



Project: COMPASS

Grant Agreement: 287829

C O M P A S S

Comprehensive Modelling for Advanced Systems of Systems

**Definition of the COMPASS Architectural
Framework Framework**

Document Number: D21.5b

Date: September 2014

Public Document

<http://www.compass-research.eu>

Contributors:

Simon Perry (SAP), ATEGO

Jon Holt (JDH), ATEGO

Editor:

Simon Perry (SAP), ATEGO

Reviewers:

K. Pierce, UNUT

L. Lima, UFPE

U. Schulze, UB

Document History

Version	Date	Author	Description
0.1	2014-07-22	S. A. Perry	Initial draft version spun-off from D21.5
1.0	2014-08-15	S. A. Perry	First complete version
1.1	2014-09-22	S. A. Perry	Updated with comments from UNUT
1.2	2014-09-25	S. A. Perry	Updated with comments from UFPE & UB

Abstract

Developing a System of Systems (SoS) architecture is an essential SoS Engineering activity and any such Architecture should be produced according to defined Architectural Viewpoints codified in an Architectural Framework (AF) that includes consistency rules defined between the various Views produced based on the AF and the information contained in them. This deliverable defines an Architectural Framework, called the COMPASS Architectural Framework Framework (CAFF). CAFF can be used for defining concrete Architectural Frameworks and Architectural Patterns, and has been used throughout the COMPASS project. The CAFF defines six Viewpoints that should be produced when defining an AF. This document describes these Viewpoints and gives, through an example AF, an illustration of their use.

Table of Contents

1	Introduction	8
2	Guidelines for Architectural Frameworks	9
	2.1 Existing Architectural Frameworks.....	9
	2.2 Overview of AF Process Approach.....	9
3	The COMPASS Architectural Framework Framework	11
	3.1 The Needs for the CAFF - The AF Context View	12
	3.2 The Ontology for CAFF – The Ontology Definition View	13
	3.3 The Viewpoints and Perspectives for CAFF – The Viewpoint Relationships View.....	17
	3.4 The Rules Constraining the CAFF – The Rule Definition View	18
	3.5 The Viewpoints of CAFF – Viewpoint Definition.....	19
4	Summary	33
5	References	34
	Appendix A Processes for AF Definition and Construction	35
	A.1 Requirements Context View	35
	A.2 Stakeholder View	36
	A.3 Process Content View.....	37
	A.4 Process Behaviour Views.....	41
	A.4.1 PBV – AF Definition Process.....	42
	A.4.2 PBV – Viewpoint Definition Process	43
	A.4.3 PBV – Ontology Definition Process.....	44
	A.4.4 PBV – Context Process.....	45
	A.5 Information Views	46
	A.5.1 IV – AF Definition process.....	46
	A.5.2 IV – Viewpoint Definition Process.....	46
	A.5.3 IV – Ontology Definition Process.....	47
	A.5.4 IV – Context Process.....	47
	Appendix B – Architectural Design Framework	48
	B.1 AF Context View	48
	B.2 Ontology Definition View	48
	B.3 Viewpoint Relationship View.....	49
	B.4 Rules Definition View.....	50
	B.5 Viewpoint Definitions	50
	B.6 Summary of Defined Viewpoints	61

Table of Figures

Figure 1 - AF Context View Showing the Requirements for the CAFF	12
Figure 2 - Ontology Definition View for Architectural Frameworks	14
Figure 3 - Ontology Definition View Expanding on Context.....	16
Figure 4 - Viewpoint Relationships View Showing Viewpoints and Perspectives that make up the AF Framework	17
Figure 5 - Rules Definition View for the AF Framework	18
Figure 6 - Viewpoint Context View for the AF Context Viewpoint	19
Figure 7 - Viewpoint Definition View for the AF Context Viewpoint.....	21
Figure 8 - Viewpoint Context View for the Ontology Definition Viewpoint.....	22
Figure 9 - Viewpoint Definition View for the Ontology Definition Viewpoint	23
Figure 10 - Viewpoint Context View for the Viewpoint Relationships Viewpoint	24
Figure 11 - Viewpoint Definition View for the Viewpoint Relationships Viewpoint	25
Figure 12 - Viewpoint Context View for the Viewpoint Context Viewpoint	26
Figure 13 - Viewpoint Definition View for the Viewpoint Context Viewpoint.....	27
Figure 14 - Viewpoint Context View for the Viewpoint Definition View	28
Figure 15 - Viewpoint Definition View for the Viewpoint Definition Viewpoint..	29
Figure 16 - Viewpoint Context View for the Rules Definition Viewpoint	31
Figure 17 - Viewpoint Definition View for the Rules Definition Viewpoint.....	32
Figure 18 - Requirements context view (RCV) for the architectural framework definition processes.....	35
Figure 19 - Stakeholder view (SV) for the architectural definition processes.....	36
Figure 20 - Process content view (PCV) for the architectural framework definition processes.....	37
Figure 21 - PCV for AF Definition Process.....	38
Figure 22 - PCV for Viewpoint Definition Process	39
Figure 23 - PCV for Ontology Definition Process.....	40
Figure 24 - PCV for Context Process.....	41
Figure 25 - PBV for AF Definition Process.....	42
Figure 26 - PBV – Viewpoint Definition Process.....	43
Figure 27 - PBV – Ontology Definition Process.....	44
Figure 28 - PBV – Context Process	45
Figure 29 - IV – AF Definition process	46
Figure 30 - IV – Viewpoint Definition Process.....	46
Figure 31 - IV – Ontology Definition Process	47
Figure 32 - IV – Context Process	47
Figure 33 AF Context View for the Architectural Design Framework.....	48
Figure 34 Ontology Definition View for the Architectural Design Framework.....	49
Figure 35 Viewpoint Relationship View for the Architectural Design Framework	50
Figure 36 Viewpoint Context View for the Logical Structure Viewpoint.....	51
Figure 37 Viewpoint Definition View for the Logical Structure Viewpoint	52
Figure 38 Viewpoint Context View for the Logical Interaction Viewpoint	52
Figure 39 Viewpoint Definition View for the Logical Interaction Viewpoint.....	53
Figure 40 Viewpoint Context View for the Component Structure Viewpoint.....	54
Figure 41 Viewpoint Definition View for the Component Structure Viewpoint...	55

Figure 42 Viewpoint Context View for the Component Interaction Viewpoint	55
Figure 43 Viewpoint Definition View for the Component Interaction Viewpoint.....	56
Figure 44 Viewpoint Context View for the Deployment Viewpoint.....	57
Figure 45 Viewpoint Definition View for the Deployment Viewpoint	58
Figure 46 Viewpoint Context View for the Environment Identification Viewpoint	58
Figure 47 Viewpoint Definition View for the Environment Identification Viewpoint.....	59
Figure 48 Viewpoint Context View for the Environment Configuration Viewpoint	60
Figure 49 Viewpoint Definition View for the Environment Configuration Viewpoint.....	61

1 Introduction

Developing an SoS architecture is an essential SoS Engineering activity. As discussed in sections 2.4.2 and 2.5 of COMPASS deliverable D22.3 “Report on Modelling Patterns for SoS Architectures” [D22.3 2013]:

1. *SoS Architectures* are *core* to systems and SoS engineering; production of architectures is *not* an optional activity.
2. *SoS Architectures* should be produced according to defined *Architectural Viewpoints* codified in an *Architectural Framework* that includes *consistency rules* defined between the various views produced and the information contained in them.
3. *SoS Architectures* should be produced to address the *concerns of stakeholders* using *relevant* architectural viewpoints; it is important to know *why* a particular viewpoint is being used.
4. It is essential that an *SoS architectural design process* is defined and followed. This must cover the definition, analysis, evaluation, documentation and maintenance of an architecture.
5. *SoS Architectures* should address both *structure and behaviour*, including communications, functionality and data flow.
6. Modelling is essential to the development of *SoS architectures*.

This document addresses points 2 and 3 above by discussing over-arching guidelines (Section 2) before introducing the COMPASS Architectural Framework Framework (CAFF), which is a meta-framework to be used in the construction of a SoS architectural framework for a given SoS project (Section 3). To assist in construction of a concrete SoS Architectural Framework a set of processes has been defined and described in Appendix A.

An example AF created using the CAFF is presented in Appendix B.

The primary target group for this document is system and software engineers involved in the development of AFs or architectural patterns, which are then to be used in architecting an SoS.

Note that the content of this document is the final version of the information found in “Initial Report on Guidelines for Architectural Level SoS Modelling” [D21.2 2013]. For this final version, it has been separated out and presented as a sub-document of “Final Report on Guidelines for Architectural Level SoS Modelling” [D21.5 2014], the successor to D21.2.

2 Guidelines for Architectural Frameworks

Architectures and Architectural Frameworks (AFs) are an essential part of systems engineering and SoS engineering as well, where *SoS Architectures* should be produced according to a set of defined *Architectural Viewpoints* codified in an *Architectural Framework*. The following sub-sections introduce existing architectural frameworks and outline an approach to developing an AF.

2.1 Existing Architectural Frameworks

While there are a number of widely-used architectural frameworks, this does not necessarily mean that they are suitable for all systems engineering and SoS engineering projects; it is important that the *correct* AF, fit for purpose, be used. For example:

- Defence frameworks, such as MODAF, DoDAF and NAF, are intended to be used in the acquisition of systems.
- TOGAF is intended to provide an approach for developing IT architectural frameworks.
- Zachman is a framework approach for defining IT-based enterprise architectures.

For a discussion of these frameworks see “D22.1 – Initial Report on SoS Architectural Models” [D22.1 2012].

The use of an AF is a prerequisite for the development of a robust architecture. Before any existing AFs can be assessed for suitability, it is essential that the needs for the architecture, and hence its guiding AF, are understood. If an existing AF is not suitable, then a project will have to define its own AF.

D22.1 concluded that “*there is no single technique that stands out as being ideal as an off-the-shelf solution that can be used for the COMPASS project*”. Therefore there is a need for an approach for defining a bespoke architectural framework.

2.2 Overview of AF Process Approach

A full set of processes for the definition of an architectural framework has been defined in Appendix A, which may be summarised as:

1. Identify needs for the architectural framework. Identify the needs that the architectural framework is to address, placing them into context.
2. Define basic concepts and terms. It is essential that the underlying ontology is defined and understood.
3. Identify an initial set of views. Relate these views together in the form of a framework and classify them into perspectives.
4. Define each of these views in terms of its needs and concepts. Understand the needs that the viewpoint must address, placing them into context.
5. Define the viewpoint using concepts from the ontology so that the identified needs *are* addressed.

6. Define any rules that constrain the viewpoints and framework.

These processes are defined in Appendix A and make use of the COMPASS Architectural Framework Framework (CAFF) which is defined in Section 3.

3 The COMPASS Architectural Framework Framework

When defining an AF it is essential that:

- The needs of the AF are understood.
- The concepts and terms that the AF can model are defined.
- The views of the AF are identified.
- The needs for each view are understood and are related to the overall needs of the AF.
- The definition of each view is specified and based on the concepts and terms identified.
- Any rules that constrain the AF are captured.

In order to address these six points in a consistent model-based systems engineering manner, COMPASS defines an AF that is used for the definition of *other* AFs - the COMPASS Architectural Framework Framework (CAFF). The complete definition of the CAFF is below. It should be used along with the AF processes defined in Appendix A and summarised in Section 2.2 above.

It is worth emphasising here what the CAFF *is* and *is not*. The CAFF is *not* an AF that can be used to produce system architectures, like MODAF [MODAF 2010]. The CAFF is an AF *that is used to define other AFs in a consistent, robust and model-based manner*.

The following sections define the CAFF. The CAFF defines a number of viewpoints that are the minimal set needed for the definition of an AF. What is more, the viewpoints are defined *using* CAFF – that is, the CAFF is defined in terms of itself. This self-referentiality is not uncommon in modelling; for example, the Unified Modelling Language (UML) is defined using the Unified Modelling Language.

The five sub-sections:

1. Define the needs (i.e. requirements) for the CAFF.
2. Define the ontology for the CAFF.
3. Identify the viewpoints that make up the CAFF, along with their grouping into related viewpoints; such a grouping is known as a perspective.
4. Define each CAFF viewpoint.
5. Define some rules that constrain the CAFF.

The sections make use of the Systems Engineering Modelling Language (SysML). For details of the SysML see [SysML 2010] and [Holt & Perry2008].

The CAFF has been used in a number of areas of the COMPASS project. An example AF for architectural design that has been defined using the CAFF can be found in Appendix B of this document; the Fault Modelling Architectural Framework (FMAF) has been defined using CAFF and can be found in “D24.2 – Report on

Timed Fault Tree Analysis - Fault Modelling” [D24.2 2013]; the enabling patterns found in “D22.6 – Final Report on SoS Architectural Models” [D22.6 2014] have all been documented using the CAFF. Finally an industrial application of the CAFF, the Streaming Architectural Framework (SAF) has been produced by B&O as part of their case study work and can be found in “A/V/HA Ecosystems Prototype Using COMPASS Methods & Tools” [D42.2 2014].

3.1 The Needs for the CAFF - The AF Context View

This section defines the needs for the CAFF, presented using a SysML use case diagram. It is an example of a CAFF *Architectural Framework Context View*.

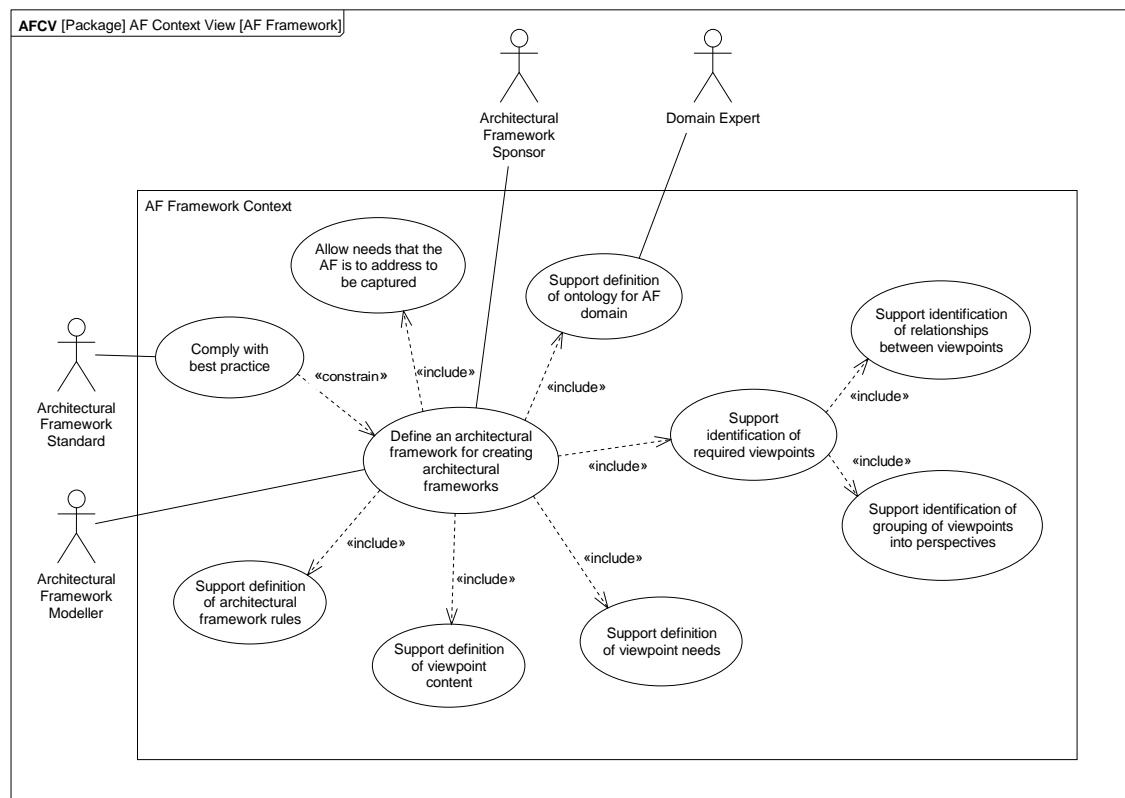


Figure 1 - AF Context View Showing the Requirements for the CAFF

Figure 1 shows the Architectural Framework Context for the definition of the CAFF. That is, it shows the needs (i.e. requirements, known as concerns in ISO42010 “ISO/IEC 42010:2011 Systems and software engineering — Architecture description” [ISO42010 2011]) that must be fulfilled when defining a Framework that is to be used for the definition of other Architectural Frameworks.

The main need that must be fulfilled is to "Define an architectural framework for creating architectural frameworks", constrained by the need to "Comply with best practice" such as Architectural Framework Standards (for example, [ISO42010 2011]). In order to "Define an architectural framework for creating architectural frameworks" it is necessary to:

- "Allow needs that the AF is to address to be captured" - When defining an Architectural Framework, it is important that the needs that the Architectural Framework is to address can be captured, in order to ensure that the Architectural Framework is fit for purpose.
- "Support definition of ontology for AF domain" - When defining an Architectural Framework, it is essential that the concepts, and the relationships between them, are defined for the domain in which the Architectural Framework is to be used. This is the Ontology that forms the foundational basis of the definition of the Architectural Framework's Viewpoints. Such an Ontology ensures the *consistency* of the Architectural Framework. The Architectural Framework must support such a definition of an Ontology.
- "Support identification of required viewpoints" - The Viewpoints that make up the Architectural Framework need to be identified. As well as supporting such an identification, the Architectural Framework must also "Support identification of relationships between viewpoints" and "Support identification of grouping of viewpoints into perspectives".
- "Support definition of viewpoint needs" - In order to define the Viewpoints that make up an Architectural Framework, it is essential that the needs of each Viewpoint be clearly understood in order to ensure each Viewpoint is fit for purpose and that the Viewpoints defined meet the overall needs for the Architectural Framework.
- "Support definition of viewpoint content" - An Architectural Framework is essentially a number of Viewpoints that conform to an Ontology. Therefore, when defining an Architectural Framework it is essential that each Viewpoint can be defined in a consistent fashion that ensures its conformance to the Ontology.
- "Support definition of architectural framework rules" - Often, when defining an Architectural Framework, it is necessary to constrain aspects of the Architectural Framework through the definition of a number of constraining Rules. It is therefore essential that an AF Framework supports the definition of such Rules.

The key Stakeholder Roles involved are:

- Architectural Framework Sponsor - the role involved in sponsoring the creation of the Architectural Framework.
- Architectural Framework Modeller - the role involved in the modelling and definition of an Architectural Framework.
- Architectural Framework Standard - the role of any appropriate standard for Architectural Frameworks. An example of a standard that could fill this role is [ISO42010 2011].
- Domain Expert - the role of an expert in the domain for which the Architectural Framework is to be used.

3.2 The Ontology for CAFF – The Ontology Definition View

This section defines the ontology for the CAFF, presented using SysML block definition diagrams. The diagrams are examples of a CAFF *Ontology Definition View*.

