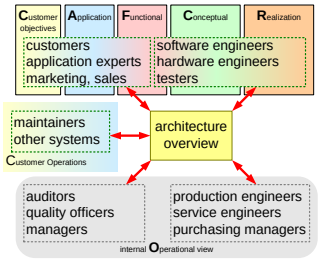
How to Create an Architecture Overview

-



Gerrit Muller
University of South-Eastern Norway-NISE
Hasbergsvei 36 P.O. Box 235, NO-3603 Kongsberg Norway
gaudisite@gmail.com

Abstract

A short specific overview of an architecture is a powerful means during product creation. This article addresses how to create such an overview. One of the main issues is the need for *breadth*, what needs to be included and for whom, and the balancing act of providing sufficient *depth*, what are crucial details that are part of this top-level description. Also the way of describing is discussed, from stakeholder needs to ambiguity and the level of formalism.

version: 1.2 status: draft March 6, 2021

1 Introduction

A broad architecture overview relating specific facts to specification and design choices is a powerful tool in product creation. The architecture overview provides:

- focus on customer and business
- direction and guidance to the project team
- insight in important choices and risks
- attention for issues that go beyond organizational entities

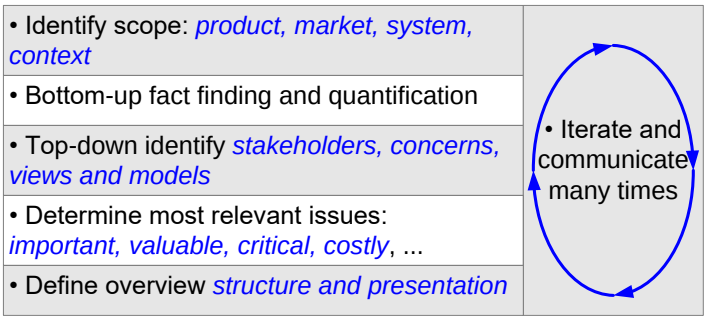


Figure 1: Approach to Create Architecture Overview

These benefits require a compact and to-the-point overview. We present in this article a stepwise approach, see 1, to create such an overview. We recommend to use time-boxes for the steps of this approach. For the first iterations we recommend 15..60 minutes per time-box, for later iterations we recommend hours to maximum a few days per time-box.

The first step is to define the scope of the architecture overview. The next step is bottom-up by exploring facts. We strongly recommend to quantify facts as much as possible, to ensure that this exploration is sufficiently specific. Next step is to work top-down, where we follow IEEE 1471: Who are the stakeholders, what are their concerns, what are relevant views, what models are needed to describe these views? The word *relevant* is more extensively addressed in the next step, where we need to identify the most valuable and most important issues from customer perspective and the most critical and expensive issues from design perspective. This selection step is the most difficult and the most crucial step in the entire approach. The structure and the way to present the overview can now be determined. All the mentioned steps are repeated several times. This iteration is used to incorporate insights obtained in earlier passes. For instance, the scope may move or get more sharper based on new stakeholder insights.

2 Fact Finding and Quantification

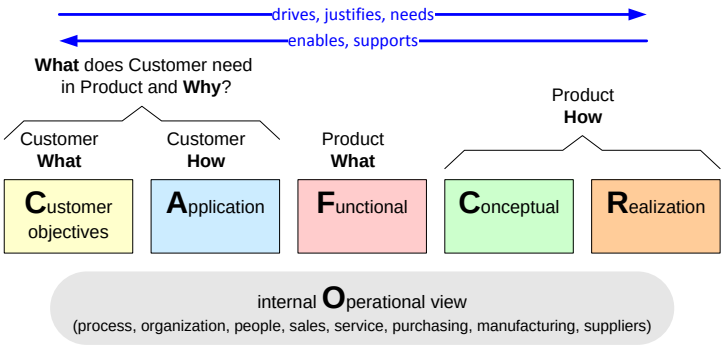


Figure 2: CAFCR model

In Figure 2 we show an extended CAFCR model[6] as a framework for fact finding and quantification. The CAFCR model is a top-level decomposition of an architecture. The *customer objectives* view and the *application* view provide the **why** from the customer. The *functional* view describes the **what** of the product, which includes (despite the name) also the *non functional* requirements. The **how** of the product is described in the *conceptual* and *realization* view, where the conceptual view is changing less in time than the fast changing realization (Moore's law!). The extension provided here is the *internal Operational view* describing all the company internal considerations, ranging from process, organization, people, to goods flow related processes, such as sales, purchasing, and manufacturing.

