WIT 2016 ITA Module

Architecture Description
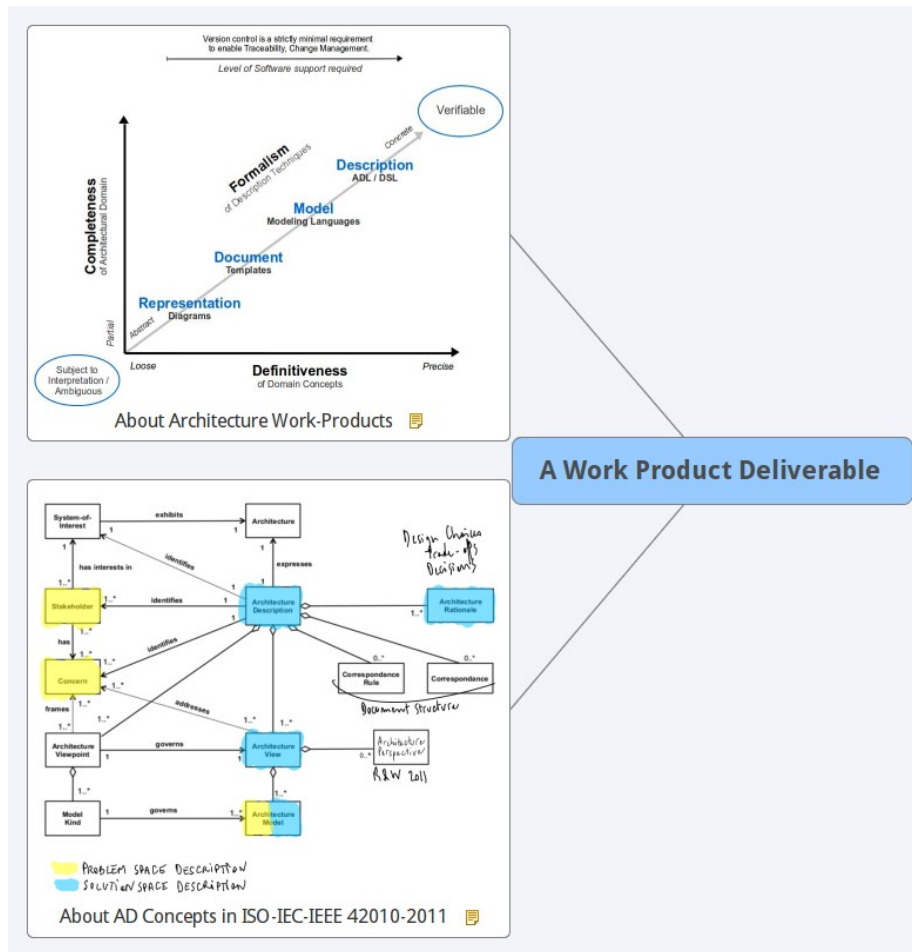# Lecture Group #2 - Part 2