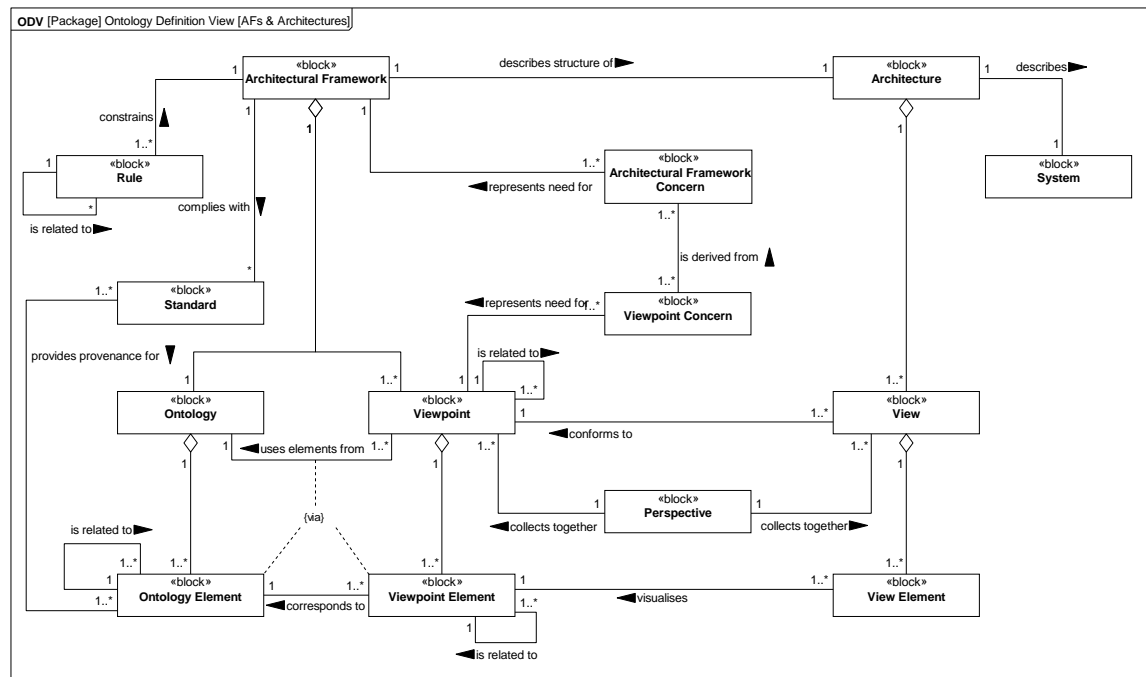


Figure 2 - Ontology Definition View for Architectural Frameworks

Figure 2 defines the ontology associated with Architectures and Architectural Frameworks:

The [Package] Ontology Definition View [AFs & Architectures] diagram defines the ontology associated with Architectures and Architectural Frameworks:

- Architectural Framework - a defined set of Viewpoints and an Ontology. The Architectural Framework is used to structure an Architecture from the point of view of a specific industry, stakeholder role set, or organisation. The Architectural Framework is defined so that it meets the needs defined by its Architectural Framework Concerns. An Architectural Framework is created so that it complies with zero or more Standards.
- Architectural Framework Concern - defines a need (requirement) that an Architectural Framework has to address.
- Ontology - an element of an Architectural Framework that defines all the concepts and terms (Ontology Elements) that relate to any architecture structured according to the Architectural Framework.
- Ontology Element - the concepts that make up an Ontology. Ontology Elements can be related to each other and are used in the definition of each viewpoint (through the Viewpoint Element that makes up a viewpoint). The provenance for Ontology Elements is provided by one or more standards.
- Viewpoint - a definition of the structure and content of a view. The content and structure of a Viewpoint uses the concepts and terms from the Ontology via the Viewpoint Elements that make up the Viewpoint. Each viewpoint is defined so that it meets the needs defined by its Viewpoint Concerns.

- Viewpoint Concern - defines a need (requirement) that a Viewpoint has to address.
- Viewpoint Element - the elements that make up a Viewpoint. Each Viewpoint Element must correspond to an Ontology Element from the Ontology that is part of the Architectural Framework.
- Architecture - a description of a System, made up of a number of Views. Related Views can be collected together into Perspectives.
- View - the visualisation of part of the Architecture of a System, that conforms to the structure and content defined in a Viewpoint. A View is made up of a number of View Elements.
- View Element - the elements that make up a View. Each View Element visualises a Viewpoint Element that makes up the Viewpoint to which the View conforms.
- Perspective - a collection of Views (and hence also their defining Viewpoints) that are related by their purpose. That is, Views which address the same architectural needs, rather than being related in some other way, such as by mode of visualisation, for example.
- Rule - a construct that constrains the Architectural Framework (and hence the resulting Architecture) in some way, for example by defining minimum required Viewpoints.
- System - set of interacting elements organised to satisfy one or more needs. The artefact being engineered that the Architecture describes.

Two concepts illustrated in Figure 2 that are important to understand are those of *viewpoint* and *view*. An architectural framework is made up of a number of viewpoints that define the information that can be presented. When an actual architecture is developed, which is based on the architectural framework, then one produces *views* that conform to the definitions in the corresponding *viewpoint*. That is, a view is a *realisation* of a viewpoint. Not all architectural frameworks make this distinction. For example, the UK Ministry of Defence Architectural Framework (MODAF – see [MODAF 2010]) makes no such distinction. It defines a number of *views* but does not differentiate between the definition and realisation in terms of the language and terms used. Even more confusingly MODAF does use the term *viewpoint*, but in MODAF *viewpoint* is the same as *perspective* in Figure 2, simply a collection of related views.

The Ontology is based on concepts from a number of sources, including: [ISO42010:2011] and [TRAK 2013]. The concept of *context* is expanded further in Figure 3 below, which is based on ideas that can be found in [Holt et al. 2011] and [D21.1 2012].

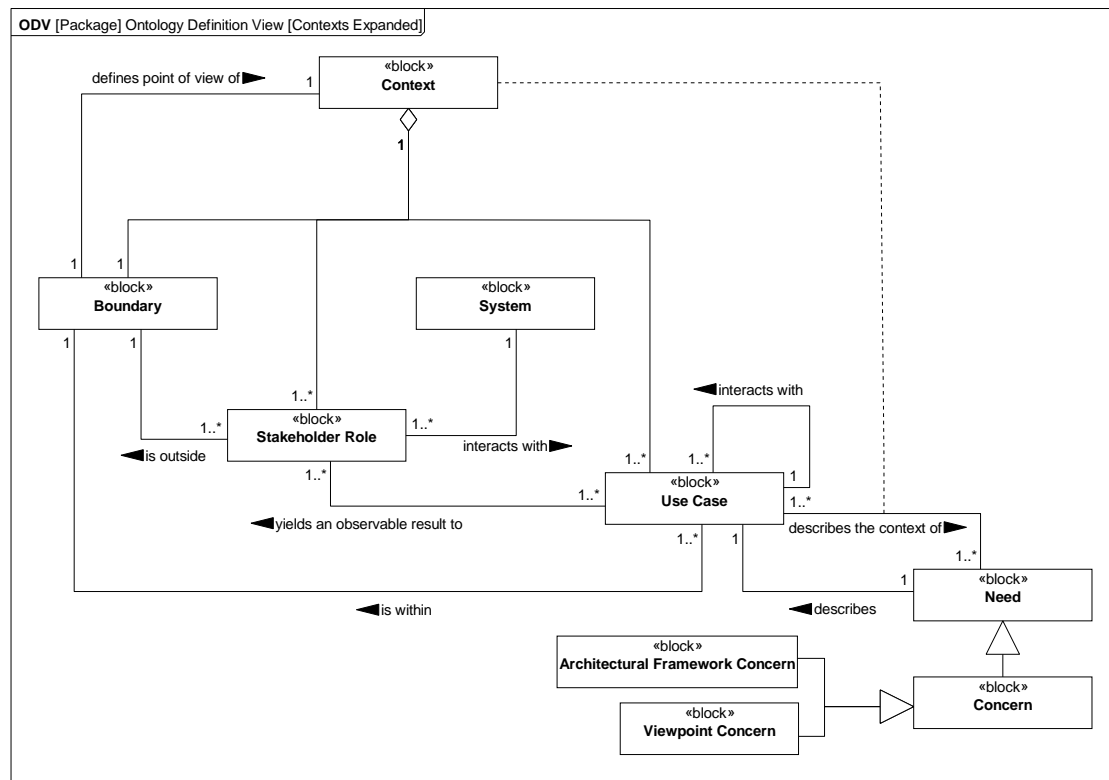


Figure 3 - Ontology Definition View Expanding on Context

Figure 3 defines additional ontology elements associated with Needs (requirements) that define the Contexts for Architectural Frameworks and the Viewpoints that make them up:

- Context - the definition of the *Needs* (for something - such as a System or an Architectural Framework) from a *specific point of view*. A Context is made up of a Boundary that defines the point of view of the Context, one or more Stakeholder Roles and one or more Use Cases.
- Boundary - defines the boundary, and hence the point of view, of a Context. The Use Cases that make up a Context are *inside* the Boundary. The Stakeholder Roles that make up a Context are *outside* the Boundary.
- Use Case - represents a need (requirement, concern) that is given meaning by being considered in a specific Context. Use Cases yield observable results to Stakeholder Roles and interact with other Use Cases.
- Stakeholder Role - the role of *anything* that has an interest in a System. Examples of a Stakeholder Role include the roles of a person, an organisational unit, a project, a standard, an enabling system, etc. Stakeholder Roles lie outside the Boundary of a Context.
- Concern - a type of Need (a requirement). Architectural Framework Concerns represent the Needs for an Architectural Framework; Viewpoint Concerns represent the Needs for a Viewpoint.

3.3 The Viewpoints and Perspectives for CAFF – The Viewpoint Relationships View

This section identifies the viewpoints that make up the CAFF, presented using a SysML block definition diagram. Perspectives are also shown, through the use of SysML packages. It is an example of a CAFF *Viewpoint Relationships View*.

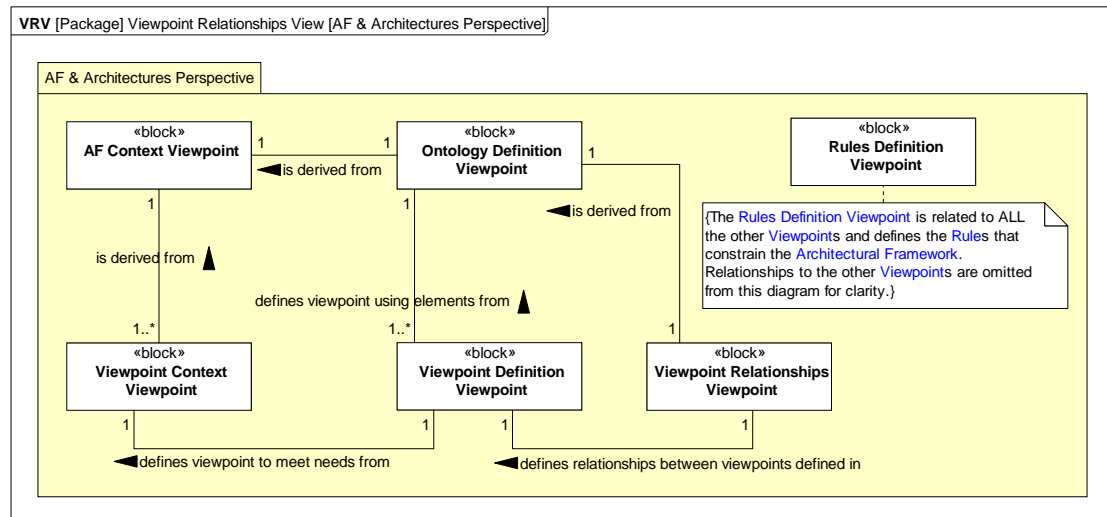


Figure 4 - Viewpoint Relationships View Showing Viewpoints and Perspectives that make up the AF Framework

Figure 4 shows the various Viewpoints that make up the CAFF, the relationships between them and the Perspectives to which the Viewpoints belong.

The CAFF is a *meta*-Architectural Framework: an Architectural Framework for defining Architectural Frameworks. It is made up of six Viewpoints:

- The AF Context Viewpoint that defines the Context for the Architectural Framework. That is, it represents the Architectural Framework Concerns in Context.
- The Ontology Definition Viewpoint defines the Ontology for the Architectural Framework. It is derived from the AF Context Viewpoint.
- The Viewpoint Relationships Viewpoint that shows the relationships between the Viewpoints that make up the Architectural Framework and groups them into Perspectives. It is derived from the Ontology Definition Viewpoint.
- The Rules Definition Viewpoint that defines the various Rules that constrain the Architectural Framework.
- The Viewpoint Context Viewpoint that defines the Context for a particular Viewpoint. That is, it represents the Viewpoint Concerns in Context for a particular Viewpoint. It is derived from the AF Context Viewpoint.
- The Viewpoint Definition Viewpoint that defines a particular Viewpoint, showing the Viewpoint Elements (and hence the Ontology Elements) that appear on the Viewpoint.

The Viewpoints are collected into a single Perspective, the AF & Architectures Perspective as shown by the enclosing package.

3.4 The Rules Constraining the CAFF – The Rule Definition View

View

This section identifies some rules that constrain the CAFF, presented using a SysML block definition diagram. It is an example of a CAFF *Rule Definition View*.

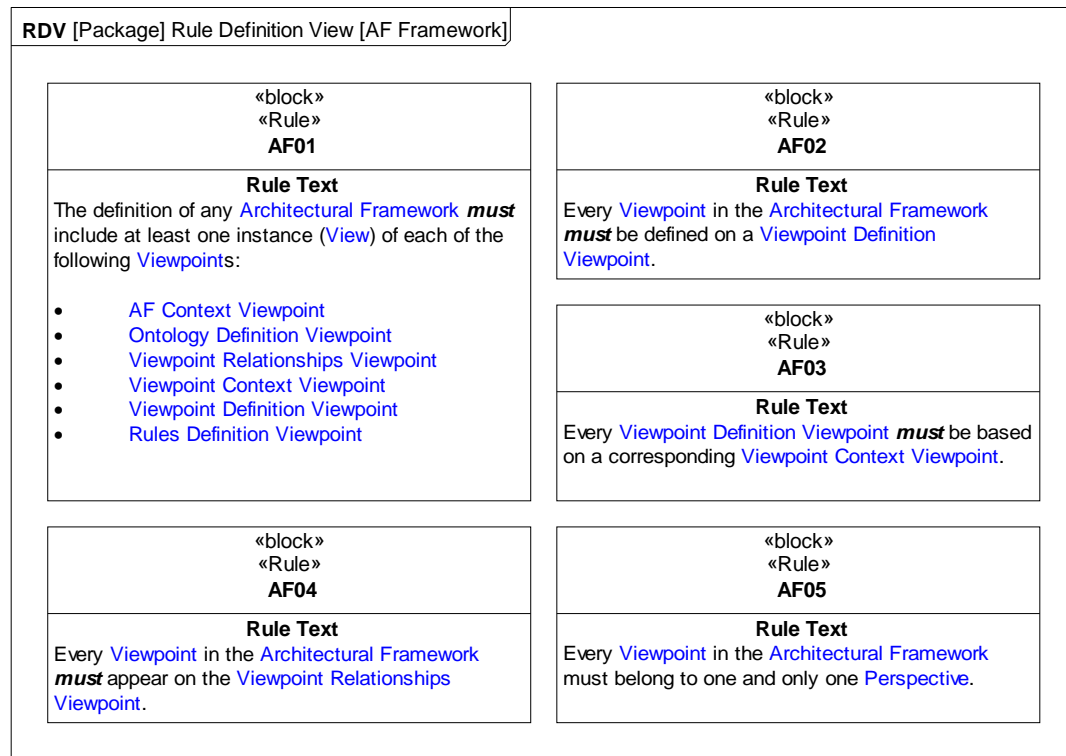


Figure 5 - Rules Definition View for the AF Framework

Figure 5 shows five example rules that constrain the AF Framework. In this diagram, each Rule is represented by a SysML block that has had the «Rule» stereotype applied. Associated with this stereotype is a tag, Rule Text, which holds the text of the Rule and which is displayed in a separate compartment.

There is no reason why SysML blocks have to be used for the definition of rules, particularly in this case when all the rules are essentially textual descriptions wrapped in SysML blocks. They could simply have been described using the rule text on a purely textual "diagram". However, using blocks does bring with it the advantage of support for traceability. If it is desirable to show that elements of the Architectural Framework conform to the various Rules, then having them defined using blocks allows this to be done using the SysML traceability mechanism of stereotypes dependencies. Defining them purely as text would not allow this to be done.

3.5 The Viewpoints of CAFF – Viewpoint Definition

This section defines the six viewpoints that make up the COMPASS AF Framework. These viewpoints are identified in Figure 4. For each viewpoint defined here there are three parts to the definition:

1. A *Viewpoint Context View* defining the needs that the viewpoint addresses.
2. A *Viewpoint Definition View* defining the contents of the viewpoint.
3. An example of the viewpoint defined.

The viewpoints are defined in the following order:

1. AF Context Viewpoint (AFCV)
2. Ontology Definition Viewpoint (ODV)
3. Viewpoint Relationships Viewpoint (VRV)
4. Viewpoint Context Viewpoint (VCV)
5. Viewpoint Definition Viewpoint (VDV)
6. Rules Definition Viewpoint (RDV)

AF Context Viewpoint (AFCV)

This section defines the AF Context Viewpoint (AFCV).

Viewpoint Context View

The needs that the AF Context Viewpoint (AFCV) is intended to address are shown in the following diagram, a CAFF Viewpoint Context View.

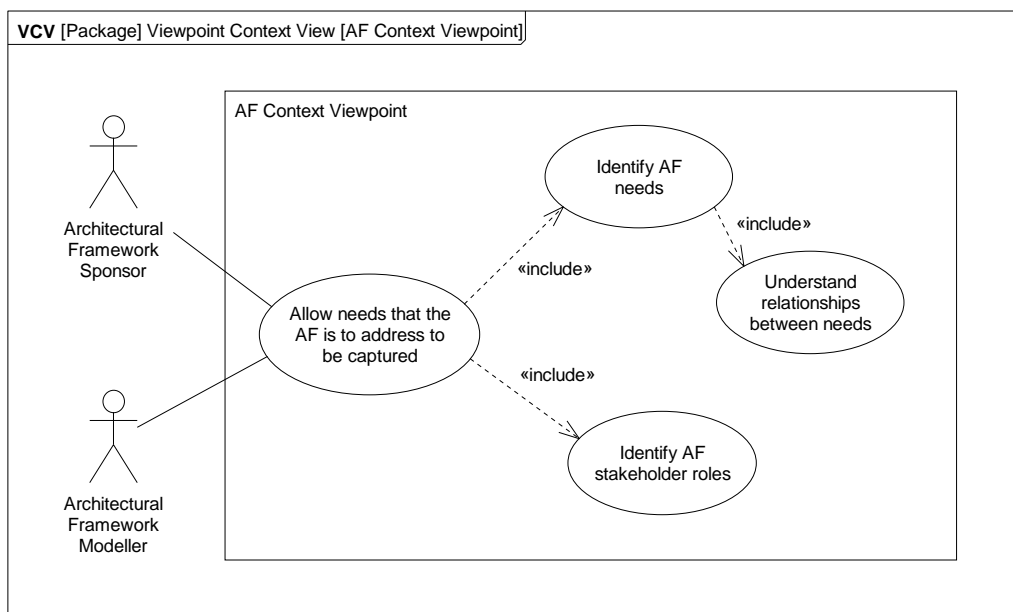


Figure 6 - Viewpoint Context View for the AF Context Viewpoint

Figure 6 shows the Viewpoint Context View for the AF Context Viewpoint. That is, it defines the needs that the AF Context Viewpoint must address, together with relevant Stakeholder Roles.

The main need, taken from Architectural Framework Context (see Figure 1) is to "Allow needs that the AF is to address to be captured"; the AF Context Viewpoint exists solely to capture the needs of the Architectural Framework being defined.

In order to do this, it is necessary to be able to:

- "Identify AF needs" - Identify the needs that the Architectural Framework is being created to address.
- "Understand relationships between needs" - Understand any relationships *between* the needs that the Architectural Framework is being created to address.
- "Identify AF stakeholder roles" - Identify the Stakeholder Roles involved in definition of the Architectural Framework that have an interest in or are affected by the identified needs.

One approach to doing this is the SoS-ACRE approach to model-based requirements engineering described in "D21.1 – Report on Guidelines for SoS Requirements" [D21.1 2012].