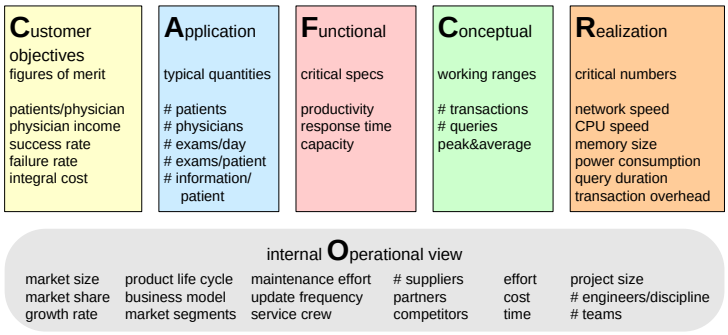


Figure 3: Bottom up fact finding and quantification

The fact finding process is a quick exploration of all views. The result of this exploration is a broad set of still disconnected data-points: the pieces of the jig-saw puzzle. The set of data-points is far from complete after the first explorations.

The main purpose of this bottom-up step is to ground the discussion with facts, to prevent a common pitfall of architecture overviews: a high degree of too abstract and too academic statements.

In order to achieve this grounding it is strongly recommended to quantify facts as much as possible. Figure 3 shows an imaginary example of an electronic patient record (EPR) service. For all views a number of quantifiable aspects are mentioned that can be addresses during the fact finding.

3 Architecture Description and IEEE 1471

Many frameworks and standards exist to create architecture descriptions, see for instance the white paper at the System Architecting Forum[1] [7]. In this paper we focus only on IEEE 1471 and CAFCR, although all other frameworks and standards can provide inspiration for creating architecture overviews.

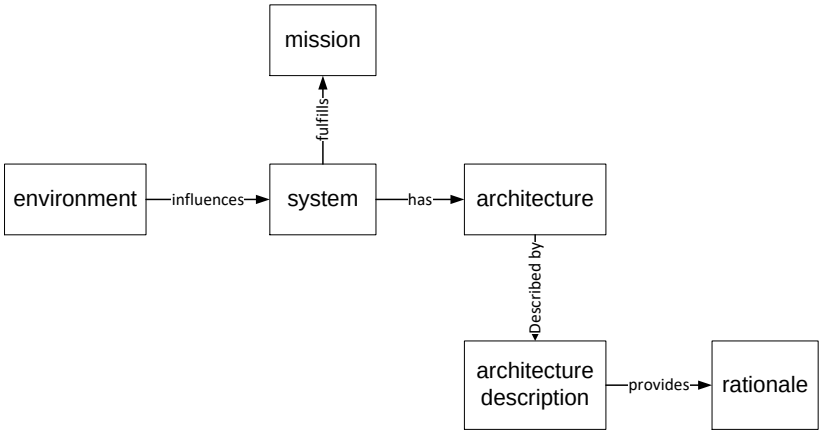


Figure 4: IEEE 1471 top level of the information model

IEEE1471 [1] elaborates the notion of architecture descriptions. It positions the system in it's environment, see Figure 4. The environment is an important aspect for the architecture, it influences the system. It also shows that the system has a purpose, called *mission* by IEEE 1471. The architecture description has to provide a rationale for the architecture, based on environment and mission at the one hand and solution options at the other hand.

The framework introduces a number of important concepts, as shown in Figure 5:

Stakeholders People or organizations that have an interest in the system under consideration.