Lecture Group #2 - Part 2
Architecture Description

WIT 2016 ITA Module
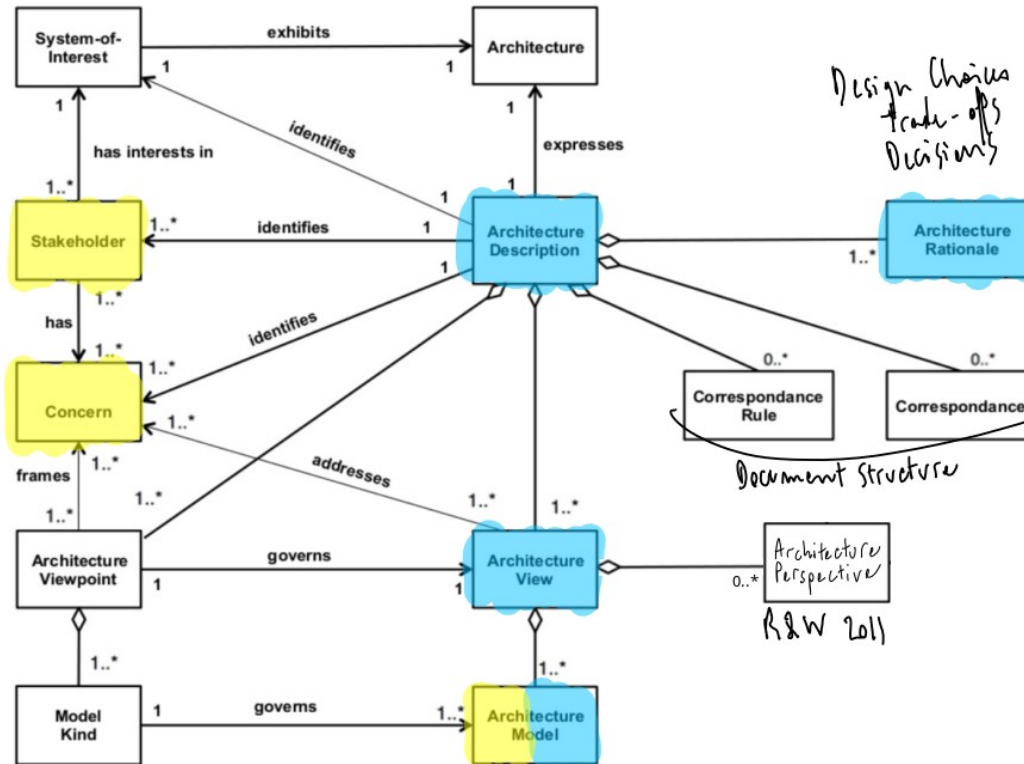Lecture Group #2 - Part 2
Architecture Description

Design Techniques - Architecture Description (AD)

A Work Product Deliverable

About Architecture Work-Products

A Work Product Deliverable

About AD Concepts in ISO-IEC-IEEE 42010-2011

About AD Concepts in ISO-IEC-IEEE 42010-2011

About AD Concepts in ISO-IEC-IEEE 42010-2011

- Once Stakeholder Concerns have been identified, they must be framed by one or more VIEWPOINTS.

- Once Viewpoints have been selected, these are used to establish Views with their Models.

- Model artifacts express Views of the architecture solution using MODEL KINDS (also referred as model-types or view-types) as templates for diagraming.

- Once Views are established, the quality attributes of each View are assessed using PERSPECTIVES.

Term: Stakeholder Concern

A concern pertains to any influence on a system from environment, (e.g. operational, organizational, economic, regulatory, social, technological, etc.)

Architects looking for an architecture viewpoint suitable for their purposes often listen to concerns (i.e. interests) from stakeholders to guide them in their search.

Once identified, these concerns act as "forces" or "constraints" on the solution scope.

It is important to document the concerns and stakeholders for which a viewpoint is intended.

Term: Stakeholder Viewpoint

A Viewpoint strictly formalises the modeling elements, modeling concepts, conventions and rules to model a Primitive Model or Composite Model (i.e. View) of an Architecture.

A Viewpoint proposes the vocabularies and rules for constructing models and how those models are interpreted and used by stakeholders.

When used in an architecture description, the viewpoint becomes a "contract" between the architect and stakeholders that their concerns will be addressed in the view resulting from this viewpoint.

A Viewpoint is VERIFIABLE.

Term: Perspectives / Quality Attributes

Quality Attributes (also referred as required Quality Properties, or Perspectives) are "areas of interest" in a system pervasive to all views of this system.

Some specific concerns are driven by the need for the system to exhibit a certain quality properties, independently from stakeholder concerns, rather than to provide a particular function.

Trying to address these aspects of an architecture by using Viewpoints does not work well, hence the concept of Architecture Perspective, horizontal to all Views of a solution.

Example of quality properties are; security, performance, availability, or usability.