As identified on Figure 1, the two key Stakeholder Roles involved are (through the «include» relationships from "Define an architectural framework for creating architectural frameworks"): the Architectural Framework Sponsor and the Architectural Framework Modeller.

Viewpoint Definition View

The definition of the AF Context Viewpoint (AFCV) is shown in the following diagram, a CAFF Viewpoint Definition View.

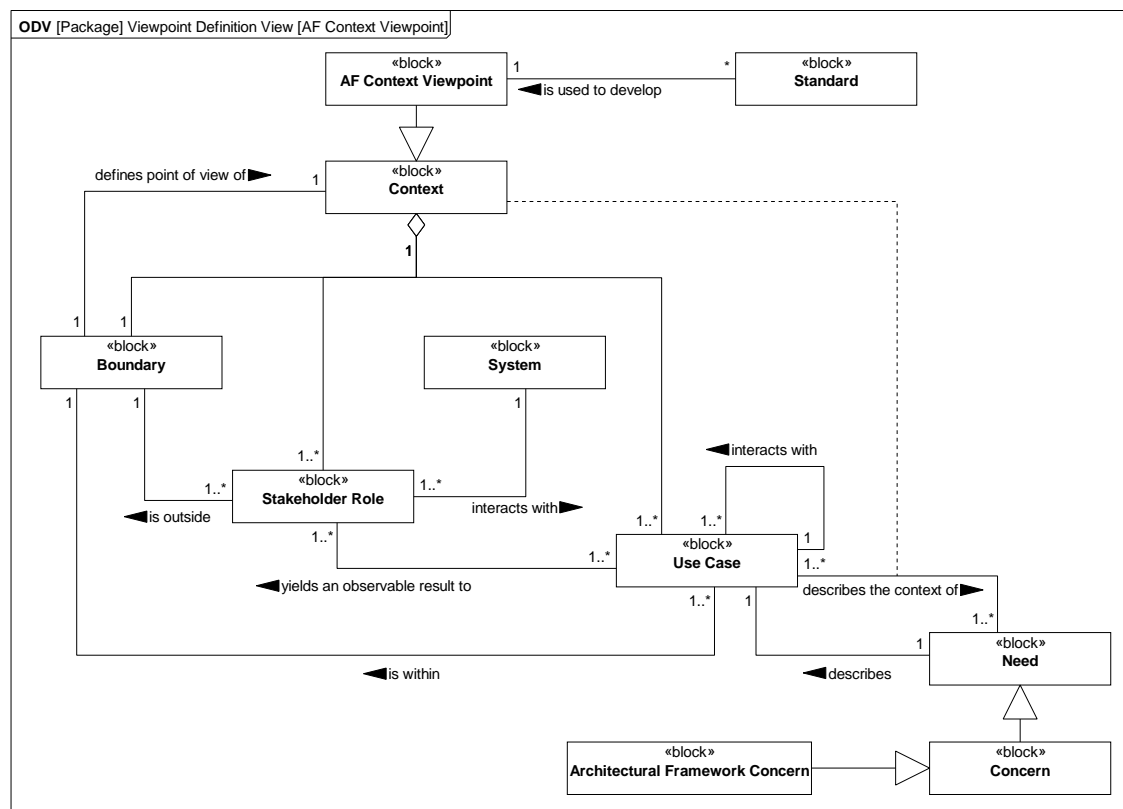


Figure 7 - Viewpoint Definition View for the AF Context Viewpoint

Figure 7 addresses the needs defined on Figure 6.

It defines the content of the AF Context Viewpoint, which defines the Context for an Architectural Framework. It is made up of a Boundary, one or more Stakeholder Roles, that are outside the Boundary, and one or more Use Cases that are inside the Boundary.

Each Use Case represents an Architectural Framework Concern (a type of Need) for the Architectural Framework, that yields observable results to one or more Stakeholder Roles. Each Use Case may interact with a number of other Use Cases.

Example View

For an example of an AF Context View, see Figure 1.

Ontology Definition Viewpoint (ODV)

This section defines the Ontology Definition Viewpoint (ODV).

Viewpoint Context View

The needs that the Ontology Definition Viewpoint (ODV) is intended to address are shown in the following diagram, a CAFF Viewpoint Context View.

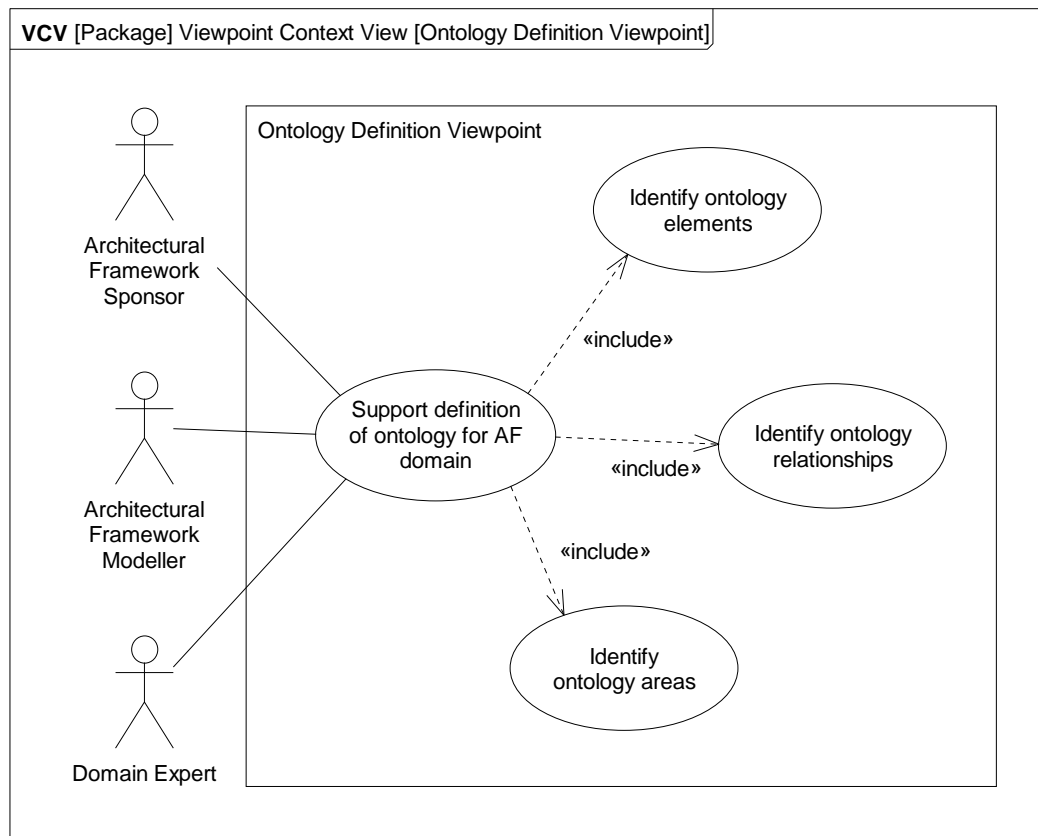


Figure 8 - Viewpoint Context View for the Ontology Definition Viewpoint

Figure 8 shows the Viewpoint Context View for the Ontology Definition Viewpoint. That is, it defines the needs that the Ontology Definition Viewpoint must address, together with relevant Stakeholder Roles.

The main need, taken from the Architectural Framework Context (see Figure 1) is to "Support definition of ontology for AF domain"; the Ontology Definition Viewpoint exists to define the Ontology that defines all the concepts and terms (Ontology Elements) that relate to any Architecture structured according to the Architectural Framework.

The main needs that must be addressed are to:

- "Identify ontology elements" - Identify the Ontology Elements for the domain in which the Architectural Framework will be used.
- "Identify ontology relationships" - Identify the relationships *between* the Ontology Elements. Such relationships are equally as important a part of the Ontology defined using this Viewpoint as are the Ontology Element.
- "Identify ontology areas" - When defining an Ontology it is often useful to group together related Ontology Elements. For example, when defining an Ontology for systems engineering, one could expect to see groupings of Ontology Element related to the concepts of System, Life Cycle, Process, Project etc. Such groupings are useful when defining an Architectural Framework, as they help to identify the Perspectives into which the Viewpoints, and the Views based on them, are grouped. See [Holt & Perry 2013] for guidelines on how to define an Ontology.

As identified on Figure 1, the three key Stakeholder Roles involved in the main need to "Support definition of ontology for AF domain" are (through the «include» relationships from "Define an architectural framework for creating architectural frameworks" and directly): the Architectural Framework Sponsor, the Architectural Framework Modeller and the Domain Expert.

Viewpoint Definition View

The definition of the Ontology Definition Viewpoint (ODV) is shown in the following diagram, a CAFF Viewpoint Definition View.

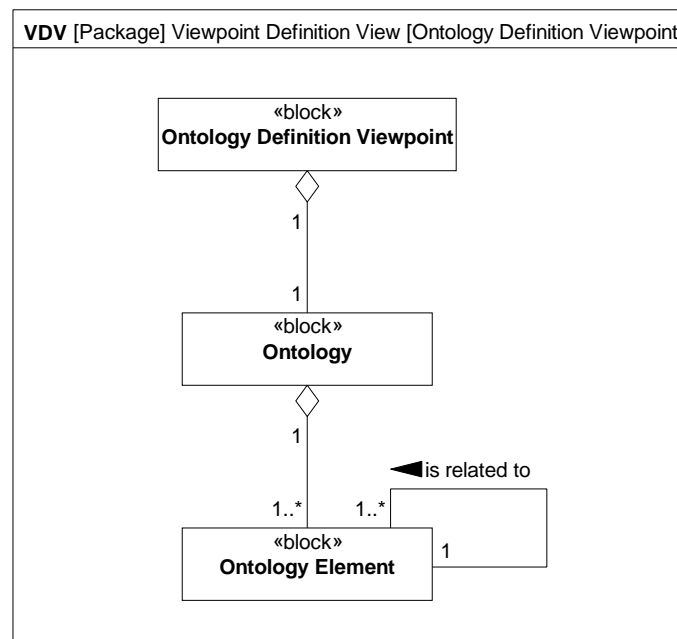


Figure 9 - Viewpoint Definition View for the Ontology Definition Viewpoint

Figure 9 addresses the needs defined on Figure 8.

It defines the content of the Ontology Definition Viewpoint, which is made up of an Ontology that is itself made up of one or more Ontology Element(s) that are related to each other.

It is important to note that the relationships between the Ontology Elements are themselves an important part of the Ontology.

Example View

For an example of an Ontology Definition View see Figure 2 and Figure 3.

Viewpoint Relationships Viewpoint (VRV)

This section defines the Viewpoint Relationships Viewpoint (VRV).

Viewpoint Context View

The needs that the Viewpoint Relationships Viewpoint (VRV) is intended to address are shown in the following diagram, a CAFF Viewpoint Context View.

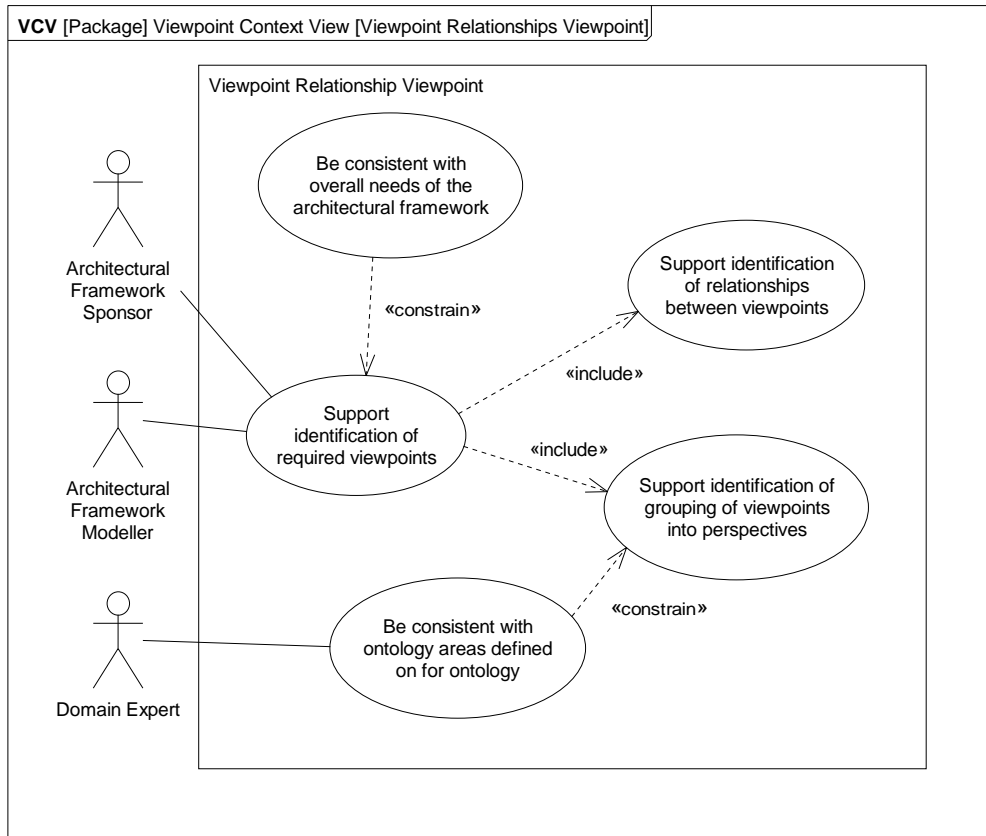


Figure 10 - Viewpoint Context View for the Viewpoint Relationships Viewpoint

Figure 10 shows the Viewpoint Context View for the Viewpoint Relationships Viewpoint. That is, it defines the needs that the Viewpoint Relationships Viewpoint must address, together with relevant Stakeholder Roles.

The main need, taken from the Architectural Framework Context (see Figure 1) is to "Support identification of required viewpoints", which includes the need to "Support identification of relationships between viewpoints" and "Support identification of grouping of viewpoints into perspectives". The Viewpoint Relationships Viewpoint exists to identify the Viewpoints that make up the Architectural Framework, show how they are related to each other and show how they are grouped into Perspectives.

The needs for this Viewpoint are subject to two constraints:

- The main need to "Support identification of required viewpoints" is constrained by the need to "Be consistent with overall needs of the architectural framework". The Viewpoints identified as being required in the Architectural Framework must meet the needs for the Architectural Framework as defined on the AF Context Viewpoint.

- The need to "Support identification of grouping of viewpoints into perspectives" is constrained by the need to "Be consistent with ontology areas defined for ontology". While it is not essential that the Perspectives identified on the Viewpoint Relationships Viewpoint should be the same as the groupings of Ontology Elements, as identified on the Ontology Definition Viewpoint, keeping them in accord with each other can help with consistency and to ensure that the Viewpoints defined cover the whole of the Ontology.

As identified on Figure 1, the two key Stakeholder Roles involved in the main need to "Support identification of required viewpoints" are (through the «include» relationships from "Define an architectural framework for creating architectural frameworks"): the Architectural Framework Sponsor and the Architectural Framework Modeller. The Domain Expert is also a key Stakeholder Role, with an interest in the need to "Be consistent with ontology areas defined on for ontology".

Viewpoint Definition View

The definition of the Viewpoint Relationships Viewpoint (VRV) is shown in the following diagram, a CAFF Viewpoint Definition View.

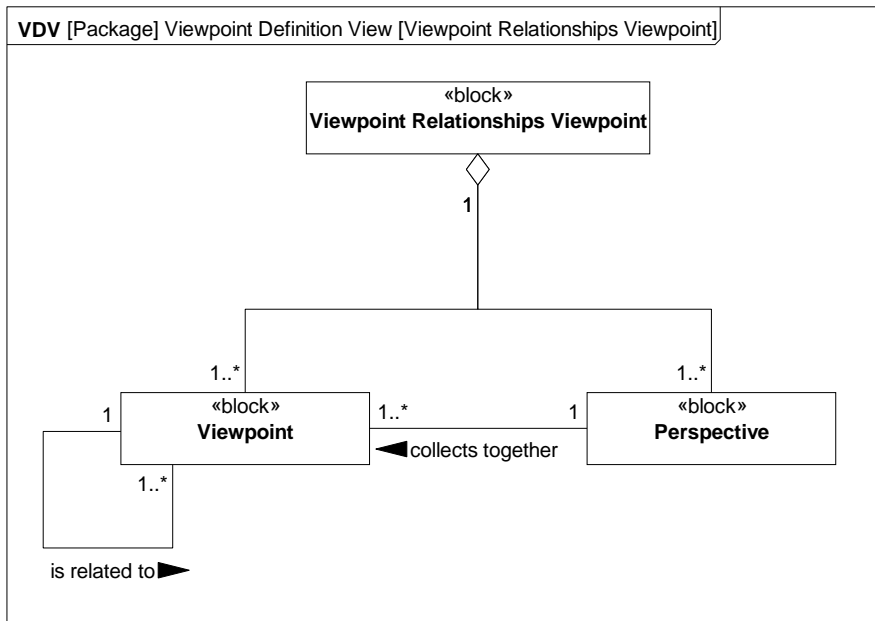


Figure 11 - Viewpoint Definition View for the Viewpoint Relationships Viewpoint

Figure 11 addresses the needs defined on Figure 10. It defines the content for the Viewpoint Relationships Viewpoint.

The Viewpoint Relationships Viewpoint is made up of one or more Viewpoints and shows the relationships between them. It is also made up of one or more Perspectives and shows which Viewpoints are in which Perspective.

Example View

For an example of a Viewpoint Relationships Viewpoint see Figure 4.

Viewpoint Context Viewpoint (VCV)

This section defines the Viewpoint Context Viewpoint (VCV).

Viewpoint Context View

The needs that the Viewpoint Context Viewpoint (VCV) is intended to address are shown in the following diagram, itself a CAFF Viewpoint Context View.

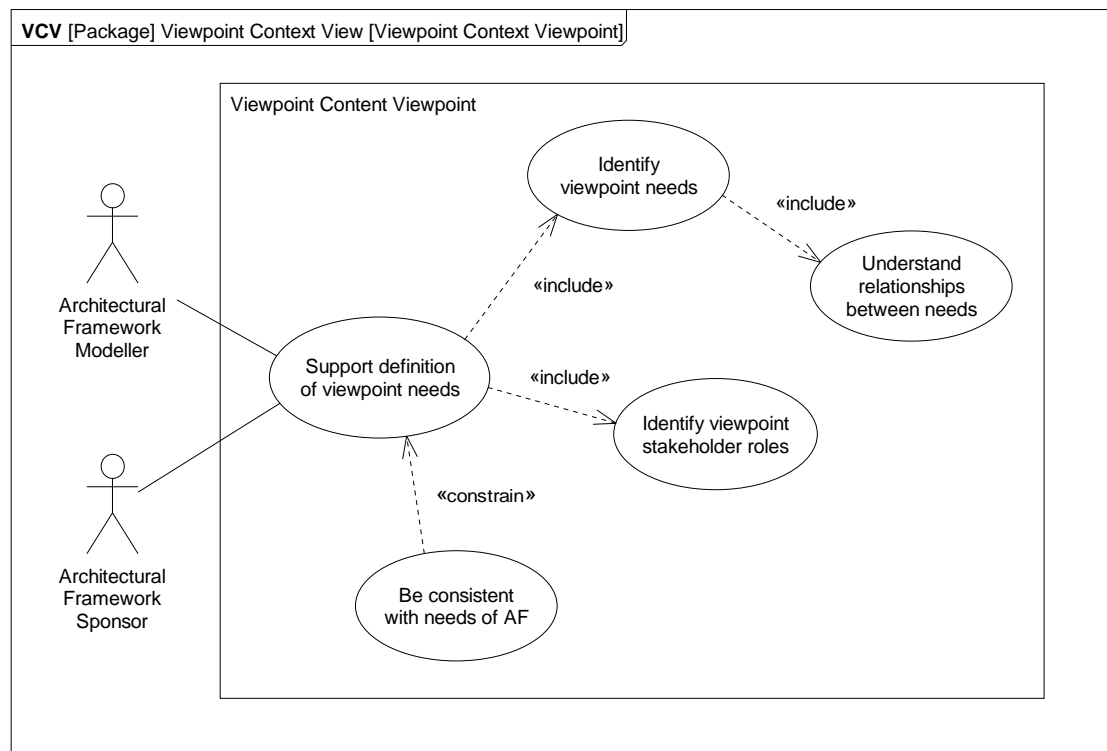


Figure 12 - Viewpoint Context View for the Viewpoint Context Viewpoint

Figure 12 shows the Viewpoint Context View for the Viewpoint Context Viewpoint. That is, it defines the needs that the Viewpoint Context Viewpoint must address, together with relevant Stakeholder Roles.