[1] `http://www.architectingforum.org/`

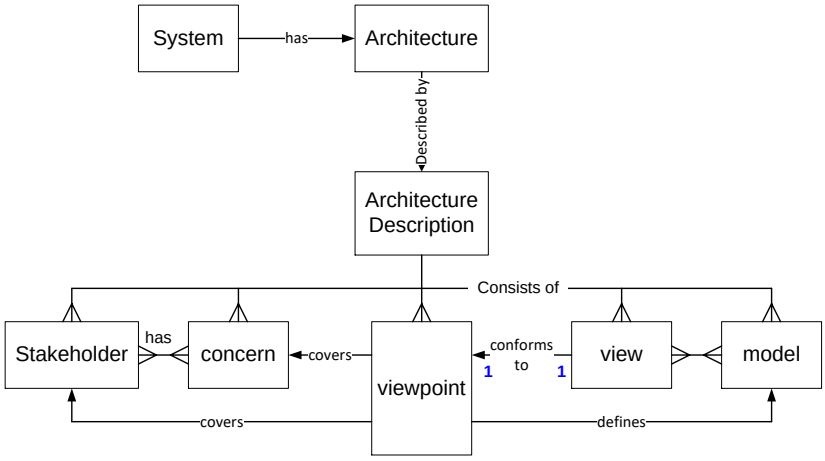


Figure 5: IEEE 1471 simplified information model

Concerns The articulation of the needs and worries of the stakeholders.

Viewpoints The points of view used to describe part of the problem or solution. IEEE 1471 makes a subtle difference between *view* and *viewpoint*. We ignore this difference here.

Models Frequently used method to make problem and solution descriptions.

Architecture description The combination of stakeholders, concerns, viewpoints and models to describe the architecture of a system.

The main contribution of IEEE 1471 is to provide a framework that covers all of these aspects. The individual concepts have been in use by many architects for a long time.

On top of providing the framework, IEEE 1471 also recognizes the fact that complete consistency in the entire architectural description is an illusion. The real world of designing complex systems is full of stakeholders with fuzzy needs, often contradictory in itself and conflicting with needs of other stakeholders. The insights of individual designers are also full of different and changing solutions. This notion of incomplete consistency is not an excuse for sloppy design; quite the opposite: recognizing the existence of inconsistencies is a much better starting point for dealing with them. In the end, no important inconsistencies may be left in the architecture description.

IEEE 1471 makes another interesting step: it discusses the architecture *description* not the *architecture* itself. The *architecture* is used here for the way the system is experienced and perceived by the stakeholders[2].

[2]Long philosophical discussions can be held about the definition of **the** architecture. These

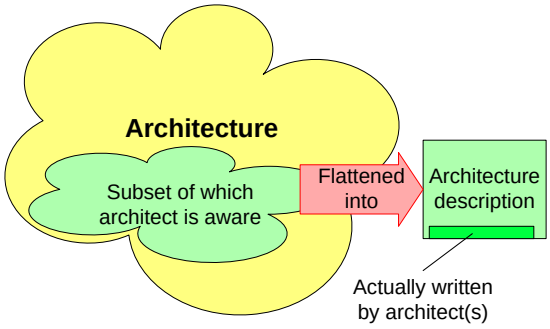


Figure 6: The architecture description is by definition a flattened and poor representation of an actual architecture.

This separation of *architecture* and *architecture description* provides an interesting insight. The *architecture* is infinite, rich and intangible, denoted by a cloud in figure 6. The *architecture description*, on the other hand, is the projection, and the extraction of this rich *architecture* into a flattened, poor, but tangible description. Such a description is highly useful to communicate, discuss, decide, verify, et cetera. We should, however, always keep in mind that the description is only a poor approximation of the *architecture* itself.

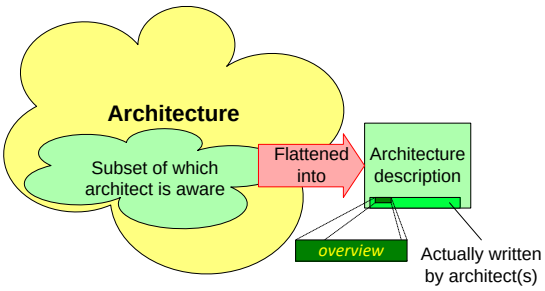


Figure 7: Architecture Overview as part of the architecture description

The overview of the architecture is again only a fraction of the architecture description, see Figure 7.