About Architecture Work-Products
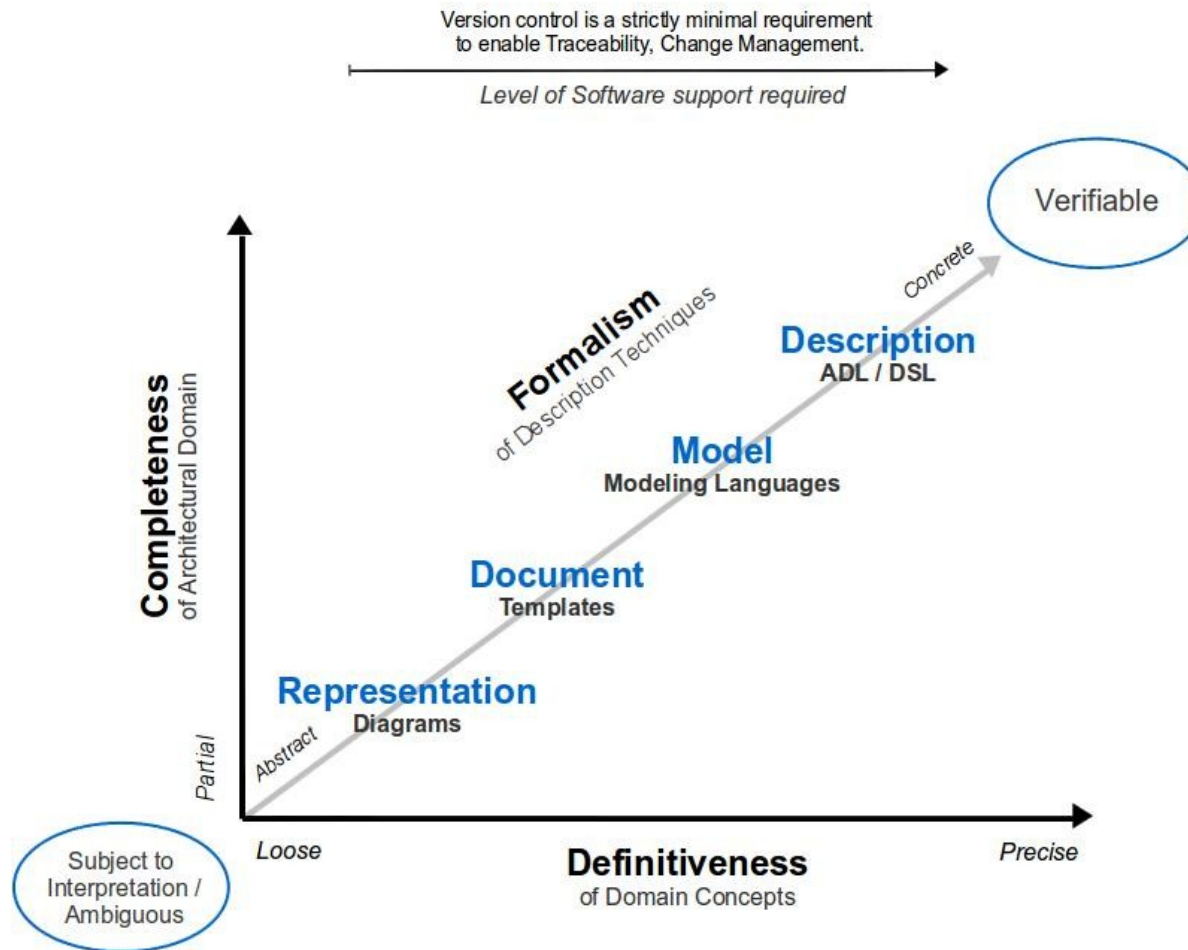
About Architecture Work-Products

The degree of specification of any Work-Product (WP) can be expressed in terms of its: (1.) Definitiveness, (2.) Completeness and (3.) Formalism.

A work-product with a low specification power is ambiguous and subject to interpretation. A work-product with a high specification power is "verifiable".

The higher the specification power, the better candidate the work-product is for knowledge transfer.

An architect chooses the appropriate degree of specification of a work-product to CONVEY the key elements of the solution design to stakeholders.

About Work-Product Artifacts

A Work-Product "ARTIFACT" is a focused aspect of a more holistic design.

A MODEL PRIMITIVE is a work-product artifact, is smallest unit of delivery of an architecture.

It represent one building block of a design and can be composed, reviewed, or used in a wide variety of ways.

A COMPOSITE MODEL or ARCHITECTURE VIEW is also a work-product artifact.

It captures enough information about the "part" of the design so that it can be thoroughly analysed and described (e.g. modeled).

Work-Product ARTIFACTS in time become precious knowledge assets.

As such, these must be version-controlled, maintained and periodically refreshed.