The main need, taken from the Architectural Framework Context (see Figure 1) is to "Support definition of viewpoint needs"; the Viewpoint Context Viewpoint exists to capture the needs of a Viewpoint being defined. In order to do this, it is necessary to "Be consistent with needs of AF" and to be able to:

- "Identify viewpoint needs" - Identify the needs that the Viewpoint is being created to address.
- "Understand relationships between needs" - Understand any relationships *between* the needs that the Viewpoint is being created to address.
- "Identify viewpoint stakeholder roles" - Identify the Stakeholder Roles involved in the definition of the Viewpoint that have an interest in or are affected by the identified needs.

As identified on Figure 1, the two key Stakeholder Roles involved are (through the «include» relationships from "Define an architectural framework for creating

architectural frameworks"): the Architectural Framework Sponsor and the Architectural Framework Modeller.

Viewpoint Definition View

The definition of the Viewpoint Context Viewpoint (VCV) is shown in the following diagram, a CAFF Viewpoint Definition View.

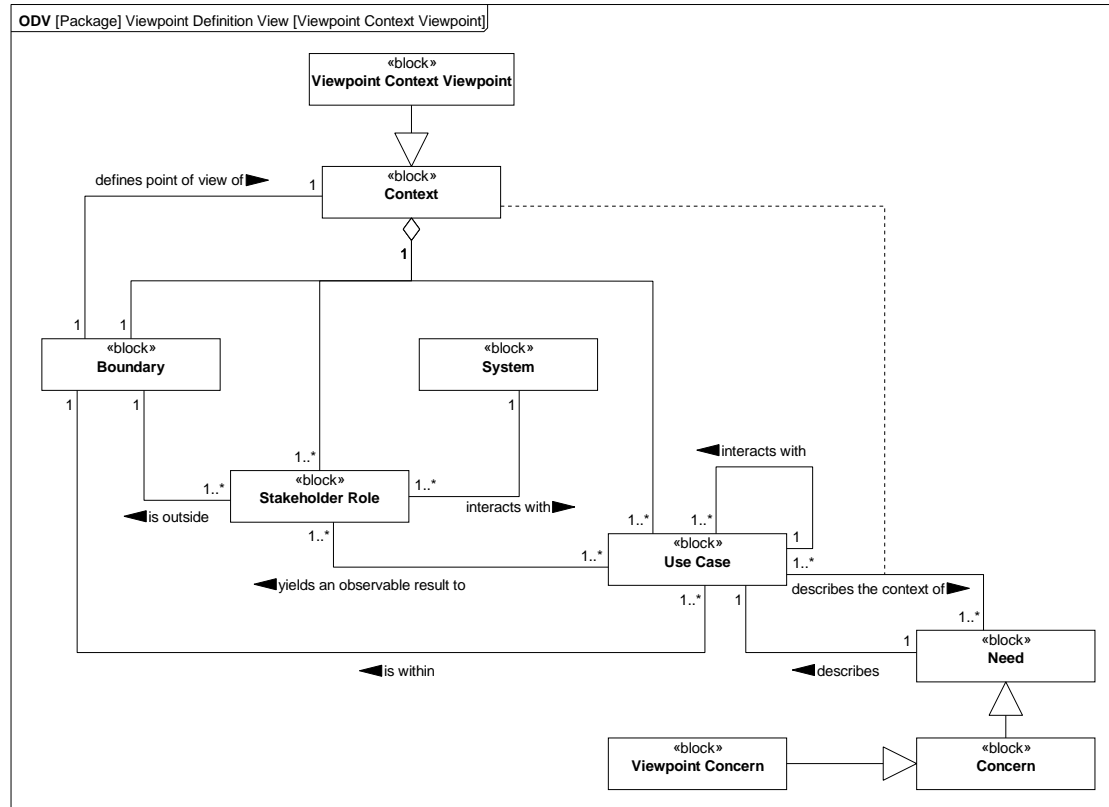


Figure 13 - Viewpoint Definition View for the Viewpoint Context Viewpoint

Figure 13 addresses the needs defined on Figure 12.

It defines the content of the Viewpoint Context Viewpoint, which defines the Context for a Viewpoint. It is made up of a Boundary, one or more Stakeholder Roles, that are outside the Boundary, and one or more Use Cases that are inside the Boundary.

Each Use Case represents a Viewpoint Concern (a type of Need) for a Viewpoint, that yields observable results to one or more Stakeholder Roles. Each Use Case may interact with a number of other Use Cases.

Example View

For an example of a Viewpoint Context View, see Figure 12. This is an example of a VCV that defines the context for itself. For an example without self-referentiality, see Figure 10.

Viewpoint Definition Viewpoint (VDV)

This section defines the Viewpoint Definition Viewpoint (VDV).

Viewpoint Context View

The needs that the Viewpoint Definition Viewpoint (VDV) is intended to address are shown in the following diagram, a CAFF Viewpoint Context View.

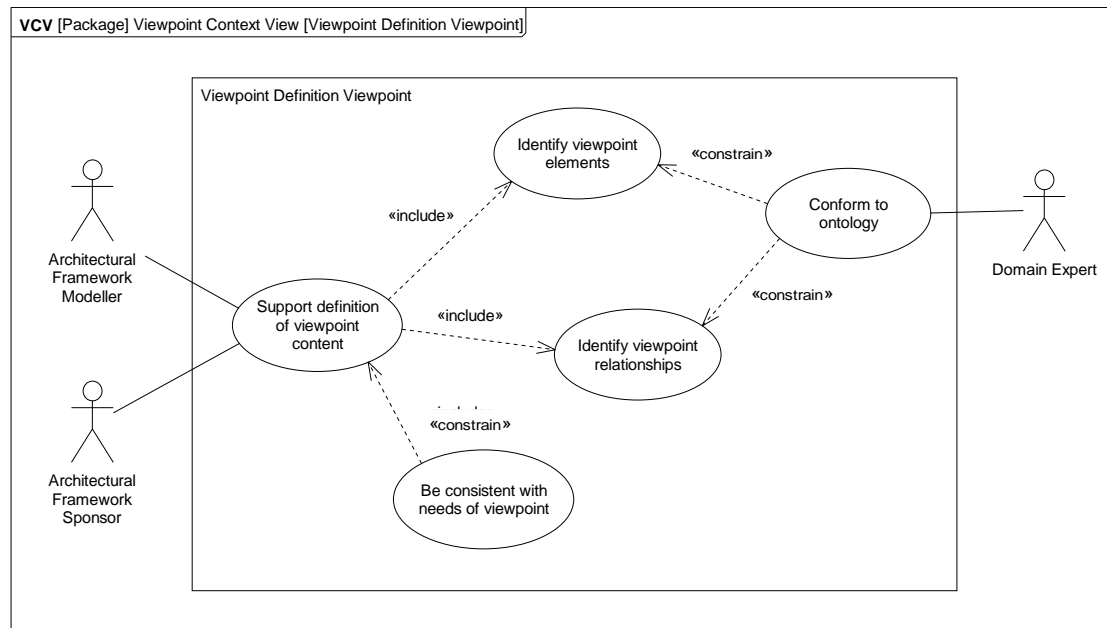


Figure 14 - Viewpoint Context View for the Viewpoint Definition Viewpoint

Figure 14 shows the Viewpoint Context View for the Viewpoint Definition Viewpoint. That is, it defines the needs that the Viewpoint Definition Viewpoint must address, together with relevant Stakeholder Roles.

The main need, taken from the Architectural Framework Context (see Figure 1) is to "Support definition of viewpoint content"; the Viewpoint Definition Viewpoint exists to define the contents of a Viewpoint. In order to do this, it is necessary to "Be consistent with needs of viewpoint". That is, the Viewpoint must be defined in such a way that it meets its needs, described on its associated Viewpoint Context Viewpoint.

The main needs that must be addressed are to:

- "Identify viewpoint elements" - Identify the Viewpoint Elements that will appear on the Viewpoint.
- "Identify viewpoint relationships" - Identify any relationships *between* the Viewpoint Elements that appear on the Viewpoint.

Both of these needs are constrained by the need to:

- "Conform to ontology" - Every Viewpoint Element (and relationship) that can appear on a Viewpoint *must* correspond to an Ontology Element (or

relationship) from the Ontology. Nothing can appear on a Viewpoint that does not exist on the Ontology.

As identified on Figure 1, the two key Stakeholder Roles involved in the main need to "Support definition of viewpoint content" are (through the «include» relationships from "Define an architectural framework for creating architectural frameworks"): the Architectural Framework Sponsor and the Architectural Framework Modeller. The Domain Expert role is also involved, given the constraint imposed by the need to "Conform to ontology".

Viewpoint Definition View

The definition of the Viewpoint Definition Viewpoint (VDV) is shown in the following diagram, itself a CAFF Viewpoint Definition View.

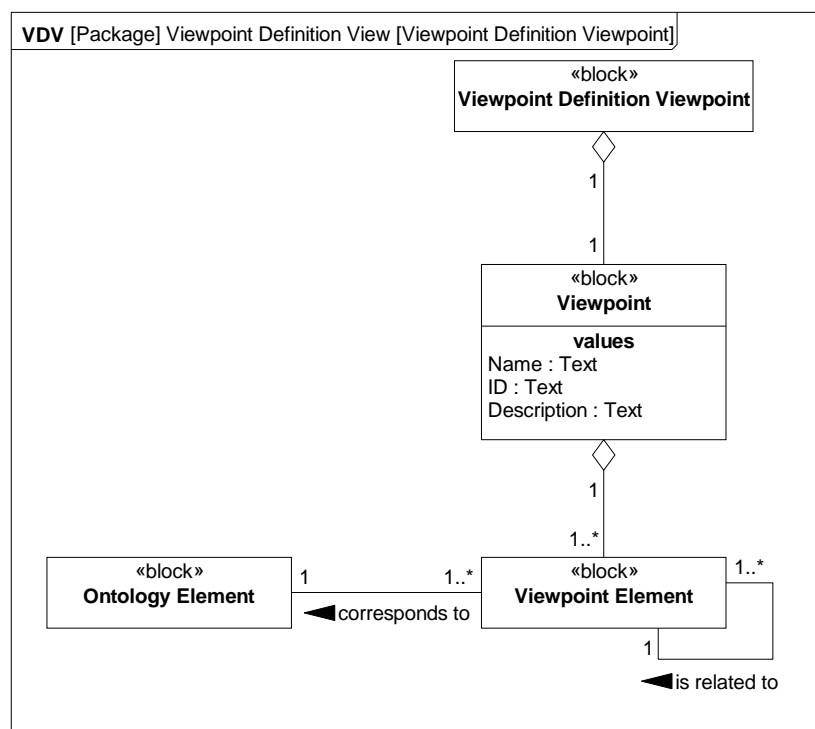


Figure 15 - Viewpoint Definition View for the Viewpoint Definition Viewpoint

Figure 15 addresses the needs defined on Figure 14. It defines the content of the Viewpoint Definition Viewpoint.

The Viewpoint Definition Viewpoint defines a single Viewpoint, giving it a Name, an ID and a Description, and showing the Viewpoint Elements that make up the Viewpoint and the relationships between them. In most cases there will be a one-to-one mapping between Viewpoint Elements and the Ontology Elements that they correspond to and in such cases the same name is usually used so that a Viewpoint can, for all practical purposes, be considered to be made up of those Ontology Elements. However, this is not always the case; in some Viewpoints a single Ontology Element is represented on the Viewpoint by multiple Viewpoint Element. In these cases, the Viewpoint Definition Viewpoint shows the relationships between Viewpoint Elements and Ontology Elements.

This diagram has a subtlety that, at first viewing, may be missed; the Viewpoint Definition Viewpoint *is its own* View. To understand this, consider the following; to define a Viewpoint, it is necessary to create an instance (i.e. a View) of the Viewpoint Definition Viewpoint for the Viewpoint being defined. That is, one creates a Viewpoint Definition View that conforms to the Viewpoint Definition Viewpoint for the Viewpoint being defined. Thus, the Viewpoint Definition View created identifies the Viewpoint and the Viewpoint Elements that make it up.

- For example, Figure 9 - Viewpoint Definition View for the Ontology Definition Viewpoint, is a Viewpoint Definition View (a View), conforming to the Viewpoint Definition Viewpoint (a Viewpoint), that defines what can appear on the Ontology Definition Viewpoint.
- In a similar way, Figure 15 - Viewpoint Definition View for the Viewpoint Definition Viewpoint, is a Viewpoint Definition View (a View), conforming to the Viewpoint Definition Viewpoint (a Viewpoint), that defines what can appear on the Viewpoint Definition Viewpoint. Thus the View and the Viewpoint for the Viewpoint Definition Viewpoint are essentially one and the same; the Viewpoint Definition Viewpoint is self-referential. It defines itself and hence an instance (a View) is its own Viewpoint and vice versa.

Example View

For an example of a Viewpoint Definition View, see Figure 15. This is an example of a VDV that defines its own content. For an example without self-referentiality, see Figure 13.

Rules Definition Viewpoint (RDV)

This section defines the Rules Definition Viewpoint (RDV).

Viewpoint Context View

The needs that the Rules Definition Viewpoint (RDV) is intended to address are shown in the following diagram, a CAFF Viewpoint Context View.

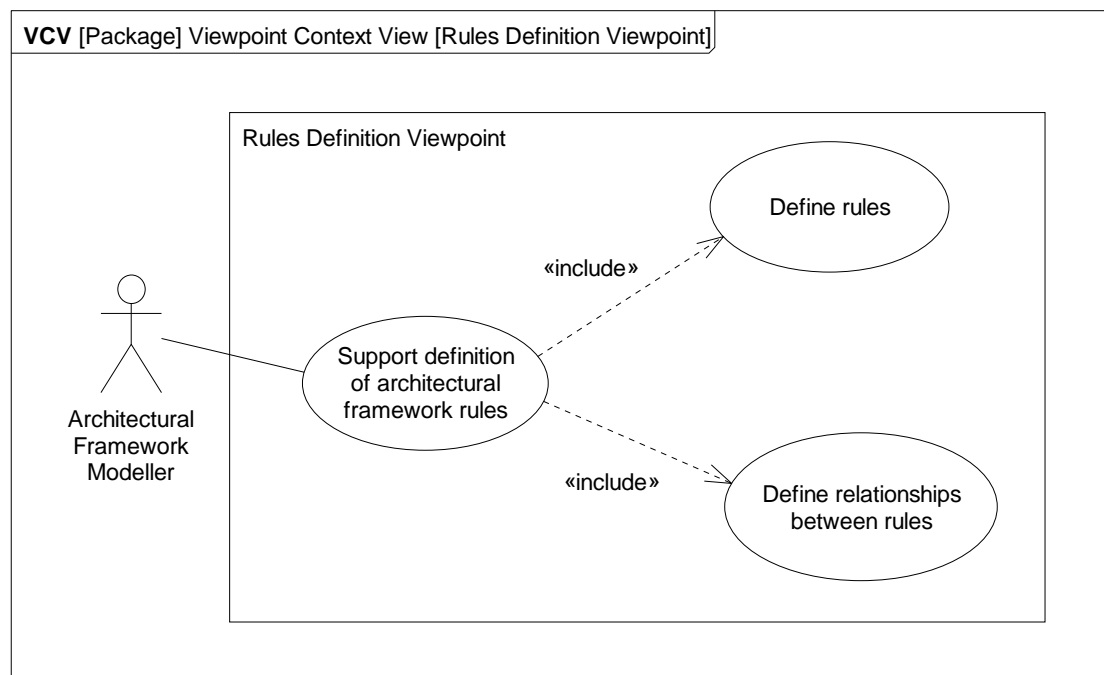


Figure 16 - Viewpoint Context View for the Rules Definition Viewpoint

Figure 16 shows the Viewpoint Context View for the Rules Definition Viewpoint. That is, it defines the needs that the Rules Definition Viewpoint must address, together with relevant Stakeholder Roles.

The main need, taken from the Architectural Framework Context (see Figure 1) is to "Support definition of architectural framework rules"; the Rules Definition Viewpoint exists to define any Rules that constrain the Architectural Framework. Note that such Rules can constrain *any* aspect of the Architectural Framework. The main needs that must be addressed are to:

- "Define rules" - Define any Rules which constrain the Architectural Framework.
- "Define relationships between rules" - Define any relationships *between* the Rules. This allows complex Rules to be built up.

As identified on Figure 1, the key Stakeholder Role involved in the main need to "Support definition of architectural framework rules" is (through the «include» relationships from "Define an architectural framework for creating architectural frameworks"): the Architectural Framework Modeller.

Viewpoint Definition View

The definition of the Rules Definition Viewpoint (RDV) is shown in the following diagram, a CAFF Viewpoint Definition View.

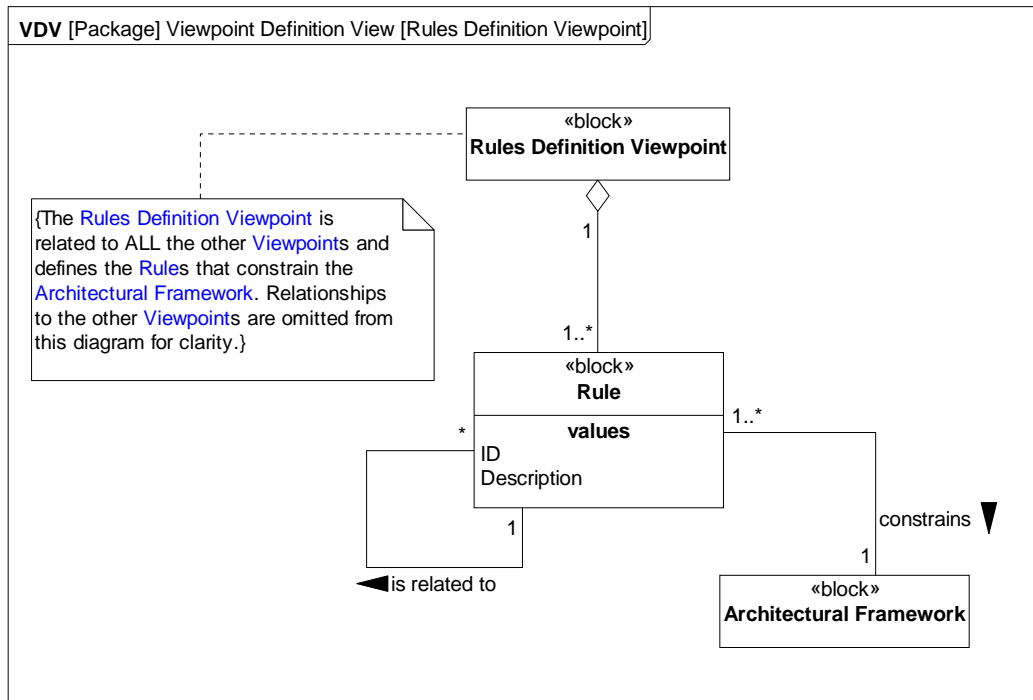


Figure 17 - Viewpoint Definition View for the Rules Definition Viewpoint

Figure 17 addresses the needs defined on the Figure 16.

