



UNIVERSITY OF
TECHNOLOGY SYDNEY

DOCUMENTING ARCHITECTURES

Software Architecture

Tom McBride

**UTS:
ENGINEERING AND
INFORMATION
TECHNOLOGY**

Contents

- Who uses architecture descriptions
- What needs to be documented
- Setting the context
- Viewpoints
- Views and how to document them
- Rationale and decision capture



UNIVERSITY OF
TECHNOLOGY SYDNEY

USES OF ARCHITECTURE DESCRIPTIONS

Architecture as a boundary object

- A boundary object is something that serves as an interface between different communities of interest.
- There is enough agreement about the features of a boundary object to enable meaningful dialogue.
- Each community of interest might interpret different parts of a boundary object differently.
- A system or software architecture is a set of models of the system that enables all of the stakeholders to understand and discuss what the system will be and will do.



Uses of architecture descriptions - 1

- Architecture descriptions have many uses by a variety of stakeholders, throughout the system life cycle and beyond. Among the uses for architecture descriptions are:
 - as basis for subsequent system design and development activities;
 - as development and maintenance documentation;
 - to document essential aspects of a system, such as:
 - architectural assumptions and its intended use and environment;
 - principles and constraints to guide future change;
 - points of flexibility or limitations of the system with respect to future changes;
 - architectural decisions and their rationales and implications;

Uses of architecture descriptions - 2

- as input to automated tools for simulation, system generation and analysis;
- to specify a group of systems sharing a common features (such as architectural styles, reference architectures and product line architectures);
- to communicate among parties involved in the development, production, deployment, operation and maintenance of a system;
- to communicate among clients, acquirers, suppliers and developers as a part of contract negotiations;
- to communicate the characteristics, features and design of a system to potential clients, acquirers, owners, operators and integrators;
- to aid in planning for transition from a legacy architecture to a new architecture;

Uses of architecture description - 3

- to guide operational and infrastructure support and configuration management;
- to support system planning, scheduling and budgeting activities;
- to support preparation of acquisition documents (e.g., requests for proposal and statements of work);
- to establish criteria for certifying implementations for compliance with an architecture;
- to ensure compliance with external and company-/project internal policies (e.g., legislation, overarching architectural principles)
- to support review, analysis, and evaluation of the system across the life cycle;
- as basis to analyze and evaluate alternative architectures;
- to share lessons learned and reuse architectural knowledge;
- to train and educate stakeholders and other parties about best practices in architecting and system evolution.

Who reads it, and why?

Stakeholder	Use
Architect and requirements engineers who represent customer(s)	To negotiate and make tradeoffs among competing requirements
Architect and designers of constituent parts	To resolve resource contention and establish performance and other kinds of runtime resource consumption budgets
Implementers	To provide inviolable constraints (plus exploitable freedoms) on downstream development activities
Testers and integrators	To specify the correct black-box behaviour of the pieces that must fit together
Maintainers	To reveal areas a prospective change will affect
Designers of other systems with which this one must interoperate	To define the set of operations provided and required, and the protocols for their operation

More of who reads it

Stakeholder	Use
Quality attribute specialists	To provide the model that drives analytical tools such as rate monotonic real-time schedulability analysis, simulations and simulation generators, theorem provers, verifiers, etc. These tools require information about resource consumption, scheduling policies, dependencies, and so forth. Architecture documentation must contain the information necessary to evaluate a variety of quality attributes such as security, performance, usability, availability, and modifiability. Analyses for each attributes have their own information needs.
Managers	To create development teams corresponding to work assignments identified, to plan and allocate project resources, and to track progress by the various teams
Product line managers	To determine whether a potential new member of a product family is in or out of scope, and if out by how much
Quality assurance team	To provide a basis for conformance checking, for assurance that implementations have been faithful to the architectural prescriptions

Summary

- An architecture serves as a boundary object between the stakeholders
- An architecture is usually the first thing that enables all of the stakeholders to examine the system and how it will affect them
- An architecture serves many purposes