TOGAF 9.1. Deliverable

A Work-Product "deliverable" is a COMPOSITE document/design presenting a holistic solution design.

It explain what makes a set of parts work together, as a coherent and successful whole.

It relies on the building-blocks provided by work-product Artifacts - but Unlike Artifacts, the substantive value provided to stakeholders has less to do with degree of formalism or format of delivery.

Instead, work-product deliverables accentuate the "narrative" and record architectural decisions, verbally explaining their outcomes & results.

Coherence and Consistency in the way that Deliverables document the required trade-offs and quality attributes of an architecture is what guarantees valuable outcomes.

A DELIVERABLE encapsulates and augments diagram PRIMITIVES into a written document (e.g. WIKI, Word, PowerPoint or other).
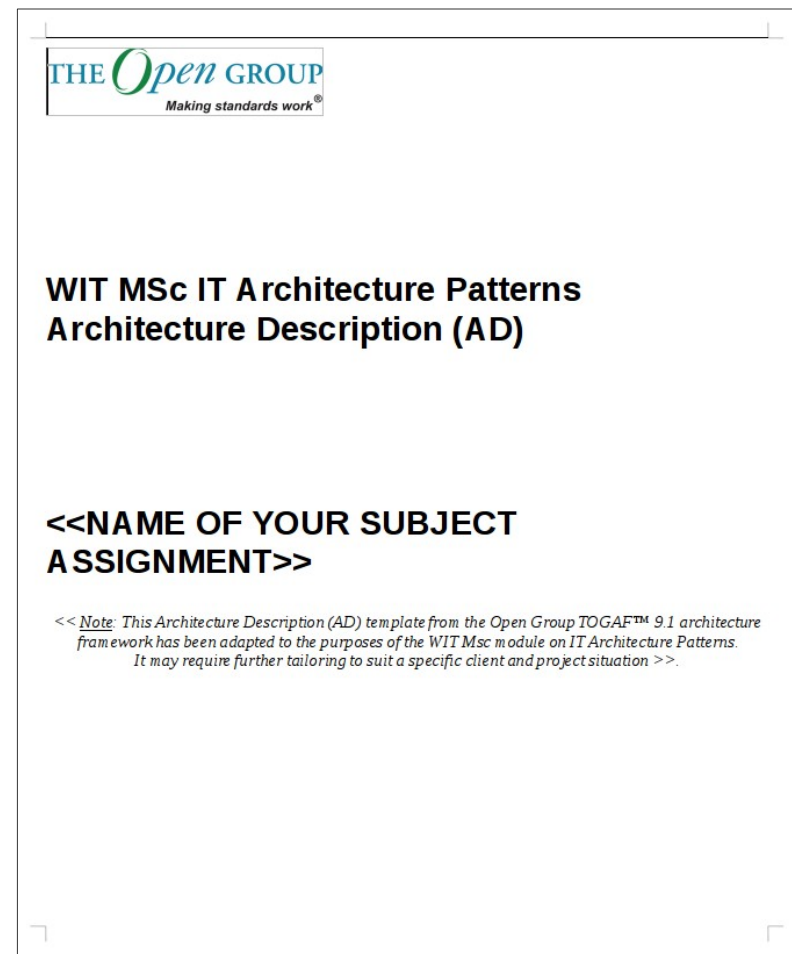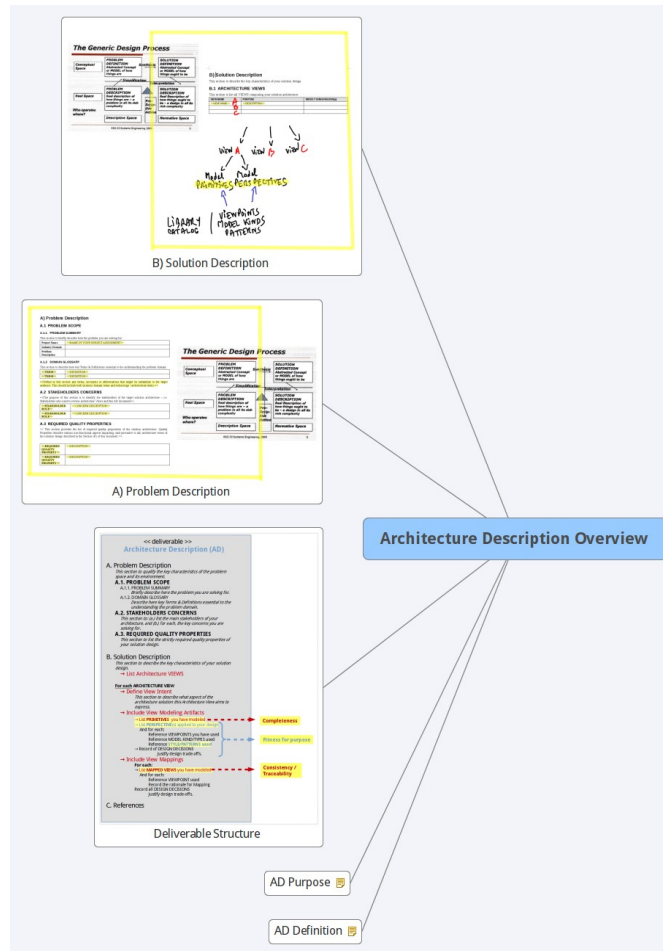
Example: TOGAF 9.1. AD Template