It defines the Rules that constrain the Architectural Framework. Each Rule has an ID and a Description. A Rule may be related to zero or more other Rules. Note that Rules constrain the entire Architectural Framework and hence can constrain any of the Viewpoints that make up an Architectural Framework.

Example View

For an example of a Rules Definition View see Figure 5.

4 Summary

This deliverable defines an Architectural Framework (AF) that can be used in the definition of other AFs. This COMPASS AF Framework (the CAFF) is presented in Section 3. Processes for using the CAFF are presented in Appendix A.

The CAFF has been used in a number of areas of the COMPASS project:

- An example AF for architectural design that has been defined using the CAFF can be found in Appendix B of this document
- The Fault Modelling Architectural Framework (FMAF) has been defined using CAFF and can be found in [D24.2 2013]
- The enabling patterns found in [D22.6 2014] have all been documented using the CAFF.
- An industrial application of the CAFF, the Streaming Architectural Framework (SAF) has been produced by B&O as part of their case study work and can be found in [D42.2 2014].

It is also worth noting that the authors are successfully using the CAFF approach with a number of industrial customers in the automotive domain.

5 References

- [D21.1 2012] COMPASS. 'D21.1 – Report on Guidelines for SoS Requirements'. COMPASS Project; 2012.
- [D21.2 2013] COMPASS. 'Initial Report on Guidelines for Architectural Level SoS Modelling'. COMPASS Project; 2013.
- [D21.5 2014] COMPASS. 'Final Report on Guidelines for Architectural Level SoS Modelling'. COMPASS Project; 2014.
- [D22.1 2012] COMPASS. 'D22.1 – Initial Report on SoS Architectural Models'. COMPASS Project; 2012.
- [D22.3 2013] COMPASS. 'D22.3 – Report on Modelling patterns for SoS Architectures'. COMPASS Project; 2013.
- [D22.6 2014] COMPASS. 'D22.6 – Final Report on SoS Architectural Models'. COMPASS Project; 2014.
- [D24.2 2013] COMPASS. 'D24.2 – Report on Timed Fault Tree Analysis - Fault Modelling'. COMPASS Project; 2013.
- [D42.2 2014] COMPASS. 'A/V/HA Ecosystems Prototype Using COMPASS Methods & Tools'. COMPASS Project; 2014.
- [Holt & Perry 2008] Holt, J. & Perry, S. 'SysML for Systems Engineering' IET Publishing, 2008
- [Holt & Perry 2013] Holt, J. & Perry, S. 'SysML for Systems Engineering – 2nd Edition: A model-based approach' IET Publishing, 2013
- [Holt et al. 2011] Holt, J., Perry, S. & Brownsword, M. 'Model-based requirements engineering', IET Publishing 2011.
- [ISO42010 2011] ISO/IEC. 'ISO/IEC 42010:2011 Systems and software engineering — Architecture description'. International Organisation for Standardisation; 2011
- [MODAF 2010] The Ministry of Defence Architectural Framework, 2010: <https://www.gov.uk/mod-architecture-framework> (Accessed February 2013)
- [SysML 2010] OMG. SysML OMG Specification, Object Management Group SysML website: <http://www.omgsysml.org> (Accessed February 2013)
- [TRAK 2013] TRAK. 'TRAK – Enterprise Architecture Framework'. Available from <http://trak.sourceforge.net/> (Accessed February 2013)

Appendix A Processes for AF Definition and Construction

This appendix presents the model for the architectural framework definition processes. These processes have been defined according to the ‘seven views’ approach to process modelling.

A.1 Requirements Context View

This section presents the requirements context view (RCV) for the architectural framework definition processes.

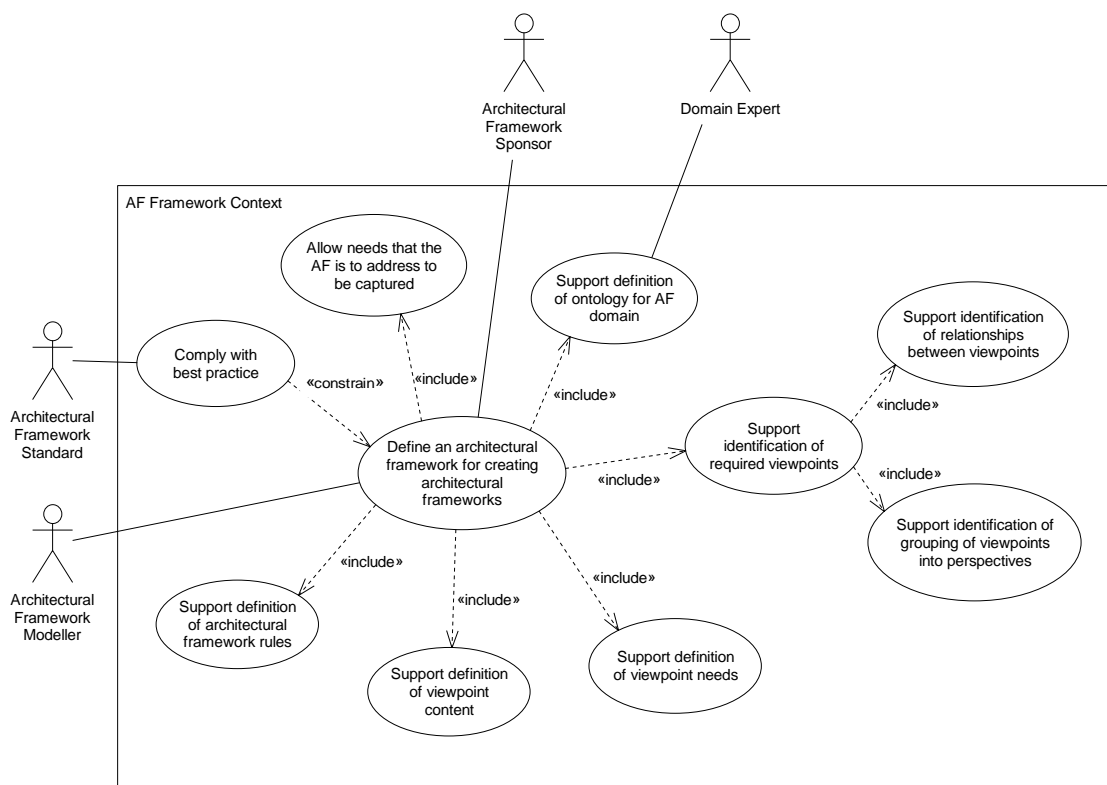


Figure 18 - Requirements context view (RCV) for the architectural framework definition processes

The main need that must be fulfilled is to "Define an architectural framework for creating architectural frameworks", constrained by the need to "Comply with best practice" such as Architectural Framework Standards (for example, [ISO42010 2011]). In order to "Define an architectural framework for creating architectural frameworks" it is necessary to:

- "Allow needs that the AF is to address to be captured" - When defining an Architectural Framework, it is important that the needs that the Architectural Framework is to address can be captured, in order to ensure that the Architectural Framework is fit for purpose.
- "Support definition of ontology for AF domain" - When defining an Architectural Framework, it is essential that the concepts, and the relationships between them, are defined for the domain in which the Architectural Framework is to be used. This is the Ontology that forms the foundational

basis of the definition of the Architectural Framework's Viewpoints. Such an Ontology ensures the *consistency* of the Architectural Framework. The Architectural Framework must support such a definition of an Ontology.

- "Support identification of required viewpoints" - The Viewpoints that make up the Architectural Framework need to be identified. As well as supporting such an identification, the Architectural Framework must also "Support identification of relationships between viewpoints" and "Support identification of grouping of viewpoints into perspectives".
- "Support definition of viewpoint needs" - In order to define the Viewpoints that make up an Architectural Framework, it is essential that the needs of each Viewpoint be clearly understood in order to ensure each Viewpoint is fit for purpose and that the Viewpoints defined meet the overall needs for the Architectural Framework.
- "Support definition of viewpoint content" - An Architectural Framework is essentially a number of Viewpoints that conform to an Ontology. Therefore, when defining an Architectural Framework it is essential that each Viewpoint can be defined in a consistent fashion that ensures its conformance to the Ontology.
- "Support definition of architectural framework rules" - Often, when defining an Architectural Framework, it is often necessary to constrain aspects of the Architectural Framework through the definition of a number of constraining Rules. It is therefore essential that an AF Framework supports the definition of such Rules.

A.2 Stakeholder View

This section presents the stakeholder view (SV) for the architectural definition processes.

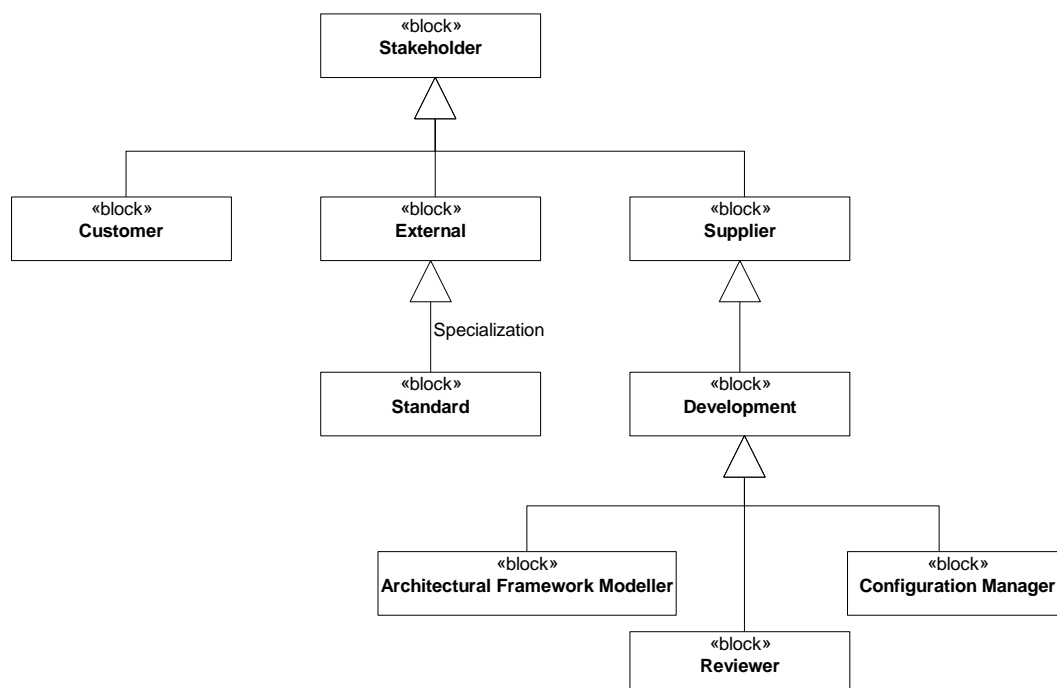


Figure 19 - Stakeholder view (SV) for the architectural definition processes

The key Stakeholder Roles involved are:

- ‘Architectural Framework Modeller’ - the role involved in the modelling and definition of an Architectural Framework.
- ‘Standard’ - the role of any appropriate standard for Architectural Frameworks. An example of a standard that could fill this role would be [ISO42010 2011].
- ‘Reviewer’ - this role is essential for all aspects of MBSE that covers both mechanical review (straightforward verification review that does not require any real human input but simply executes a pre-defined rule) and human reviews (that require reasoning and will tend to be qualitative and are typically very difficult, if not impossible to automate using a tool).
- ‘Configuration Manager’ - this role is responsible for ensuring that the model is correctly controlled, managed and configured.

A.3 Process Content View

This section presents the process content view (PCV) for the architectural framework definition processes.

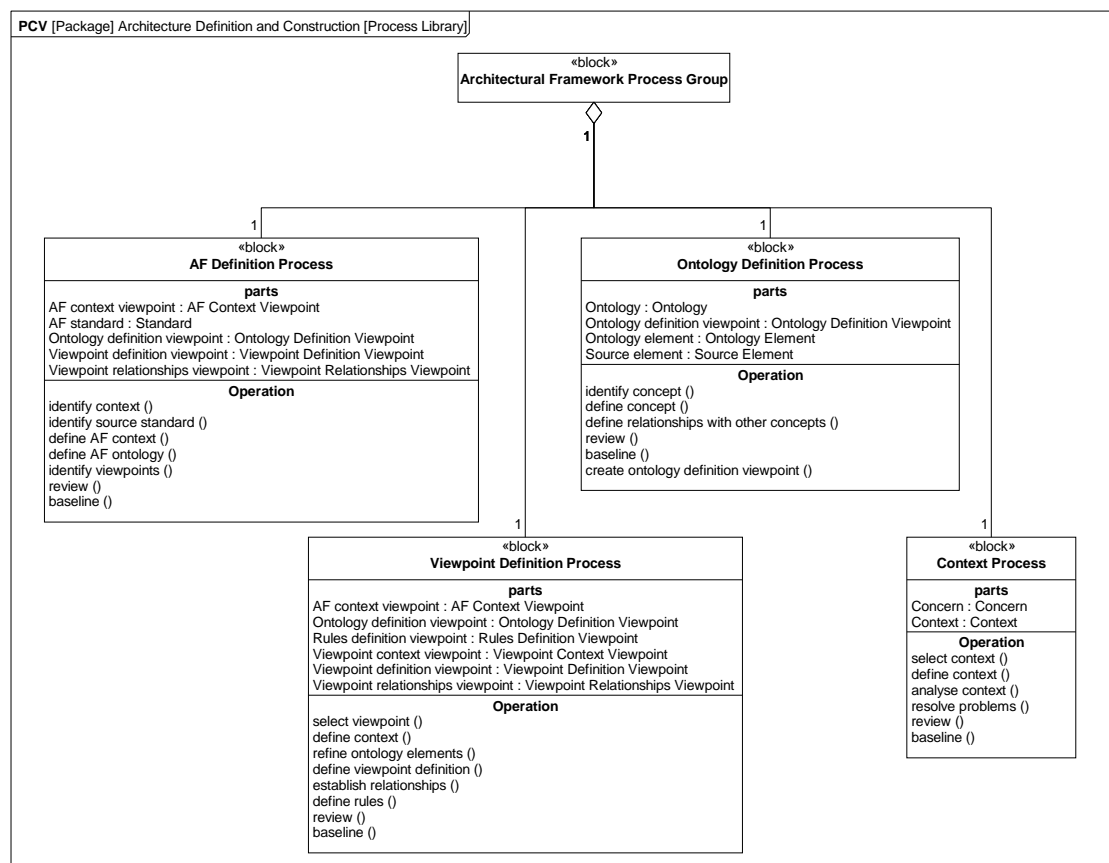


Figure 20 - Process content view (PCV) for the architectural framework definition processes

The PCV for the architectural framework definition processes consistent of four processes, that are detailed as follows.

‘AF Definition Process’. The aim of this process is to understand the underlying need for the AF.

«block» AF Definition Process
parts AF context viewpoint : AF Context Viewpoint AF standard : Standard Ontology definition viewpoint : Ontology Definition Viewpoint Viewpoint definition viewpoint : Viewpoint Definition Viewpoint Viewpoint relationships viewpoint : Viewpoint Relationships Viewpoint
Operation identify context () identify source standard () define AF context () define AF ontology () identify viewpoints () review () baseline ()

Figure 21 - PCV for AF Definition Process

This process is made up of the following activities:

- ‘Identify context’ - This activity identifies the context of the architectural framework that is under development. For example, some architecture frameworks, such as MODAF, DoDAF, etc., have been developed in a acquisition context, whereas others are more aimed at development, such as Zachman, etc.
- ‘Identify source standard’ – This activity is concerned with identifying any source standards that the AF under development may need to comply with.
- ‘Define AF context’ - The context for the architecture framework is now created by invoking the ‘Context Process’ that will return the AF Context Viewpoint
- ‘Define AF ontology’ - Based on the AF Context Viewpoint, the ontology for the architecture framework under development is now defined by invoking the ‘Ontology Definition Process’ that returns the Ontology Definition Viewpoint’
- ‘Identify viewpoints’ - Based on the Ontology and the AF Context, a number of Viewpoints are now identified along with the relationships between them. This results in the Viewpoint Definition Viewpoint.
- ‘Review’ - The review activity is concerned with the following:
 - AF context viewpoint, where the context is reviewed for consistency in terms of its associated source standards, and that the Use Cases pass a sanity check.
 - Ontology definition viewpoint, where the ontology and its associated elements are reviewed to ensure that each ontology element has provenance, that all ontology elements have at least one relationship to another elements, and that the ontology can pass a sanity check.
 - Viewpoint definition viewpoint, where each viewpoint that has been identified is checked against the underlying ontology and that each has relationships to at least one other viewpoint.

- If the review is passed, then control is passed onto the 'baseline' activity, whereas if the review is failed, then control reverts back to the 'define AF context' activity
- 'Baseline' - All process artefacts: are recorded, are held under configuration control, form part of an overall baseline and are stored on the project repository

'Viewpoint Definition Process'. The main aim of this process is to identify the key viewpoints and to classify them into perspectives.

«block» Viewpoint Definition Process
parts
AF context viewpoint : AF Context Viewpoint Ontology definition viewpoint : Ontology Definition Viewpoint Rules definition viewpoint : Rules Definition Viewpoint Viewpoint context viewpoint : Viewpoint Context Viewpoint Viewpoint definition viewpoint : Viewpoint Definition Viewpoint Viewpoint relationships viewpoint : Viewpoint Relationships Viewpoint
Operation
select viewpoint () define context () refine ontology elements () define viewpoint definition () establish relationships () define rules () review () baseline ()

Figure 22 - PCV for Viewpoint Definition Process

This process consists of the following activities:

- 'Select viewpoint' - This activity selects a single Viewpoint from the Viewpoint Definition Viewpoint that is to be used as basis for this process
- 'Define context' - The context for the selected viewpoint is now created by invoking the 'Context Process' that will return the Viewpoint Context Viewpoint
- 'Refine ontology elements' - based on the ontology definition viewpoint, a subset of the ontology is now identified that is judged to be relevant from the selected viewpoint
- 'Define viewpoint definition' - Based on the subset of the ontology that was identified in the previous process, the ontology for the viewpoint under development is now defined by invoking the 'Ontology Definition Process' that returns the Viewpoint Definition Viewpoint
- 'Establish relationships' - Any relationships that exist between the selected viewpoints and any others are identified at this point in the form of the viewpoint relationship viewpoint. These will be used for both a sanity check against the original viewpoint definition viewpoint and as an input for defining rules
- 'Define rules' - A set of rules for ensuring consistency of the viewpoints is defined
- 'Review' - The review activity is concerned with the following:
 - Viewpoint Context Viewpoint, where the context is reviewed for consistency with the AF context definition view, internal consistency and a sanity check is performed.

- Viewpoint Definition Viewpoint, where the view is checked against the AF ontology for consistency and a sanity check is performed.
- Viewpoint relationship viewpoint, where consistency against the AF viewpoint definition view is checked.
- Rules Definition Viewpoint, where the rules are checked for consistency against the AF ontology and the viewpoint relationships view.
- If the review is passed and there are no more viewpoints to be defined, then control passes to the 'baseline activity'. If the review is passed and there are more viewpoints to be defined, then control reverts back to the 'select viewpoint' activity. If the review fails, then control reverts back to the 'define context' activity
- 'Baseline' - All process artefacts: are recorded, are held under configuration control, form part of an overall baseline and are stored on the project repository

'Ontology Definition Process'. The main aim of this process is to identify and define the main concepts and terms used for the AF in the form of an ontology.

