ISO/IEC/IEEE 42020.FDIS 2019].
17.	Architecture Description Element	«Referenced Entity»	Any construct in an architecture description [SOURCE: ISO/IEC/IEEE 42010:2011].
18.	Architecture Description Framework	«Referenced Entity»	Conventions, principles and practices for the description of architectures established within a specific domain of application and/or community of stakeholders [SOURCE: ISO/IEC/IEEE 42010:2011].
19.	Architecture Framework Area (e.g. NAF)	«Area»	Environment where an architecture framework takes place.
20.	Area of Interest	«Area»	Business domain Environment where the enterprise, the stakeholders and the entity of interest take place.
21.	Artifact	«Entity»	Object made or shaped by a human, such as a tool or an art object ISO 18461:2016 Note: Artefact is a synonym.
22.	Calendar Period	«Plan»	Portion of time in system by which the beginning, length, and subdivisions of the year are fixed.
23.	Capability	«Entity»	The ability of one or more resources to deliver a specified type of effect or a specified course of action [Source: NAFv3].
24.	Capability Configuration	«Entity»	A composite structure representing the physical and human resources (and their interactions) in an enterprise, assembled to meet a capability).
25.	Concerns	«Referenced Entity»	A matter of interest or importance to the <i>stakeholder</i> [SOURCE: ISO/IEC/IEEE 42020.FDIS 2019].
26.	Connector	«Entity»	Anything which links two or more entities together.
27.	Correspondence	«Referenced Entity»	A relation between architecture elements used to express architecture relations of interest within an architecture description (or between architecture descriptions). Correspondences can be governed by correspondence rules. [SOURCE: ISO/IEC/IEEE 42010:2011].

28.	Correspondence Rule	«Referenced Entity»	Correspondence rules are used to enforce relations within an architecture description (or between architecture descriptions). [SOURCE: ISO/IEC/IEEE 42010:2011].
29.	Data Element	«Entity»	A formalized representation of data.
30.	Data Relationship	«Entity»	Any relationship between data elements.
31.	Desirer	«Entity»	Entity who expect something with a high or specified level of success.
32.	Effect In Context	«Entity»	A change which is a result or consequence of an action or other cause in a particular context.
33.	Enduring Task	«Plan»	A type of template behaviour recognized by an enterprise as being essential to achieving its goals - i.e. a template for a strategic specification of what the enterprise does.
34.	Enterprise Phase	«Phase»	period of time in the life cycle of an enterprise during which activities are performed that enable achievement of objectives for that phase. [Source: ISO/IEC/IEEE 42020.FDIS 2019].
35.	Entity Of Interest	«entity»	Anything being considered, described, discussed, studied or otherwise addressed during the architecting effort. Synonym: architecture entity. [Source: ISO/IEC/IEEE 42020.FDIS 2019].
36.	Environmental Factor	«Entity»	A circumstance, fact, or influence that interacts with an entity of interest.
37.	Function	«Entity»	An activity which is specified in context of the resource (human or machine) that performs it. [Source: NAFv3].
38.	Human Resource	«Entity»	Human considered in a business or organization, regarded as a significant asset in terms of skills and abilities.
39.	Implementation Delay	«Event»	Amount of time necessary for implementation of an entity.
40.	Implementation Event	«Event»	Any event occurring during an implementation period.
41.	Implementation Scenario	«Entity»	A postulated sequence or development of events occurring during an implementation period.
42.	Information Element	«Entity»	An item of information. Synonym: Information product.
43.	Interface Specification	«Entity»	A detailed formulation which provides a definitive description of an interface.
44.	Logical Data Model	«Entity»	A specification of business information requirements as a formal data structure, where relationships and entities are used to specify the logic which underpins the information. [Source: NAFv3].
45.	Logical Delay	«Event»	Amount of time considered in a logical context by which an entity is late or postponed.

46.	Logical Entity	«Entity»	Any entity considered in a logical context.
47.	Logical Event	«Event»	Any event considered in a logical context.
48.	Logical Scenario	«Entity»	Any postulated sequence or development of events occurring in a logical context.
49.	Logical Domain	«Configuration»	Any business domain considered in a logical context.
50.	Logical Relationship	«Entity»	Relationship between logical entities.
51.	Measure	«Entity»	Means of expressing a quantity [Source: ISO 6707-1:2017].
52.	Measure Category	«Entity»	A class or division of measures regarded as having particular shared characteristics.
53.	Measure In Context	«Entity»	Any measure considered in a particular context.
54.	Metadata	«Entity»	(Architecture) An annotation that can be applied to any element in an architecture.
55.	Needline	«Entity»	A relationship specifying the need to exchange something between nodes. Note: The needline does not indicate how the transfer is implemented. [Adapted from NAFv3].
56.	Node	«Entity»	A logical entity that performs operational activities. Note: nodes are specified independently of any physical realization. [Source: NAFv3].
57.	Node Lifeline	«Entity»	Description of the life of a node.
58.	Node Mode	«Entity»	Functional domain considered for a node at a point of time.
59.	Node State	«Entity»	Functional status of a node at a point of time.
60.	Operational Activity	«Behaviour»	Activity directly contributing to deliver a specified type of effect or to achieve a goal.
61.	Operational Effect	«Entity»	A change which is a result or consequence of an action or other cause in the context of a mission.
62.	Operational Exchange	«Entity»	Asserts that a flow can exist between nodes (i.e. flows of information, people, materiel, or energy).
63.	Operational Role	«Entity»	The function assumed or part played by an entity in an operational context.
64.	Organization	«Entity»	A group of people and facilities with an arrangement of responsibilities, authorities and relationships. [Source: ISO/IEC/IEEE 42020.FDIS 2019].
65.	Organizational Resource	«Entity»	An abstract entity for Organization, Person Post and Responsibility. [Adapted from UAF DMM].
66.	Person	«Entity»	A human being regarded as an individual.
67.	Physical	«Entity»	Any entity having an existence in the real world.