An Architecture Description (AD) is a document authored by an Architect.

A partial description of an architecture solution is referred as a work-product ARTIFACT.

A COMPLETE description of the solution architecture - i.e. the AD - is a work-product DELIVERABLE.

THE *Open* GROUP
Making standards work®

WIT MSc IT Architecture Patterns Architecture Description (AD)

<<NAME OF YOUR SUBJECT ASSIGNMENT>>

<< *Note: This Architecture Description (AD) template from the Open Group TOGAF™ 9.1 architecture framework has been adapted to the purposes of the WIT Msc module on IT Architecture Patterns. It may require further tailoring to suit a specific client and project situation >>.*

Architecture Description Overview

B) Solution Description

A) Problem Description

Architecture Description Overview

Deliverable Structure

AD Purpose

AD Definition

AD Definition

An Architecture Description (AD) expresses the Architecture of a System of interest (ISO-IEC-IEEE 42010-2011).

It is a composition of model artifacts used to define, describe and detect gaps in the architecture - to demonstrate coverage COMPLETENESS of stakeholders requirements.

It links together views of the solution architecture, and records the reasons for decisions made - to demonstrate design CONSISTENCY and TRACEABILITY between design primitives.

It uses Perspectives to assess that all required quality properties are met all through the system, and across all views - to demonstrate that the solution is FIT FOR PURPOSE.
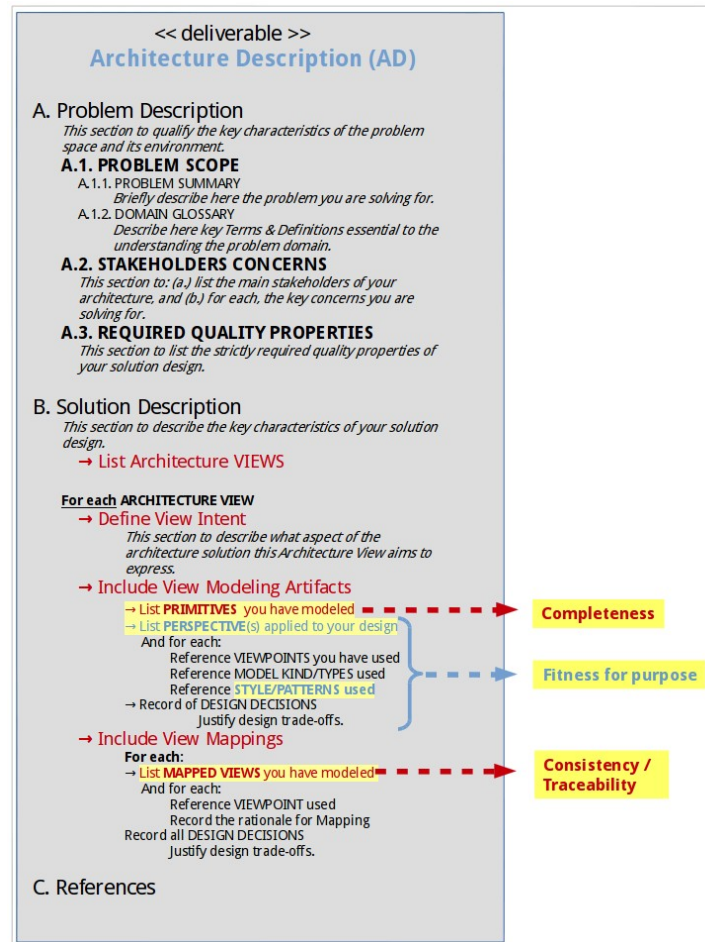
AD Purpose