«block» Ontology Definition Process
parts Ontology : Ontology Ontology definition viewpoint : Ontology Definition Viewpoint Ontology element : Ontology Element Source element : Source Element
Operation identify concept () define concept () define relationships with other concepts () review () baseline () create ontology definition viewpoint ()

Figure 23 - PCV for Ontology Definition Process

This process consists of the following activities:

- 'Identify concepts' - This activity identifies a concept that is relevant to the AF under development and that needs to be defined
- 'Define concepts' - The concept that has been identified in the previous activity is now defined and its provenance is established against a source elements. This concept now becomes an ontology element
- 'Define relationships with other concepts' - The concept that has been identified and defined in the previous steps is now related to other concepts. These concepts and relationships now become part of the ontology
- 'Create ontology definition viewpoint' – The ontology is collected together to form the Ontology Definition Viewpoint.
- 'Review' - The review activity is concerned with the ontology, where each element is checked for provenance, that it relates to at least one other ontology element and that it is consistent with the overall AF ontology. If the review is passed, then control continues on to the 'base-

line’ activity. If the review is failed, then control reverts back to the ‘identify concept’ activity

- ‘Baseline’ - All process artefacts: are recorded, are held under configuration control, form part of an overall baseline and are stored on the project repository

‘Context Process’. The main aim of this process is to create a context that can be used to create either an ‘AF Context View’ or a ‘Viewpoint Context View’.

<p>«block» Context Process</p>
<p>parts Concern : Concern Context : Context</p>
<p>Operation select context () define context () analyse context () resolve problems () review () baseline ()</p>

Figure 24 - PCV for Context Process

This process consists of the following activities:

- ‘Select context’ - Based on the elements in the views, this activity identifies the source of the context. The context represents the needs of the system from a particular point of view and, therefore, must have a context source defined
- ‘Define context’ - This activity will produce a context that: identifies Use Cases based on the Needs, identifies the system boundary, identifies stakeholders that are external to this system boundary, identifies relationships between the Use Cases and the external stakeholders and identifies relationships between Use Cases
- ‘Analyse context’ – This activity considers the context and ensures that all Use Cases, stakeholders and their relationships are understood.
- ‘Resolve problems’ – This activity resolves any problems that may have been identified during the context analysis.
- ‘Review’ - The review activity is concerned with the Concern and Context that are generated by the process.
- ‘Baseline’ - All process artefacts: are recorded, are held under configuration control, form part of an overall baseline and are stored on the project repository

A.4 Process Behaviour Views

This section presents the process behaviour views (PBVs) for the architectural framework definition processes.

A.4.1 PBV – AF Definition Process

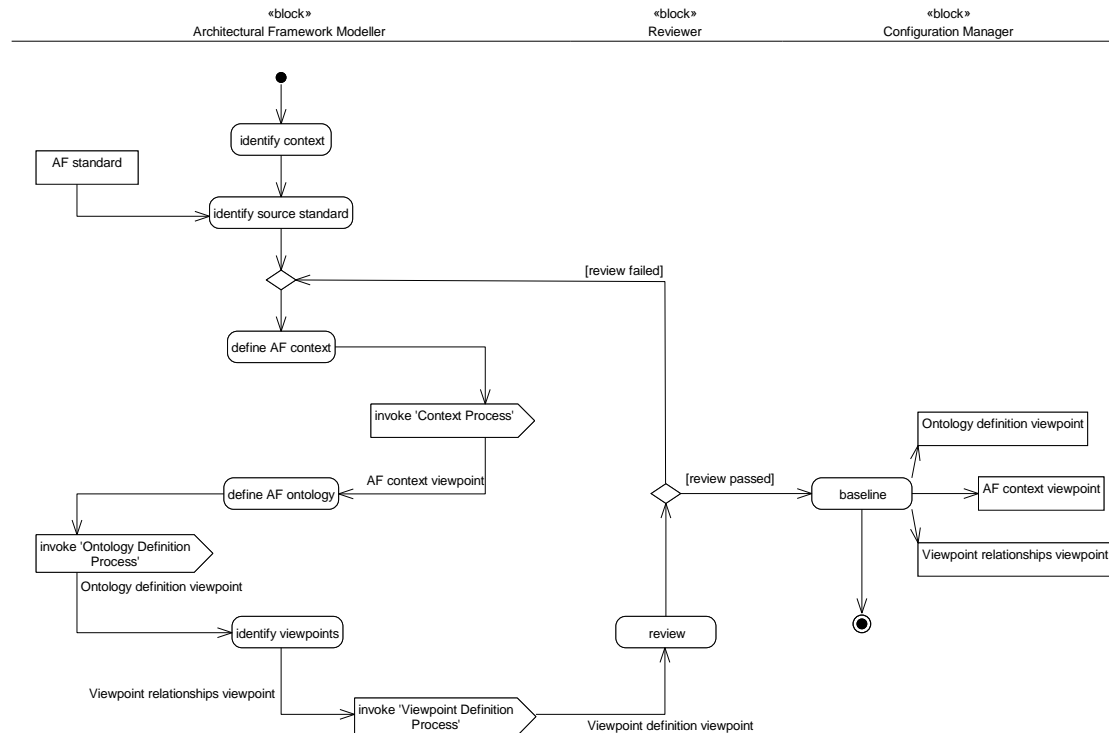


Figure 25 - PBV for AF Definition Process

The AF Definition Process begins with the Architectural Framework Modeller who identifies the context and then the source standard. Next, the AF context is defined and the Context Process is invoked. This process returns the AF Context Viewpoint and, based on this, the AF ontology is defined by invoking the Ontology Definition Process. This process returns the Ontology Definition Viewpoint and, based on this, the viewpoints are identified by invoking the Viewpoint Definition Process. This process returns the Viewpoint Definition Viewpoints and then the Reviewer reviews all artefacts. If the outcome of the review is positive all process artefacts are baselined by the Configuration Manager. If, however, the outcome is not positive, then the process reverts to defining the AF context.

A.4.3 PBV – Ontology Definition Process

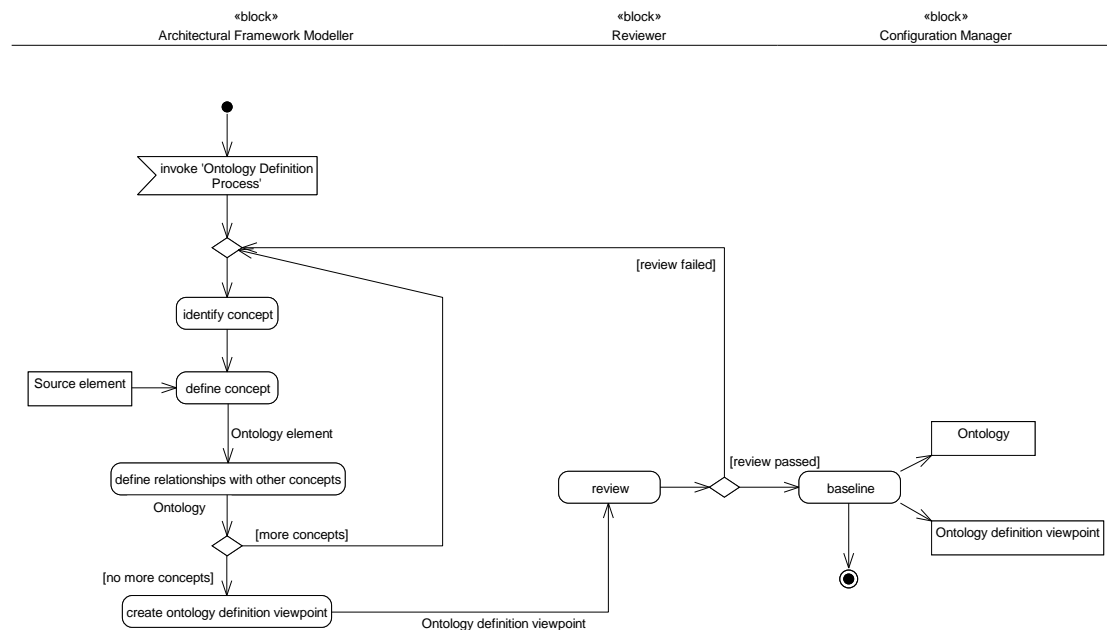


Figure 27 - PBV – Ontology Definition Process

The Ontology Definition Process begins with the Architectural Framework Modeller identifying a concept in the form of an ontology element, which is then defined based on a Source Element. Relationships with other concepts and the ontology updated. If there are more concepts to define, then the process reverts back to the beginning, otherwise the Ontology Definition Viewpoint is created.

The Reviewer reviews all artefacts. If the outcome of the review is positive all process artefacts are baselined by the Configuration Manager. If, however, the outcome is not positive, then the process reverts to the beginning.

A.4.4 PBV – Context Process

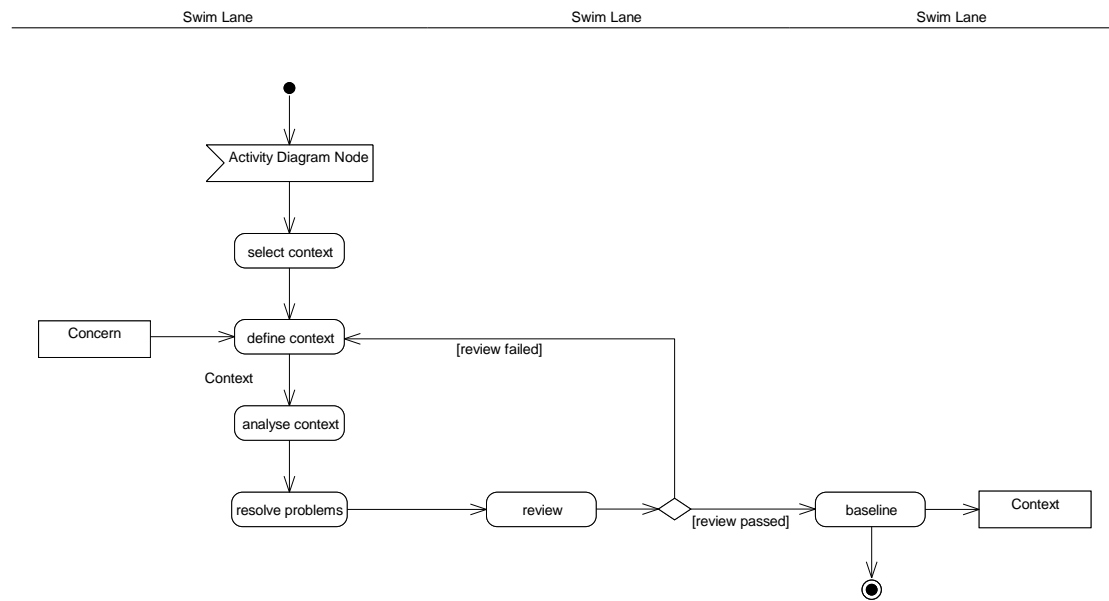


Figure 28 - PBV – Context Process

The Context Process begins with the Architectural Framework Modeller selecting a context which is then defined based on concerns. This context is then analysed and any problems resolved.

The Reviewer reviews all artefacts. If the outcome of the review is positive all process artefacts are baselined by the Configuration Manager. If, however, the outcome is not positive, then the process reverts to the beginning.

A.5 Information Views

This section presents the information views for the architectural framework definition processes. For a full description of each artefact, see Section 2 of this document.

A.5.1 IV – AF Definition process

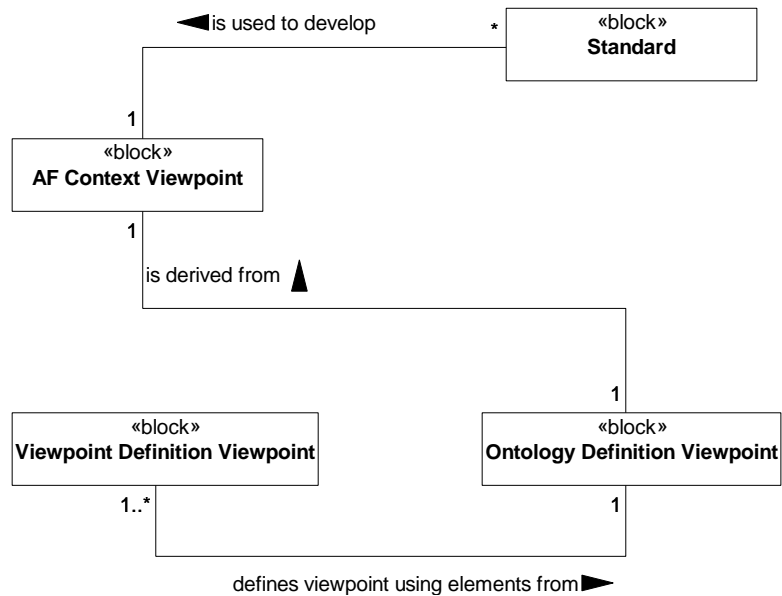


Figure 29 - IV – AF Definition process

A.5.2 IV – Viewpoint Definition Process

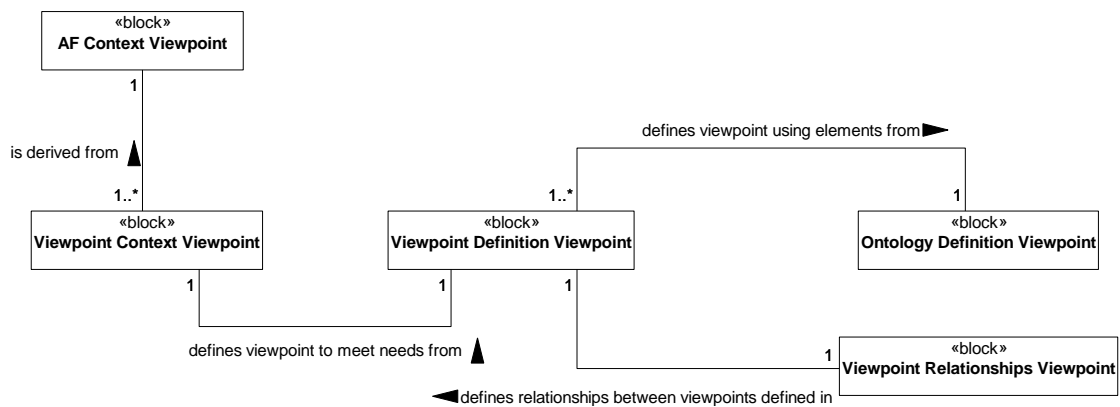


Figure 30 - IV – Viewpoint Definition Process

A.5.3 IV – Ontology Definition Process

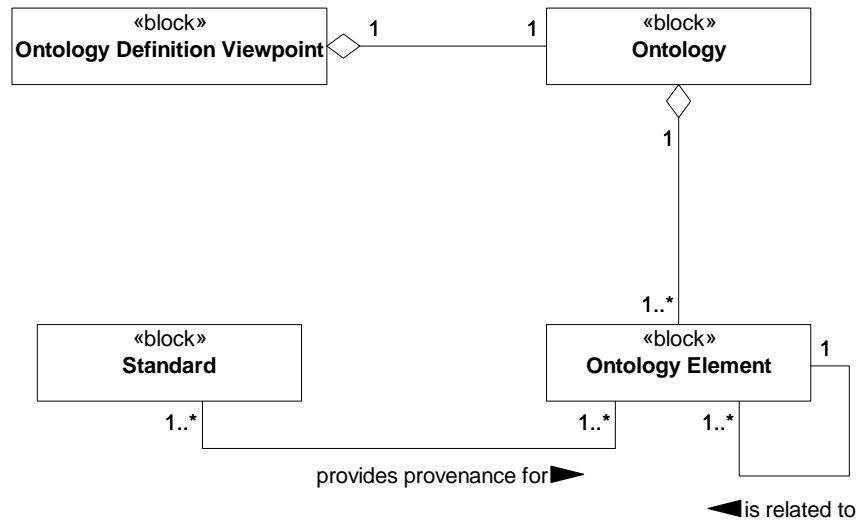


Figure 31 - IV – Ontology Definition Process

A.5.4 IV – Context Process

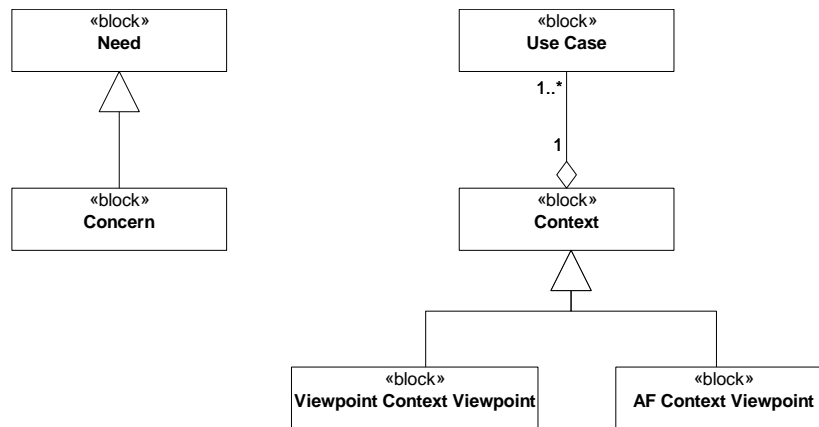


Figure 32 - IV – Context Process

Appendix B – Architectural Design Framework

This appendix presents an Architectural Design Framework that can be applied at two levels – the System of Systems level and the Constituent System level. This Architectural Design Framework is an example of an AF that has been created and defined using the CAFF.

B.1 AF Context View

The basic needs of the Architectural Design Framework are shown in the AF Context View shown in the following diagram.

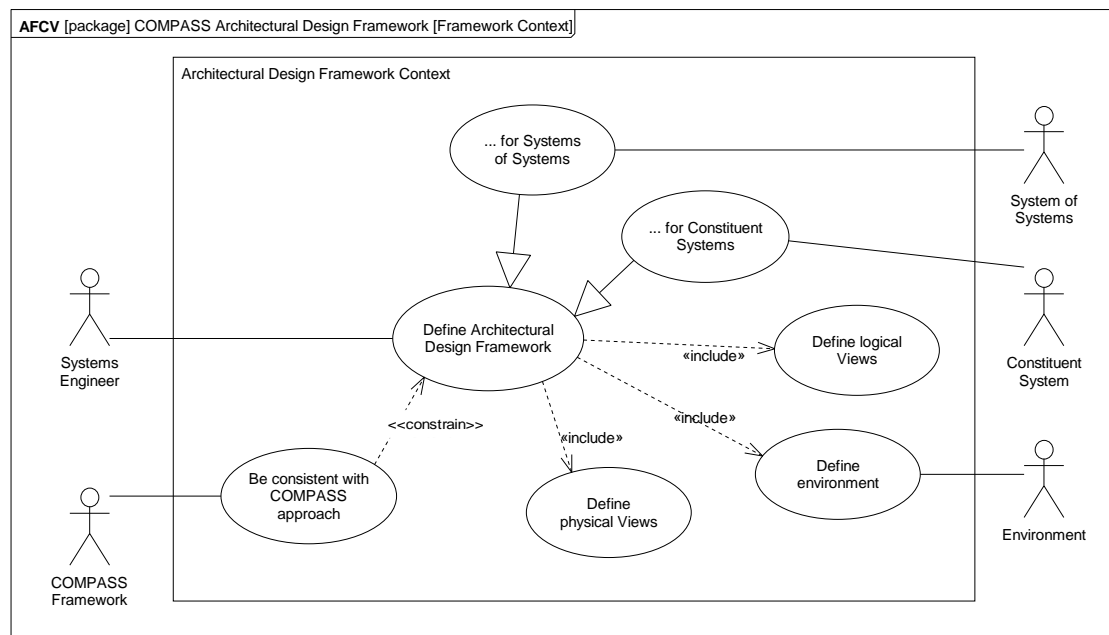


Figure 33 AF Context View for the Architectural Design Framework

The main *use case* shown in Figure 33 is to ‘Define Architectural Design Framework’ that must be applied at both the System of Systems level (‘... for Systems of Systems’) and the Constituent Systems level (‘... for Constituent Systems’). There are three main sub *use cases* which are to ‘Define logical Views’, to ‘Define physical Views’ and to ‘Define environment’. There is a single high-level constraint which is to ‘Be consistent with COMPASS approach’.