	Entity		
68.	Physical Data Model	«Entity»	An implementable specification of a data structure. A Physical Data Model realizes a Logical Data Model, taking into account implementation restrictions and performance issues whilst still enforcing the constraints, relationships and typing of the logical model. [Source: NAFv3].
69.	Post	«Entity»	A job or occupation in an organization.
70.	Project	«Plan»	A proposed or planned undertaking.
71.	Project Milestone	«Event»	A significant stage, phase or event in the life cycle of a project.
72.	Protocol	«Entity»	An official procedure or system of rules governing exchanges between entities.
73.	Provided Interface	«Entity»	Interface effectively existing in a capability configuration.
74.	Rectangle1	«rectangle»	
75.	Required Interface	«Entity»	Interface required in a capability configuration.
76.	Requirement	«Entity»	A statement that translates or expresses a need and its associated constraints and conditions [Source: ISO/IEC/IEEE 12207:2017].
77.	Resource	«Entity»	Anything that can be drawn on by a person or organization in order to function effectively.
78.	Resource Exchange	«Entity»	Asserts that a flow can exist between resources (i.e. flows of data, people, material, or energy).
79.	Resource Function	«Entity»	Function provided by a resource in a particular context.
80.	Resource Lifeline	«Entity»	Description of the life of a resource.
81.	Resource Mode	«Entity»	Functional domain considered for a node at a point of time.
82.	Resource Package	«Entity»	A collection of resources deployed according to a capability configuration.
83.	Resource State	«Entity»	Functional status of a node at a point of time.
84.	Resource Interaction	«Entity»	Reciprocal action or influence between two or more resources.
85.	Resource Structure	«Configuration»	The arrangement of and relations between the parts or elements of an entity.
86.	Resource Asset	«Entity»	A useful or valuable resource held by an organization.
87.	Responsible Entity	«Entity»	Any entity having an obligation to do something, or having control over or care for someone, as part of one's job or role.
88.	Service	«Entity»	A function, capability or behaviour that is provided by a producer to a consumer. Notes: A service should be implementation independent.

			Services are relevant in all aspects of an architecture. Services are delivered in accordance with defined patterns and agreements. The service term should not be used without a relevant context. [Source: NAFv3].
89.	Service Consumer	«Entity»	Entity representing a known or unknown service user.
90.	Service Delay	«Event»	Amount of time to get a service completion.
91.	Service Event	«Event»	Any event occurring in the context of a service completion.
92.	Service Function	«Behaviour»	An Activity that describes the abstract behaviour of Service Specifications, regardless of the actual implementation.
93.	Service Provider	«Entity»	Entity representing a known body for service provision. Note: the service provider is not necessarily the service producer.
94.	Service Scenario	«Entity»	Any postulated sequence or development of events occurring in the context of a service completion.
95.	Service Specification	«Entity»	The specification of a set of functionality provided one element for the use of others.
96.	Service State	«Entity»	Functional domain considered for a service at a point of time.
97.	Service Lifeline	«Entity»	Description of the life of a service.
98.	Service Mode	«Entity»	Functional status of a service at a point of time.
99.	Service Participant	«Entity»	Any actor involved in the context of a service completion.
100.	Stakeholder	«Referenced Entity»	Role, position, individual or organization having a right, share, claim or other interest in an architecture entity or its architecture that reflects their needs and expectations [Source: ISO/IEC/IEEE 42020.FDIS 2019].
101.	Standard	«Entity»	Item or document, established by consensus and approved by a recognized body, that provides, for common and repeated use, rules, guidelines or characteristics for activities or their results, aimed at the achievement of the optimum degree of order in a given context [Adapted from ISO/IEC TR 29110-1:2016].
102.	Standard Activity	«Behaviour»	Activity, established by consensus and approved by a recognized body, that provides, for common and repeated use, rules, guidelines or characteristics for activities or their results, aimed at the achievement of the optimum