The purpose of an Architecture Description (AD) is to:

- document all stakeholders of the system and document their concerns

- select the Viewpoints that best frame, or cover, those concerns

- document the Views of the architecture, such that each satisfies one of those viewpoints

- link together those Views with correspondences and recording any known inconsistencies between Views

- document the Perspectives of the architecture, such that each satisfies its required quality properties

- provide rationale for key decisions made in the architecture design.

Deliverable Structure

<< deliverable >>
Architecture Description (AD)

A. Problem Description
This section to qualify the key characteristics of the problem space and its environment.
   A.1. PROBLEM SCOPE
      A.1.1. PROBLEM SUMMARY
         Briefly describe here the problem you are solving for.
      A.1.2. DOMAIN GLOSSARY
         Describe here key Terms & Definitions essential to the understanding the problem domain.
   A.2. STAKEHOLDERS CONCERNS
      This section to: (a.) list the main stakeholders of your architecture, and (b.) for each, the key concerns you are solving for.
   A.3. REQUIRED QUALITY PROPERTIES
      This section to list the strictly required quality properties of your solution design.

B. Solution Description
This section to describe the key characteristics of your solution design.
   → List Architecture VIEWS

For each ARCHITECTURE VIEW
   → Define View Intent
         This section to describe what aspect of the architecture solution this Architecture View aims to express.
   → Include View Modeling Artifacts
      → List **PRIMITIVES** you have modeled ----→ **Completeness**
      → List **PERSPECTIVE**(s) applied to your design
      And for each:
         Reference VIEWPOINTS you have used
         Reference MODEL KIND/TYPES used ----→ **Fitness for purpose**
         Reference STYLE/PATTERNS used
      → Record of DESIGN DECISIONS
            Justify design trade-offs.
   → Include View Mappings
      For each:
      → List **MAPPED VIEWS** you have modeled ----→ **Consistency / Traceability**
      And for each:
         Reference VIEWPOINT used
         Record the rationale for Mapping
      Record all DESIGN DECISIONS
            Justify design trade-offs.

C. References

A) Problem Description

A) Problem Description

A.1 PROBLEM SCOPE

A.1.1 PROBLEM SUMMARY

This section to briefly describe here the problem you are solving for.

Project Name	<<NAME OF YOUR SUBJECT ASSIGNMENT>>
Industry Domain	
Problem Description	

A.1.2 DOMAIN GLOSSARY

This section to describe here key Terms & Definitions essential to the understanding the problem domain.

<<TERM>>	<<DEFINITION>>
<<TERM>>	<<DEFINITION>>

<<Define in this section any terms, acronyms or abbreviations that might be unfamiliar to the target audience. This should include both business domain terms and technology / architectural terms.>>

A.2 STAKEHOLDERS CONCERNS

<<The purpose of this section is to identify the stakeholders of the target solution architecture – i.e. Stakeholders who need to review architecture Views and this AD document>>.

<<STAKEHOLDER ROLE>>	<<CONCERN DESCRIPTION>>
<<STAKEHOLDER ROLE>>	<<CONCERN DESCRIPTION>>

A.3 REQUIRED QUALITY PROPERTIES

<< This section provides the list of required quality propoerties of the solution architecture. Quality Properties describe various non-functional aspects impacting (and pervasive to all) architecture views of the solution design described in the Section (B) of this document >>.