The role of a system architecture description can be formulated as:

- Guiding and constraining framework
- Spanning from opportunity exploration via development, manufacturing to support and retirement

discussions tend to be more entertaining than effective. Many definitions and discussions about the definition can be found, for instance in [3], [2], or [4]

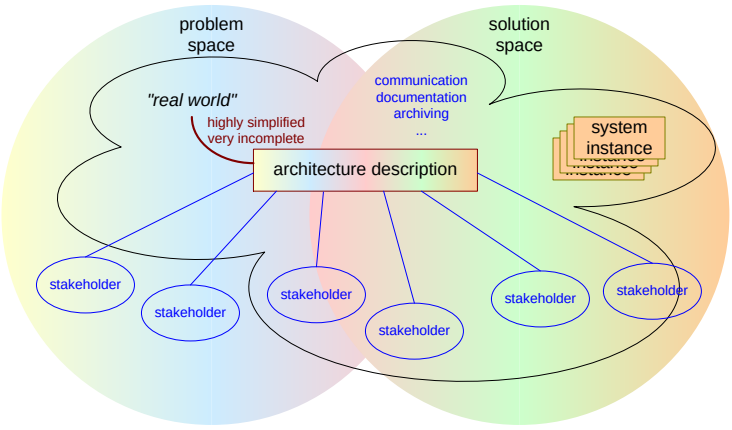


Figure 8: Role of Architecture Description

- Supporting communication and decision-making
- Providing an audit trail from problem/opportunity to solution

Figure 8 shows the breadth of the architecture description, bridging the problem space and the solution space, and as communication means with many different stakeholders, also in both problem space as well as solution space.

4 Presentation and Writing Aspects

The target audience of an architecture overview is a broad set of stakeholders, see Figure 9, with different interests, knowledge and backgrounds. Despite these differences the overview must be *clear* and *easily accessible* for the entire target audience. Note that also the information processing characteristics of the stakeholders differs, some are auditive, some are tactile, and some are visual.

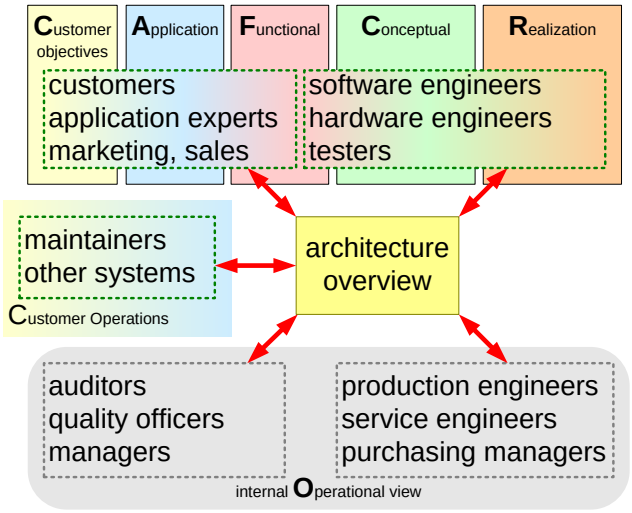


Figure 9: Stakeholders of an Architecture Overview

The architecture overview does not have to be a conventional document. Figure 10 shows a number of forms and mediums that can be used for the architecture overview. The big question mark in this figure indicates that even more creative forms may exist that serve even better in communicating the architecture essence.

We recommend to provide both visual as well as textual information. Figure 11 shows that the core of the architecture information should be partially visual, as diagrams, tables and lists, and partially textual. The text should explain the visual information and glue the information into a coherent set of information.