			degree of order in a given context [Adapted from ISO/IEC TR 29110-1:2016].
103.	Standar Configuration	«Configuration»	Configuration, established by consensus and approved by a recognized body, that provides, for common and repeated use, rules, guidelines or characteristics for activities or their results, aimed at the achievement of the optimum degree of order in a given context [Adapted from ISO/IEC TR 29110-1:2016].
104.	Standard Role	«Entity»	The function assumed or part played by an entity in a standard context.
105.	Standards (applied throughout)	«Message»	Standards used in the current application.
106.	Statement Of Goal	«Entity»	A formulation which provides a definitive description of a goal.
107.	Status Indicator	«Entity»	Something that indicates the state or level of an entity.
108.	System	«Referenced Entity»	A combination of interacting elements organized to achieve one or more stated purposes [ISO/IEC/IEEE 15288:2015].
109.	Thing	«Entity»	Anything which could be considered for any purpose.
110.	Vision Statement	«Entity»	A formulation which provides a definitive description of a vision.
111.	Whole-Life Architecture	«Plan»	Set of distinguishable <i>phases</i> or <i>stages</i> that an <i>architecture</i> goes through Note 1 to entry: The architecture life cycle starts with the identification of a need for the architecture and ends when it is no longer needed. [Source: ISO/IEC/IEEE 42020.FDIS 2019].
112.	Whole-Life Enterprise	«Plan»	The enterprise involved in the life cycle of the entity of interest.
113.	Change Log	«Document»	A log or record of all notable changes made to a project.
114.	Architecture Meta-Data	«Entity»	[Refer to Meta-Data].
115.	Basic Concepts	«Entity»	Any abstract idea elaborated with common sense.
116.	Behaviour	«Behaviour»	The way in which an entity acts or conducts oneself, especially towards others.
117.	Concepts	«Entity»	Any abstract idea.
118.	Configuration	«Configuration»	An arrangement of parts or entities in a particular form, figure, or combination.
119.	Entity	«Entity»	Anything with distinct and independent existence.
120.	Event	«Event»	Anything that happens or takes place, especially one of importance.
121.	Key Concepts	«Entity»	Any abstract idea being critical or essential in a particular context.

122.	Logical Specification	«Entity»	A detailed formulation which provides a definitive description of one or more entities in a logical way.
123.	Phase	«Phase»	period of time in the <i>life cycle</i> during which activities are performed that enable achievement of objectives for that phase. [Source: ISO/IEC/IEEE 42020.FDIS 2019].
124.	Plan	«Plan»	Information item that presents a systematic course of action for achieving a declared purpose, including when, how, and by whom specific activities are to be performed [Source: ISO/IEC/IEEE 15289:2017].
125.	Programmatic Concepts	«Entity»	Any abstract idea elaborated in the context of a programme.
126.	Referenced Entity	«Referenced Entity»	Any entity provided one element for the use of others.
127.	Resource Specification	«Entity»	A detailed formulation which provides a definitive description of a resource.
128.	Service Specification	«Entity»	A detailed formulation which provides a definitive description of a service.
129.	Action	«Behaviour»	The way in which an entity has an effect or influence.
130.	Actor	«Entity»	An entity who behaves in a way that is not genuine.
131.	Data Model	«Entity»	A specification of information requirements as a formal data structure, where relationships and entities are used to specify information items. [Adapted from NAFv3].
132.	Data-Entity	«Entity»	Any data element with distinct and independent existence.
133.	Data-Entity Relationship	«Entity»	The way in which two or more data entities are connected, or the state of being connected.
134.	Delay	«Event»	A period of time by which an entity is late or postponed.
135.	Event	«Event»	Anything that happens or takes place, especially one of importance.
136.	Interface	«Entity»	A point or boundary where two entities can meet and interact.
137.	Lifeline	«Entity»	Description of the life of an entity.
138.	Mode	«Entity»	Functional status of an entity at a point of time.
139.	Scenario	«Entity»	A postulated sequence or development of events occurring during a given period.
140.	Scenario Part	«Entity»	Any items of a scenario.
141.	State	«Entity»	Functional status of an entity at a point of time.
142.	State Machine	«Entity»	A description of a postulated sequence of states of an entity occurring during the life time of this entity.
143.	State Machine Action	«Behaviour»	A state machine element describing a postulated action to be performed.

144.	Transition	«Entity»	A process or a period of changing from state of entity, mode of an entity or condition to another.
145.	Trigger	«Entity»	An event that is the cause of a particular action, process, or situation.



NATO ARCHITECTURE FRAMEWORK Version 4

Modeling Guidelines for use of the UAF-DMM

Architecture Capability Team
Consultation, Command & Control Board
March 2019

Document Version 1.0.0

DECLASSIFIED - PUBLIC DISCLOSURE / DÉCLASSIFIÉ - MIS EN LECTURE PUBLIQUE