B.2 Ontology Definition View

The COMPASS Ontology has been expanded to include new concepts that are associated with Architectural Design, and it is shown in the following diagram.

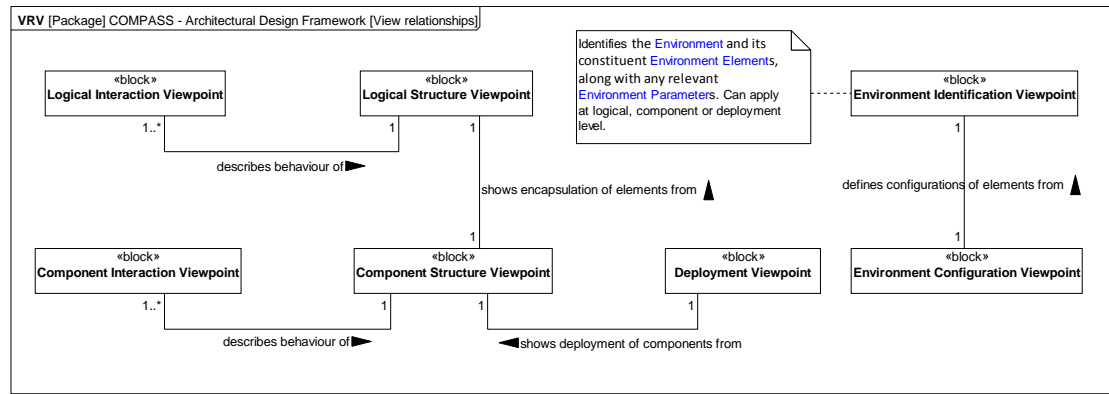


Figure 35 Viewpoint Relationship View for the Architectural Design Framework

The diagram in Figure 35 shows the relationships between the Viewpoints for the Architectural Design Framework in the form of a Viewpoints Relationship View.

B.4 Rules Definition View

The following Rules are defined:

ID	Rule Text
R1	At least one Logical Structure View and Logical Interaction View must be produced.
R2	For a System to be implemented, at least one Component Structure View and Component Interaction View must be produced.
R3	Before a System can be deplyed, at least one Deployment View, Environment Identification View and Environment Configuration View must be produced.

Note the use of a simple text table for the definition of these Rules, illustrating that the realisation of a CAFF Viewpoint need not be in a notation such as SysML. Note also the reference in the table to Views rather than Viewpoints. This is because an engineer using the Architecture Design Framework will produce Views based on the Viewpoints it defines.

B.5 Viewpoint Definitions

This section defines the five new Viewpoints that make up the Architectural Design Framework.

Logical Structure View

The Viewpoint Context View for the Logical Structure Viewpoint is shown in the following diagram.

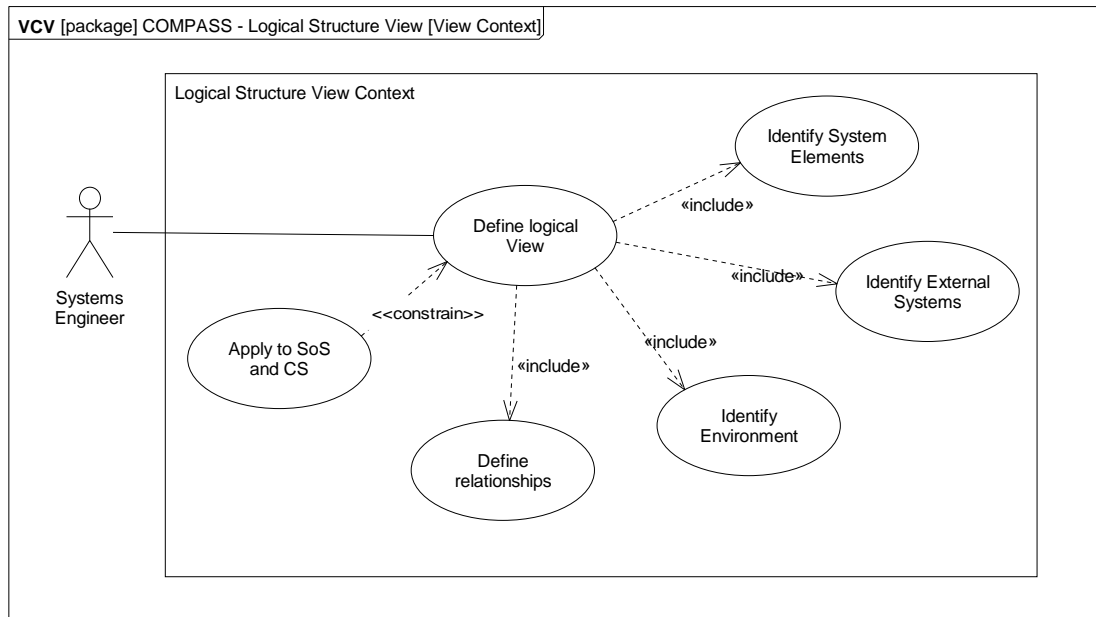


Figure 36 Viewpoint Context View for the Logical Structure Viewpoint

The diagram in Figure 36 shows the Viewpoint Context View for the Logical Structure Viewpoint that has a single main *use case*, which is to ‘Define logical View’. This has four sub *use cases* which are:

- ‘Identify System Elements’, which identifies the System Elements that make up the System. In the case of a System of Systems, these System Elements will be Constituent Systems, whereas in the case of a Constituent System they will be System Elements of that Constituent System.
- ‘Identify External Systems’, which involves identifying Systems that are not part of the System of Systems, but which are interacted with.
- ‘Identify Environment’, where the target deployment environment is identified in terms of the Environment itself and any relevant Environment Elements.
- ‘Define relationships’, where any relationships between the System, System Elements, Environment and External Systems are defined.

Based on this Context, the following subset of the COMPASS Ontology has been identified as relevant to the Architectural Design Framework.

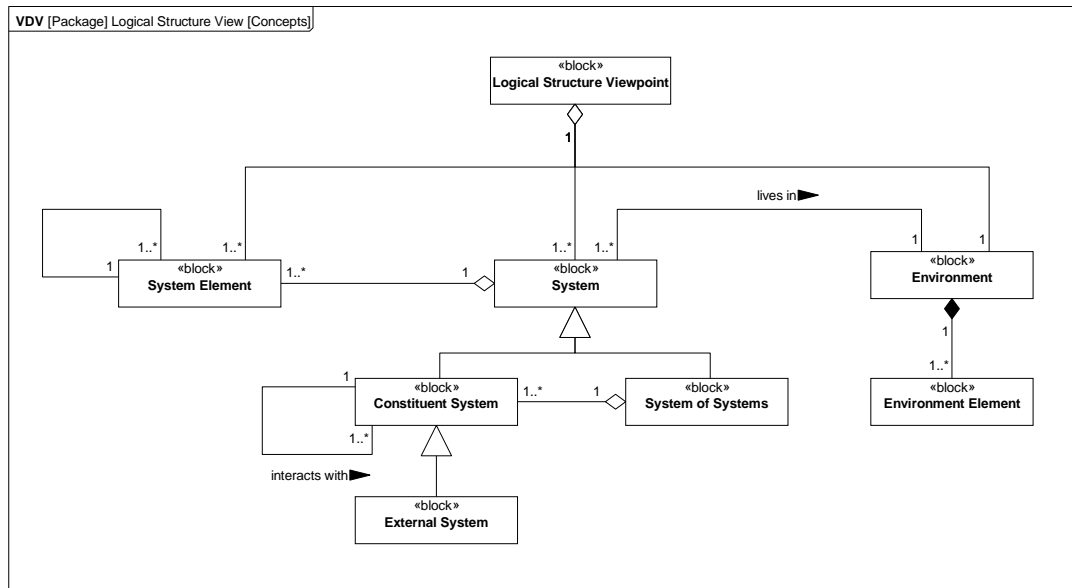


Figure 37 Viewpoint Definition View for the Logical Structure Viewpoint

The diagram in Figure 37 shows the Ontology Elements that are relevant for the Logical Structure Viewpoint that will form the basis for the Logical Structure Views.

Logical Interaction View

The Viewpoint Context View for the Logical Interaction Viewpoint is shown in the following diagram.

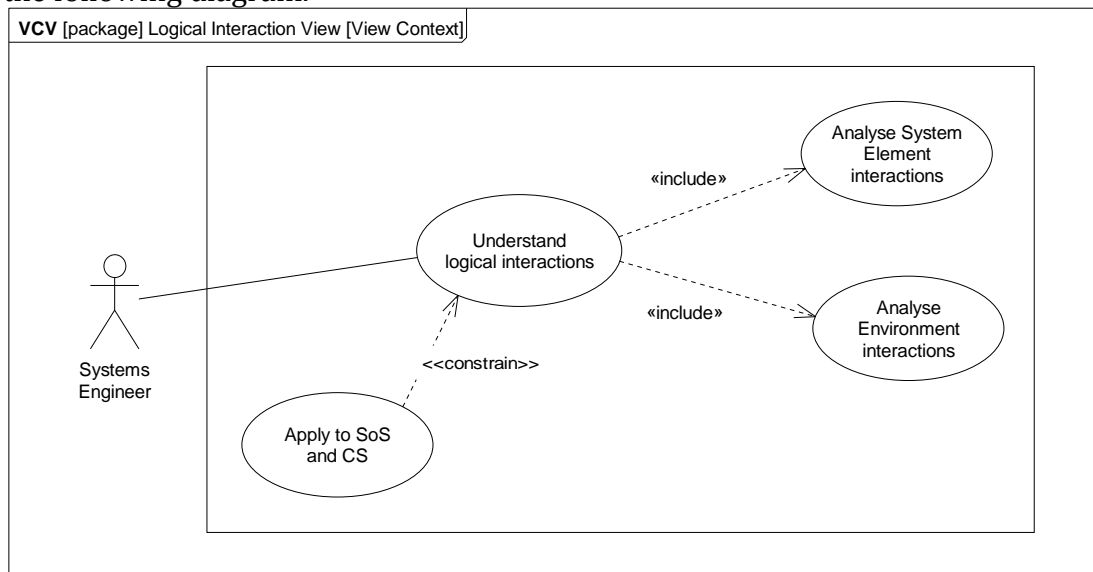


Figure 38 Viewpoint Context View for the Logical Interaction Viewpoint

The diagram in Figure 38 shows the basic needs for the Logical Interaction Viewpoint that will form the basis for all the Logical Interaction Views. There is a single main *use case* which is to 'Understand logical interactions' that includes:

- 'Analyse System Element Interaction', where the interactions between the System Elements are modelled and understood. In the case of a System of Systems, these System Elements will be Constituent Systems, whereas in the

case of a Constituent System they will be the System Elements of that Constituent System.

- ‘Analyse Environment interactions’, where the interactions between the System and its Environment and Environment Elements are modelled and understood.

There is a single constraint which is to ‘Apply to SoS and CS’ that reflects the need to apply the Logical Interaction Viewpoint at these two levels.

Based on this Context, the following subset of the COMPASS Ontology has been identified as relevant to the Architectural Design Framework.

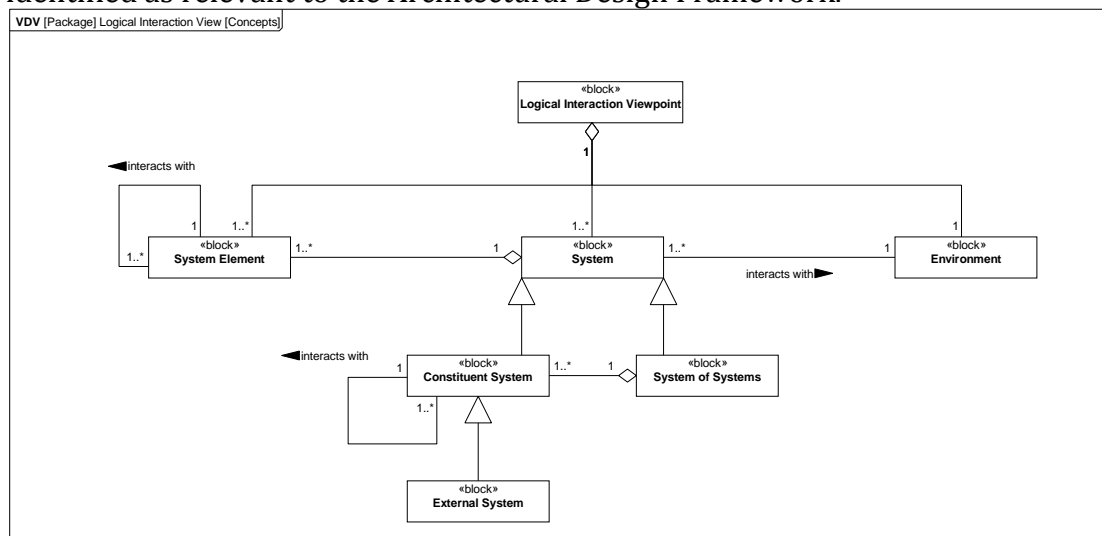


Figure 39 Viewpoint Definition View for the Logical Interaction Viewpoint

The diagram in Figure 39 shows the Ontology Elements that are relevant for the Logical Interaction Viewpoint that will form the basis for the Logical Interaction Views.

Component Structure View

The Viewpoint Context View for the Component Structure Viewpoint is shown in the following diagram.

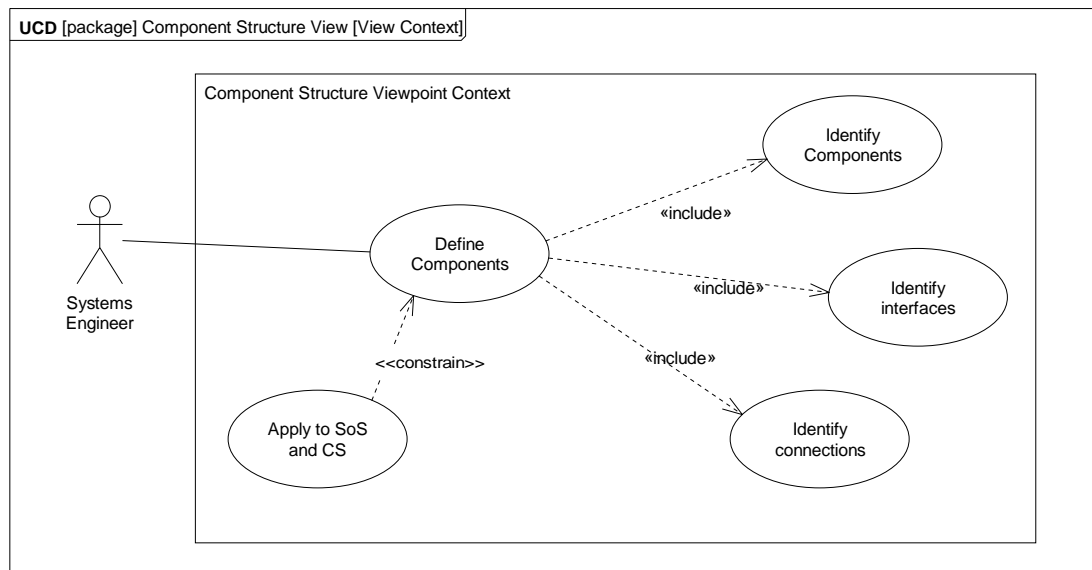


Figure 40 Viewpoint Context View for the Component Structure Viewpoint

The diagram in Figure 40 shows the basic needs for the Component Structure Viewpoint that will form the basis for all the Component Structure Views.

There is a single main *use case* which is to ‘Define Components’ and which includes the following sub *use cases*:

- ‘Identify Components’, where System Elements are partitioned into Components.
- ‘Identify Interfaces’, where the relevant interfaces for each Component are identified and defined.
- ‘Identify connections’, where the connections between interfaces for the Components are identified.

There is a single constraint which is to ‘Apply to SoS and CS’ that reflects the need to apply the Component Structure View at these two levels.

Based on this Context, the following subset of the COMPASS Ontology has been identified as relevant to the Architectural Design Framework.

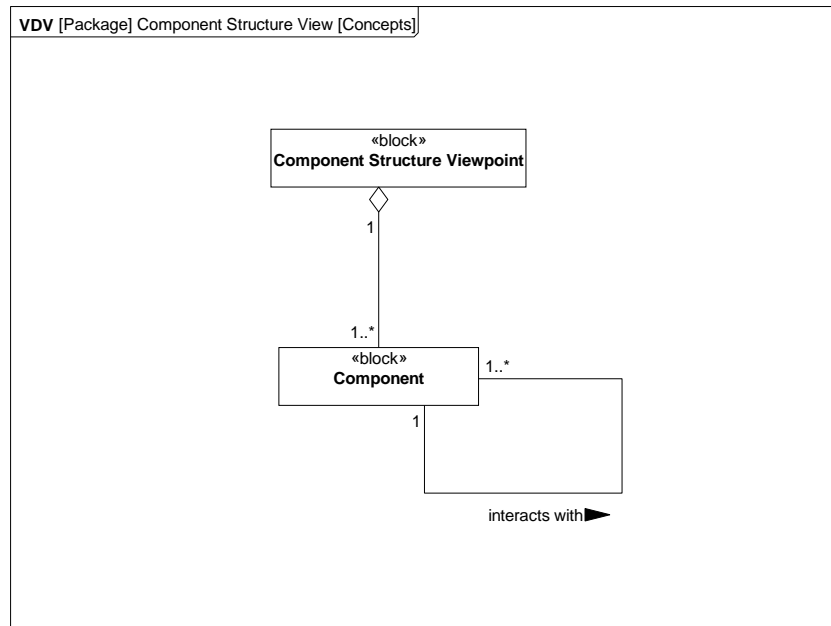


Figure 41 Viewpoint Definition View for the Component Structure Viewpoint

The diagram in Figure 41 shows the Ontology Elements that are relevant for the Component Structure Viewpoint that will form the basis for the Component Structure Views.

Component Interaction View

The Viewpoint Context View for the Component Interaction Viewpoint is shown in the following diagram.

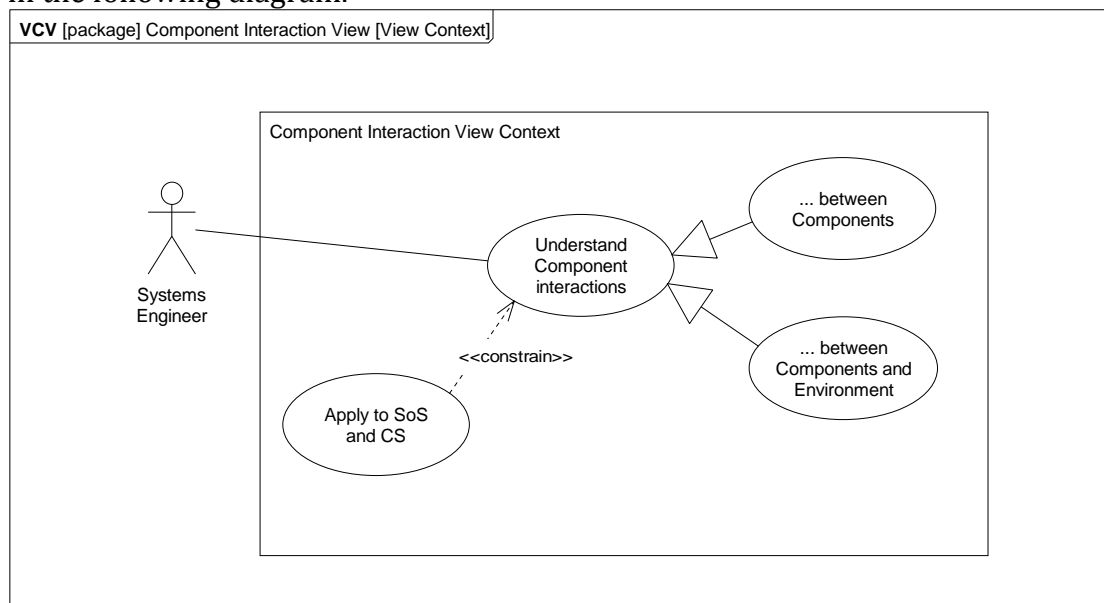


Figure 42 Viewpoint Context View for the Component Interaction Viewpoint

The diagram in Figure 42 shows the basic needs for the Component Interaction Viewpoint that will form the basis for all the Component Interaction Views.