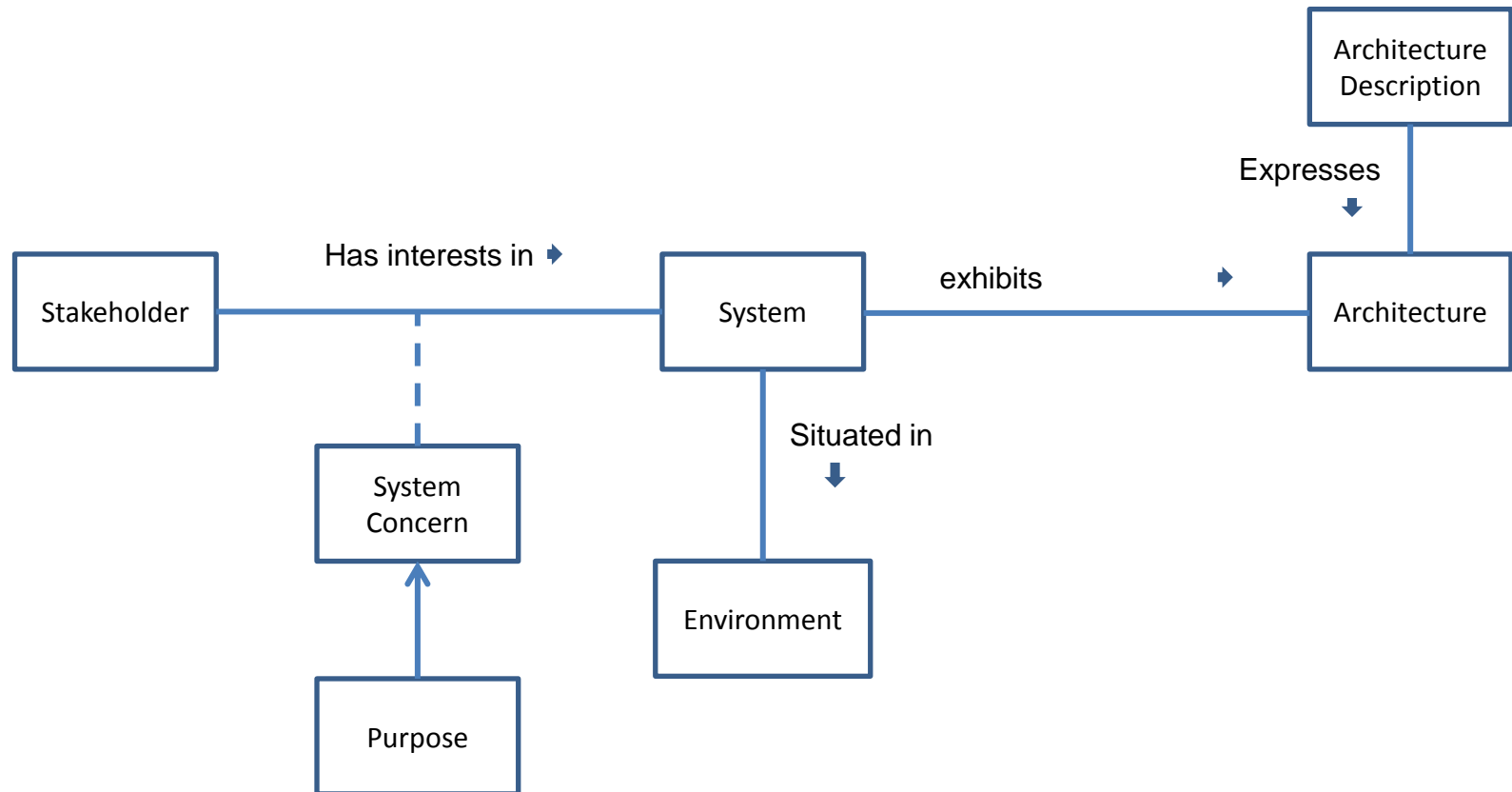


SETTING THE CONTEXT

References

- ISO/IEC 42010:2007 – Systems and software engineering – Architecture description
- IEEE Std 1471-2000 ISO/IEC Standard for Systems and Software Engineering - Recommended Practice for Architectural Description of Software-Intensive Systems
- Bass, L., Clements, P. and Kazman, R. (2003), *Software architecture in practice*, Addison-Wesley, 2nd ed – Chapter 9