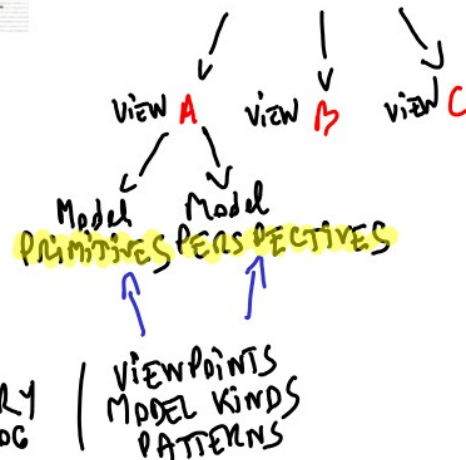| <<REQUIRED QUALITY PROPERTY>>	<<DESCRIPTION>>
<<REQUIRED QUALITY PROPERTY>>	<<DESCRIPTION>>

The Generic Design Process

Conceptual Space	PROBLEM DEFINITION Abstracted Concept or MODEL of how things are	Synthesis	SOLUTION DEFINITION Abstracted Concept or MODEL of how things ought to be

Simplification — Interpretation

Real Space	PROBLEM DESCRIPTION Real description of how things are – a problem in all its rich complexity	Pre-Scrip-tive Action	SOLUTION DESCRIPTION Real Description of how things ought to be – a design in all its rich complexity

Who operates where?

Descriptive Space		Normative Space

ESD.33 Systems Engineering 2009   5

B) Solution Description

B) Solution Description

View Mappings (Composite Models)

Design Decision / Trade-offs (Textual description)

Perspectives (Model + Textual Annotation)

View Primitives (Model Primitives)

B) Solution Description

View Primitives (Model Primitives)

B.2.2 Include VIEW MODELLING ARTIFACTS

B.2.2.1 For each MODEL VIEW PRIMITIVE
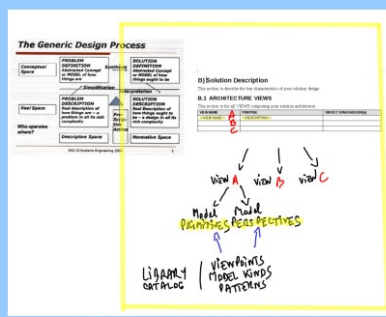
Insert here [.png] or [.jpg] snapshots of Archimate Models,

- or -

Embed here [.archimate] file objects.

Model #.archimate

VIEWPOINT USED	JUSTIFICATION / INTENT
<<VIEWPOINT NAME>>	<<DESCRIPTION>>
MODEL KIND USED	**JUSTIFICATION / INTENT**
<<MODEL-KIND / VIEW-TYPE NAME>>	<<DESCRIPTION>>
STYLE/PATTERNS USED	**JUSTIFICATION / INTENT**
<<ARCHITECURE STYLE NAME>>	<<DESCRIPTION>>
<<ARCHITECTURE PATTER NAME>>	<<DESCRIPTION>>

Perspectives (Model + Textual Annotation)

B.2.2.2 For each MODEL PERSPECTIVE

Insert here [.png] or [.jpg] snapshots of Archimate Models + *descriptive Textual Annotations*

- or -

Embed here [.archimate] file objects including *descriptive Textual Annotations*.

Model #.archimate

STYLE/PATTERNS USED	JUSTIFICATION / INTENT
<<ARCHITECURE STYLE NAME>>	<<DESCRIPTION>>
<<ARCHITECTURE PATTER NAME>>	<<DESCRIPTION>>

Design Decision / Trade-offs (Textual description)

B.2.2.3 RECORDING of DESIGN DECISIONS

.ID	DECISION ITEM	DECISION MADE
<<#>>	<<DESCRIPTION>>	<<DESCRIPTION>>

View Mappings (Composite Models)

B.2.3   Include VIEW MAPPINGS

B.2.3.1   For each VIEW MAPPING

Insert here [.png] or [.jpg] snapshots of Archimate Models,

- or -

Embed here [.archimate] file objects.

Model #.archimate

VIEWPOINT USED	JUSTIFICATION / INTENT
<<VIEWPOINT NAME>>	<<DESCRIPTION>>

B.2.3.2   RECORDING of DESIGN DECISIONS

.ID	DECISION ITEM	DECISION MADE
<<#>>	<<DESCRIPTION>>	<<DESCRIPTION>>
		<<DESCRIPTION>>