There is a single main *use case* shown here which is to ‘Understand Component interaction’ both between Component (‘... between Components’) and between

the Components and the Environment ('... between Components and Environment').

There is a single constraint which is to 'Apply to SoS and CS' that reflects the need to apply the Component Structure View at these two levels.

Based on this Context, the following subset of the COMPASS Ontology has been identified as relevant to the Architectural Design Framework.

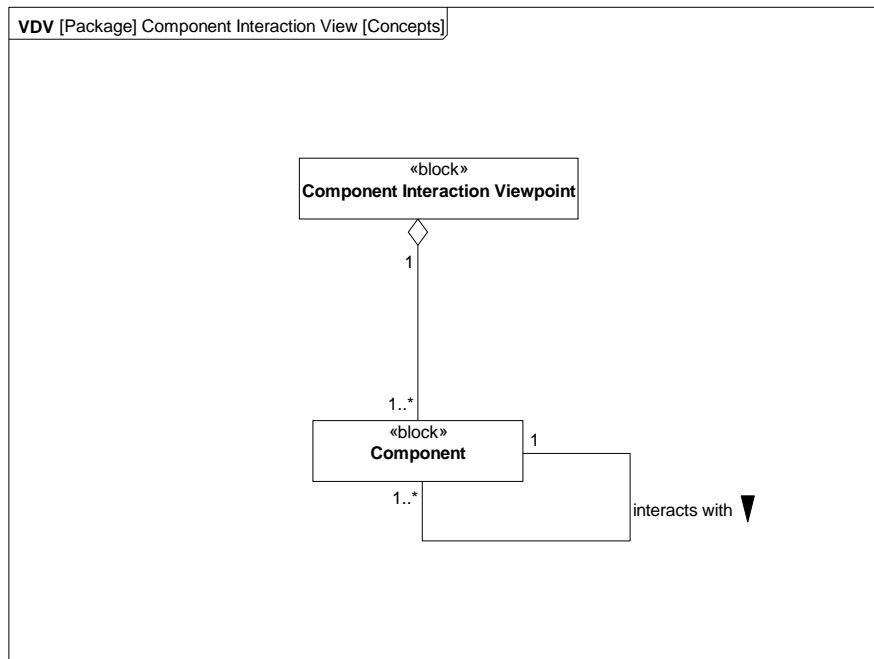


Figure 43 Viewpoint Definition View for the Component Interaction Viewpoint

The diagram in Figure 43 shows the Ontology Elements that are relevant for the Component Interaction Viewpoint that will form the basis for the Component Interaction Views.

Deployment View

The Viewpoint Context View for the Deployment Viewpoint is shown in the following diagram.

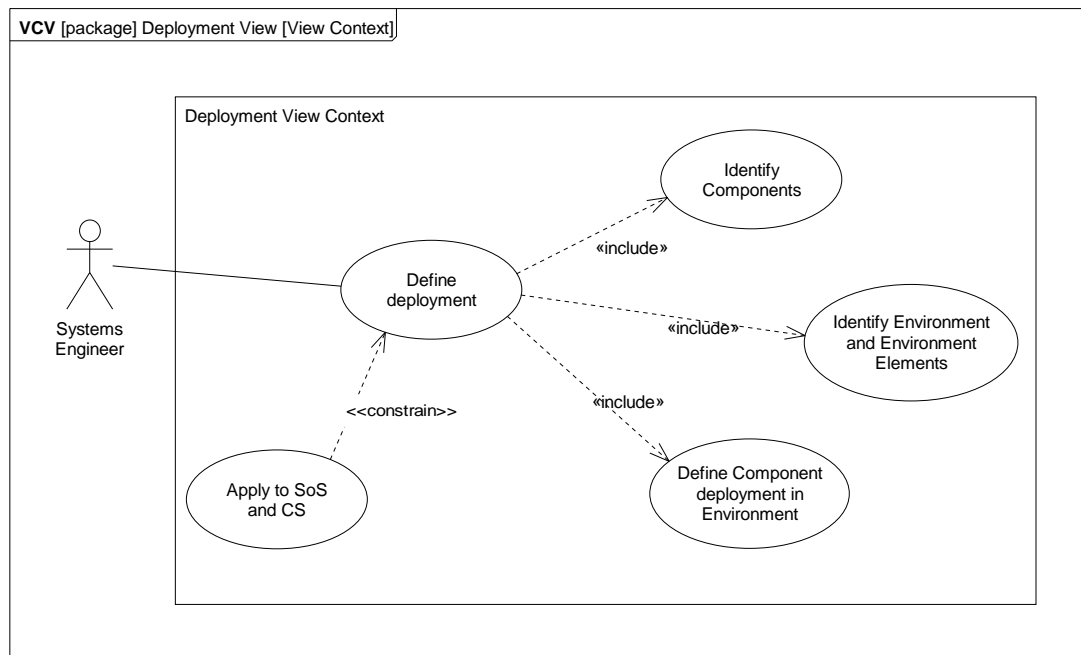


Figure 44 Viewpoint Context View for the Deployment Viewpoint

The diagram in Figure 44 shows the basic needs for the Deployment Viewpoint that will form the basis for all the Deployment Views.

There is a single main *use case* which is to ‘Define deployment’ and that includes the following sub *use cases*:

- ‘Identify Components’, where individual Component or set of Components are identified for deployment purposes.
- ‘Identify Environment and Environment Elements’, where the relevant parts of the Environment are identified for deployment purposes.
- ‘Defining Component deployment in Environment’, where the selected Component are related to their target deployment Environment Elements in the Environment.

There is a single constraint which is to ‘Apply to SoS and CS’ that reflects the need to apply the Deployment Viewpoint at these two levels.

Based on this Context, the following subset of the COMPASS Ontology has been identified as relevant to the Architectural Design Framework.

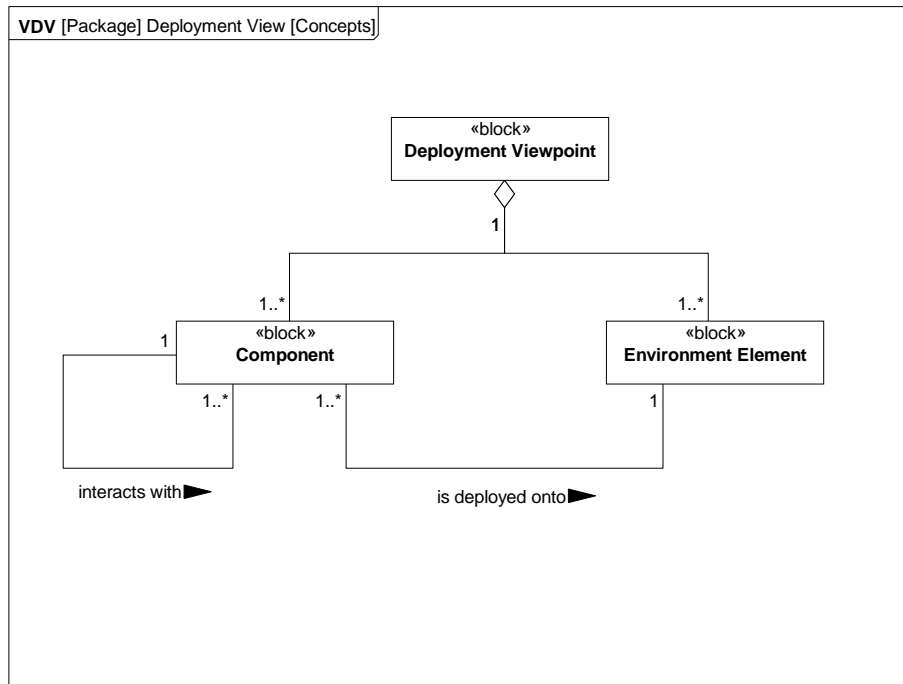


Figure 45 Viewpoint Definition View for the Deployment Viewpoint

The diagram in Figure 45 shows the Ontology Elements that are relevant for the Deployment Viewpoint that will form the basis for the Component Interaction Views.

Environment Identification View

The Viewpoint Context View for the Environment Identification Viewpoint is shown in the following diagram.

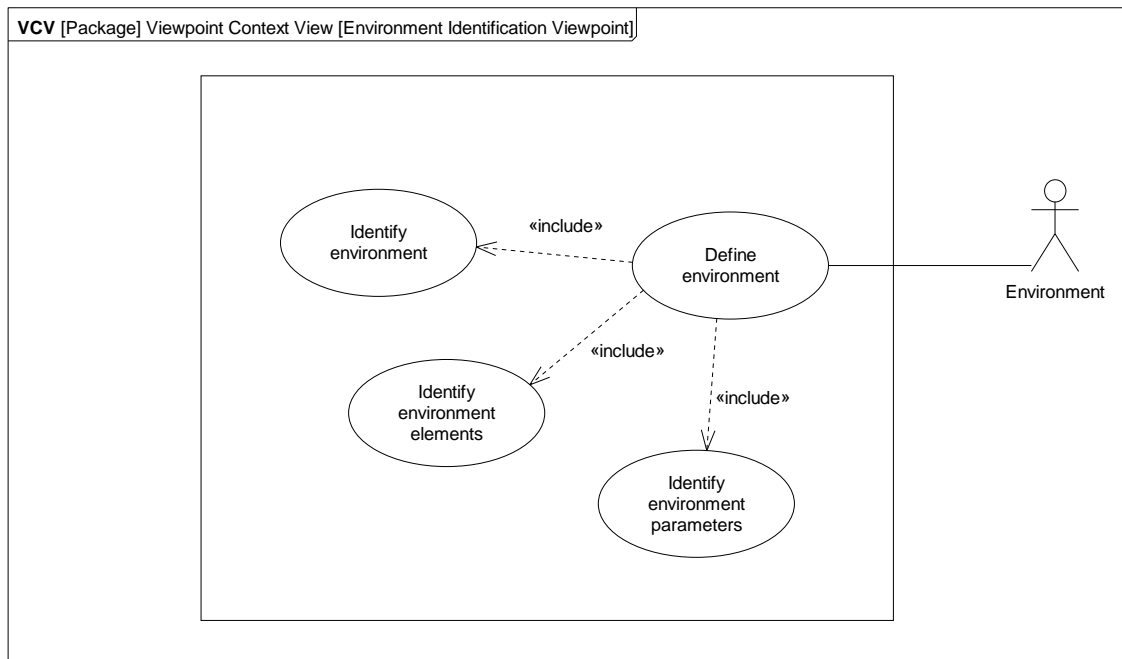


Figure 46 Viewpoint Context View for the Environment Identification Viewpoint

The diagram in Figure 46 shows the Viewpoint Context View for the Environment Identification Viewpoint that has a single main *use case*, which is to ‘Define environment’. This has three sub *use cases* which are:

- ‘Identify environment’, where the Environment that the System of Systems is to be deployed into is identified and named.
- ‘Identify environment elements’, where the Environment Elements that make up the Environment are identified and named.
- ‘Identify environment parameters’, where any relevant parameters that describe the individual Environment Elements are identified and defined.

Based on this Context, the following subset of the COMPASS Ontology has been identified as relevant to this Viewpoint.

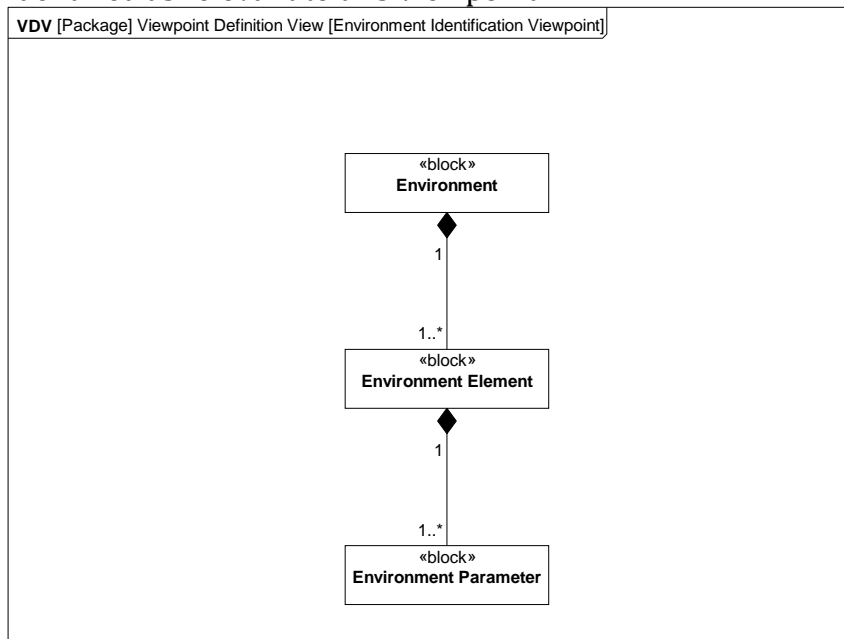


Figure 47 Viewpoint Definition View for the Environment Identification Viewpoint

The diagram in Figure 47 shows the Ontology Elements that are relevant for the Environment Identification Viewpoint. A new Ontology Element is introduced here which is:

- ‘Environment Parameter’, that describes a specific property of the Environment Element

This will form the basis for the Environment Identification Views.

Environment Configuration View

The Viewpoint Context View for the Environment Configuration Viewpoint is shown in the following diagram.

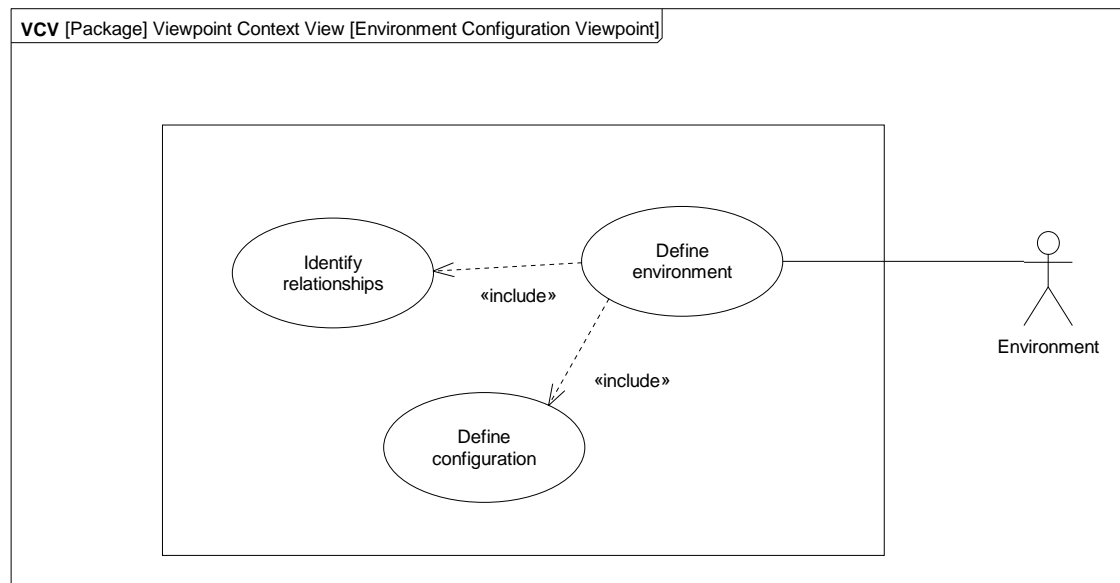


Figure 48 Viewpoint Context View for the Environment Configuration Viewpoint

The diagram in Figure 48 shows the Viewpoint Context View for the Environment Configuration Viewpoint that has a single main *use case*, which is to ‘Define environment’. This has two sub *use cases* which are:

- ‘Identify relationships’, where the relationships between the Environment Elements are identified.
- ‘Define configuration’, where the configuration of the Environment Elements that make up the Environment are defined, according to the identified relationship between them.

Based on this Context, the following subset of the COMPASS Ontology has been identified as relevant to this Viewpoint.

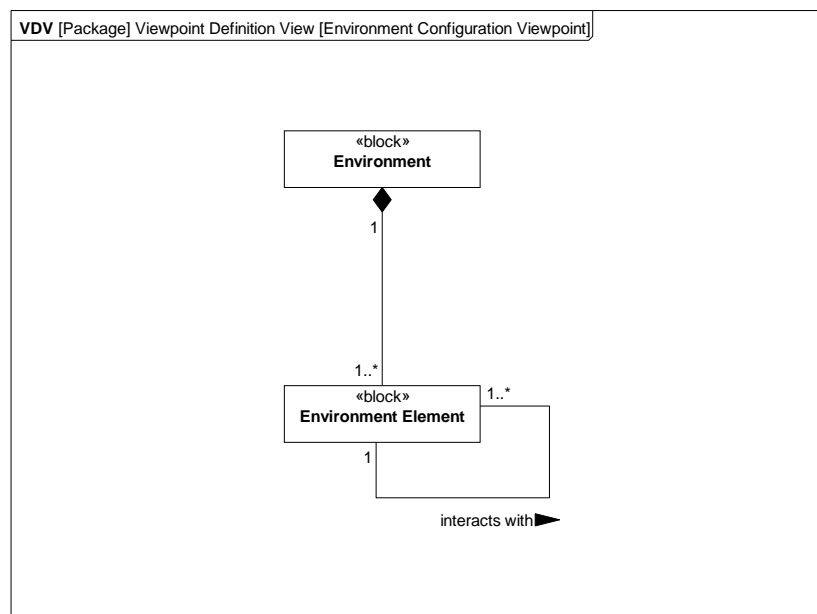


Figure 49 Viewpoint Definition View for the Environment Configuration Viewpoint

The diagram in Figure 49 shows the Ontology Elements that are relevant for the Environment Identification Viewpoint that will form the basis for the Environment Configuration Views.

B.6 Summary of Defined Viewpoints

The defined Viewpoints are summarised in the following table,

Viewpoint	Usage
Logical Structure Viewpoint	To define the logical view of a System (CS or SoS) through the identification of the System Elements, identification of External Systems that are not part of the System, but which are interacted with, identification of the target deployment Environment, and definition of any relationships between the System, System Elements, Environment and External Systems are defined.
Logical Interaction Viewpoint	To understand logical interactions, where the interactions between the System Elements are modelled and understood, and where the interactions between the System and its Environment and Environment Elements are modelled and understood.
Component Structure Viewpoint	To define Components through partitioning of System Elements into Components, with the relevant interfaces for each Component identified and defined, and with the connections between interfaces for the Components identified.
Component Interaction Viewpoint	To understand Component interaction, both between Components and between the Components and the Environment.
Deployment Viewpoint	To define deployment, where the individual Components or set of Components are identified for deployment purposes, where the relevant parts of the Environment are identified for deployment purposes, and where the selected Components are related to their target deployment Environment Elements in the Environment.
Environment Identification Viewpoint	To define the Environment, where the Environment that the System of Systems is to be deployed into is identified and named, where the Environment Elements that make up the Environment are identified and named, and where any relevant parameters that describe the individu-

	al Environment Elements are identified and defined.
Environment Configuration Viewpoint	To define the configuration of the Environment, where the relationships between the Environment Elements are identified, and where the configuration of the Environment Elements that make up the Environment are defined, according to the identified relationship between them.