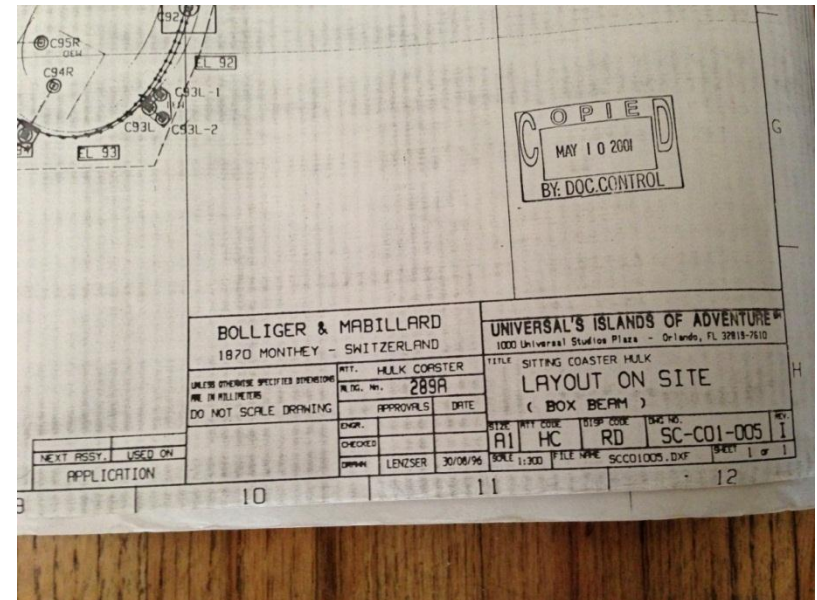
Context of an architecture description



From ISO 42010:2011

Architecture Description identification

- System of interest
- Version
- Other information as specified by the organisation concerned



Stakeholders and concerns

- Identify the stakeholders
- Identify their concerns about the system of interest
- Consider
 - Users
 - Operators
 - Acquirers
 - Owners
 - Suppliers
 - Developers
 - Maintainers



© Can Stock Photo - csp11371128

Narrative architecture description

- This is a short narrative description of what the system's function is, who its users are, and any important background or constraints. The intent is to provide readers with a consistent mental model of the system and its purpose. Sometimes the project at large will have a system overview, in which case this section of the architectural documentation simply points to that.



Summary

- The context for an architecture description identifies
 - The architecture description itself
 - The stakeholders and their concerns
 - A summary narrative about the system



UNIVERSITY OF
TECHNOLOGY SYDNEY

ARCHITECTURE VIEWPOINTS

Viewpoint definition

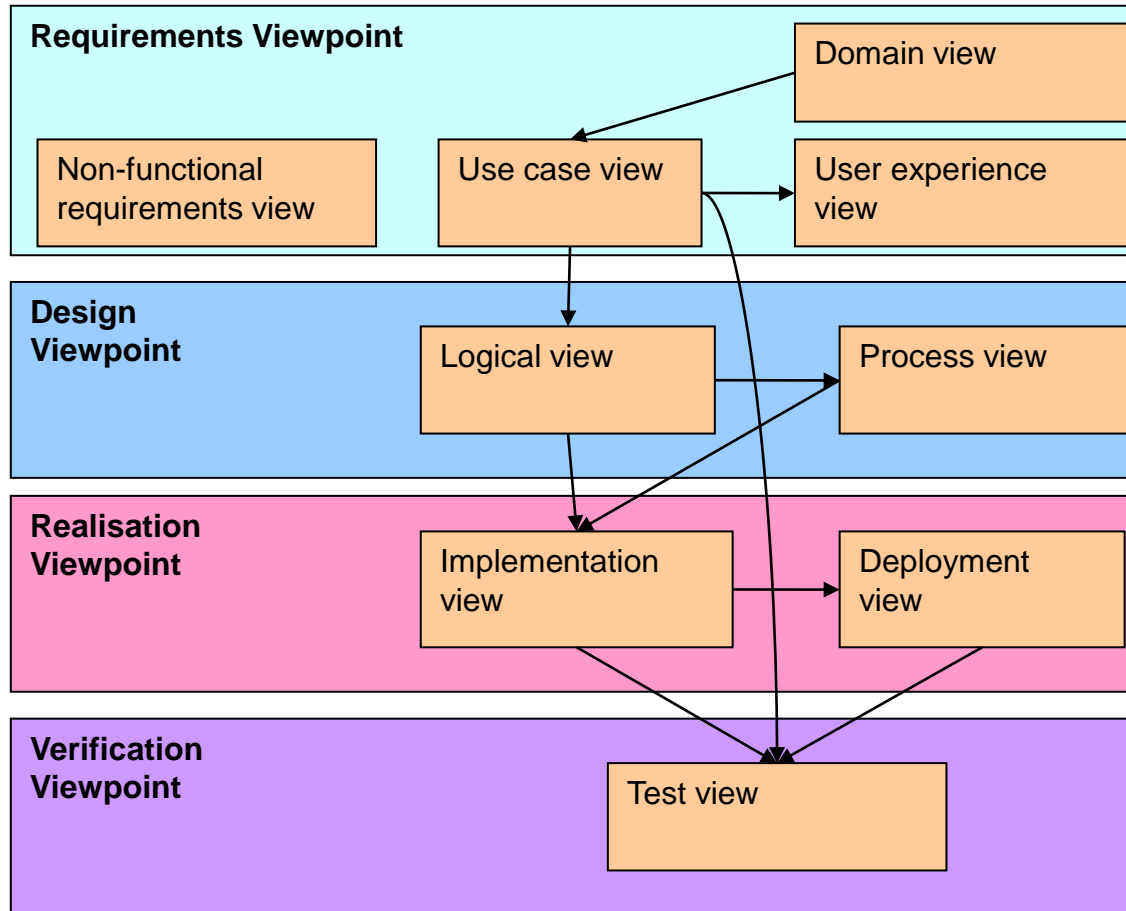
- Work product establishing the conventions for the construction, interpretation and use of architecture views to frame specific system concerns – ISO 42010:2011
- Many existing practices express architectures through collections of models. Typically, these models are further organized into cohesive groups, called *views*. The cohesion of a group of models is determined by the concerns addressed by that group of models. What has been missing in recent practice is a distinct term for the mechanism for formalizing these groupings and referring to the conventions by which the models are made. In this International Standard, *viewpoint* refers to the conventions for expressing an architecture with respect to a set of concerns.
- A viewpoint is a way of looking at systems; a view is the result of applying a viewpoint to a particular system-of-interest.

Architecture viewpoints

- An architectural viewpoint frames one or more system concerns held by stakeholders
- Examples (from Rational™ ADS)
 - Requirements viewpoint
 - Design viewpoint
 - Realisation viewpoint
 - Verification viewpoint

Rational™ Architecture Description Specification

– Viewpoints and views



Summary

- Viewpoint is a perspective, along with the conventions for the construction, interpretation and use of architecture views to frame specific system concerns
- Imagine the architecture from the perspective of system security



UNIVERSITY OF
TECHNOLOGY SYDNEY

ARCHITECTURE VIEWS

Architecture views

- An architecture is too large and too complex to show on one small piece of paper – that is a conceptual diagram
- Recall that “An architecture is the structure or structures of a system which comprise elements, the externally visible properties of those elements, and the relationships among them.”
- Recall also the different views
 - **Conceptual view** – to identify the elements and the relationships among them
 - **Execution view** – to identify the executable components and data flows
 - **Implementation view** – to identify how the system will be built and how it will be deployed
- Documenting an architecture is a matter of documenting the various views and matters that are common among them
- A documented architecture needs to communicate to its reader what the architecture is and why it is that way.

Choosing the relevant views

	Architecture views					
Stakeholder	Conceptual view	High level class diagrams	Collaboration view	Sequence diagrams	Component diagram	State machine diagram
Customer	S	O				
End User	S	O				
Sponsor	S	O				
Project manager	S	S	O	O	D	S
Developer	D	D	D	D	D	D
Testers and integrators	S	S	D	D	S	D
Maintainers	D	D	D	D	D	D
Current and future architect	D	D	D	D	D	D

Key: d = detailed information, s = some details, o = overview, x = anything

Choosing the relevant views – take 2

	Architecture views			
Stakeholder	Conceptual view	High level class diagrams	Collaboration view	
Customer	S	O		
End User	S	O		
Sponsor	S	O		
Project manager	S	S	O	
Developer	D	D	D	
Testers and integrators	S	S	D	
Maintainers	D	D	D	
Current and future architect	D	D	D	

Key: d = detailed information, s = some details, o = overview, x = anything

Documenting a view – ISO 42010

Architecture descriptions include the following contents, as specified in the remainder of this clause:

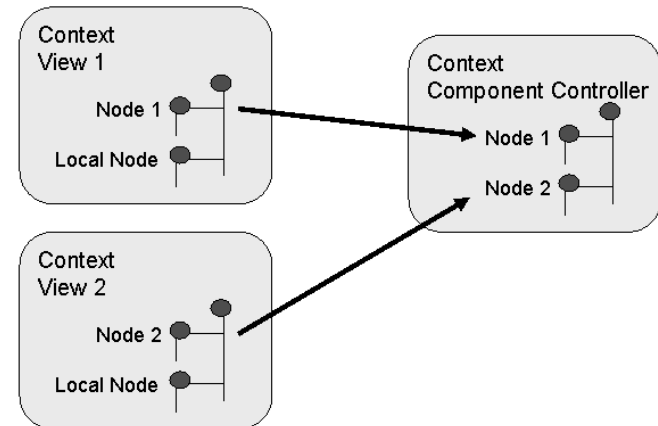
- a definition for each architecture viewpoint used in the architecture description;
- an architecture view and architecture models for each architecture viewpoint used;
- applicable AD correspondence rules, AD correspondences and a record of known inconsistencies among the architecture description's required contents;
- the rationales for architecture decisions made.

View catalogue

- A view catalogue is the reader's introduction to the views that the architect has chosen to include in the suite of documentation.
- There is one entry in the view catalogue for each view given in the documentation suite. Each entry should give the following:
 - The name of the view and what style it instantiates
 - A description of the view's element types, relation types, and properties
 - A description of what the view is for
 - Management information about the view document, such as the latest version, the location of the view document, and the owner of the view document
- The view catalogue is intended to describe the documentation suite, not the system being documented. Specifics of the system belong in the individual views, not in the view catalog. For instance, the actual elements contained in a view are listed in the view's element catalog.

Mapping between views

- Since all of the views of an architecture describe the same system, any two views will have much in common.
- Helping a documentation reader understand the relationships among views will give him a powerful insight into how the architecture works as a unified conceptual whole.
- Being clear about the relationship by providing mappings between views is the key to increased understanding and decreased confusion.
- Example
 - What is the mapping between elements in the conceptual view and elements in the implementation view

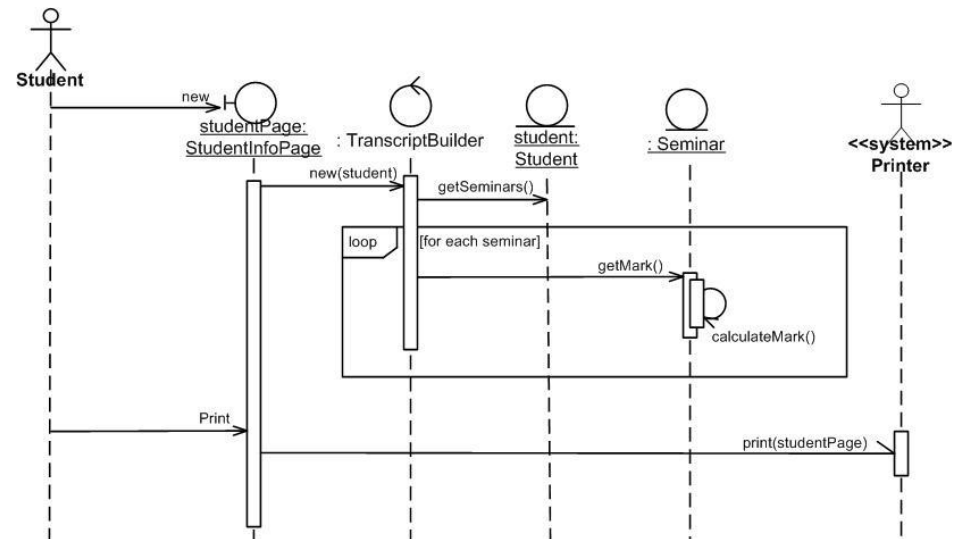


Element list

- The element list is simply an index of all of the elements that appear in any of the views, along with a pointer to where each one is defined. This will help stakeholders look up items of interest quickly.
- Note – ISO 42010 does not require an element list

Documenting behaviour

- Structural information is insufficient to allow reasoning about some properties
- Examples:
 - Reasoning about deadlock
 - System response to real-time events
- Use state charts or sequence diagrams



Documenting interfaces

- An interface is a boundary across which two independent entities interact or communicate with each other
- Documenting an interface consists of identifying the interface and documenting the semantic and syntactic information – the signature
- Signature example;
 - C or C++ header file.
 - Java interface

Documenting interface details

- Interface identity and version
- Resources provided.
 - Resource syntax. This is the resource's signature. The signature includes any information another program will need to write a syntactically correct program that uses the resource. The signature includes the resource name, names and logical data types of arguments (if any), and so forth.
 - Resource semantics. This describes the result of invoking the resource.
 - Resource usage restrictions.
- Locally defined data types.
- Exception definitions
- Variability provided by the interface
- Quality attribute characteristics of the interface
- Element requirements
- Rationale and design issues
- Usage guide

Documentation across views

- How the documentation is laid out and organized so that a stakeholder of the architecture can find the information he or she needs efficiently and reliably. This part consists of a view catalogue and a view template.
- What the architecture is. Here, the information that remains to be captured beyond the views themselves is a short system overview to ground any reader as to the purpose of the system; the way the views are related to each other; a list of elements and where they appear; and a glossary that applies to the entire architecture.
- Why the architecture is the way it is: the context for the system, external constraints that have been imposed to shape the architecture in certain ways, and the rationale for coarse-grained large-scale decisions.

Architecture rationale

- An architecture description shall include a rationale for each architecture viewpoint identified for use in terms of stakeholders, concerns, model kinds, notations and methods
- An architecture description shall include rationale for each decision considered to be a key architecture decision
- An architecture description should provide evidence of the consideration of alternatives and the rationale for the choices made.
- When an architecture description is being prepared for a legacy or existing system, the rationale for this legacy system architecture should be included, if known.



The importance of rationale

- Not enough to document what the architecture is.
- The documented architecture, views etc., show the result of deliberations but not the deliberations themselves
- It is important to know why the system is the way it is
- Lacking a rationale, maintenance tends to compromise the system's qualities
- Its really important to know which are the bearer walls before starting house renovations



Decision capture

- An architecture description may capture architecture decisions considered to be key to its architecture.
- It is not practical to capture every architecture decision about a system. A decision capture and sharing strategy should be applied by the organization and/or project to establish criteria for selecting decisions to be captured and supported with rationales in the architecture description. Criteria to consider are:
 - decisions regarding architecturally significant requirements;
 - decisions needing a major investment of effort or time to make, implement or enforce;
 - decisions affecting key stakeholders or a number of stakeholders;
 - decisions necessitating intricate or non-obvious reasoning;
 - decisions that are highly sensitive to changes;
 - decisions that could be costly to change;
 - decisions that form a base for project planning and management (e.g. work breakdown structure creation, quality gate tracking);
 - decisions that result in capital expenditures or indirect costs.

Summary

- Architecture documentation has many uses
- Architecture documentation includes all that you have done so far
 - Identification
 - Stakeholders and their concerns
 - Quality attributes
 - Viewpoints & views
 - View catalogue
 - Documentation across views
 - Architecture rationale