Figure 11 also provides a size indication for an overview. The meta-information of the overview should be 2 pages or less, because the meta-information distracts from the actual architecture. Some meta-information is necessary to manage and understand the overview in the organizational context. The ”payload” of the overview, the real overview of the architecture, should be 18 pages or less. If more than 18 pages are needed then this is an indication that the essence of the architecture is not yet articulated. If the overview is too long, then readers have the tendency to skip parts, or worse the total. A long overview is a threshold for reading. Note that the architecture description will be more than the overview only. Relevant architectural

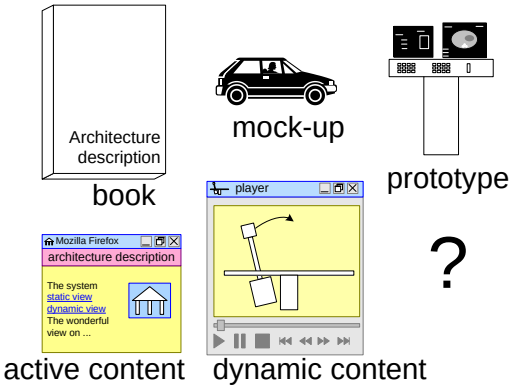


Figure 10: Many alternatives exist for form and medium of the architecture overview

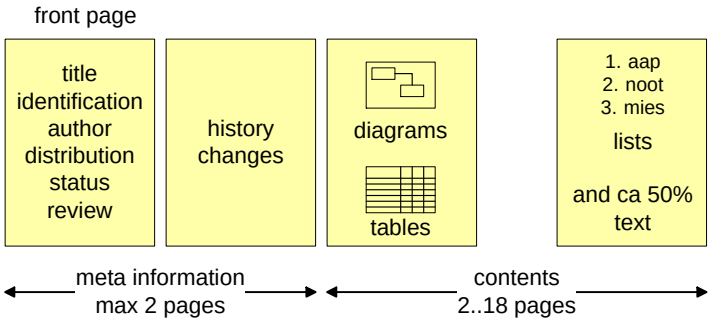


Figure 11: Payload: the Ratio between Content and Overhead

information can be captured in other parts of the architecture description.

The language used in the text should be clear and ”plain English”, see `http://www.plainenglish.co.uk/plainenglishguide.html`. This guide addresses the aspects mentioned in Figure 12.

There are many possible dimensions that can be used to structure the overview. Unfortunately no single dimension is ideal to structure. The structure of the architecture overview must serve it's communication purpose. In other words the structure itself is less important than clarity and understandability of the content.

We recommend to get feedback on visualization and text from different potential readers. Not only listen to their immediate feedback, but also observe the impact on them:

- Did the reader understand the essence?
- Did text or visualizations suggest unexpected interpretations?

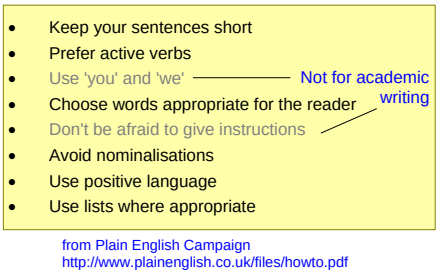


Figure 12: Recommendations from the guide ”How to write in plain English”

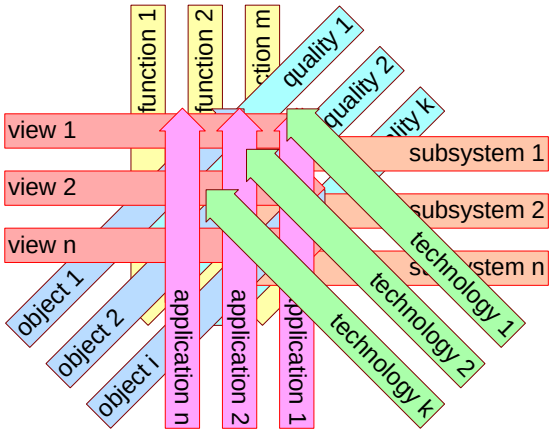


Figure 13: Many possible dimensions to structure the architecture overview

5 Acknowledgements

This article was triggered by the organization of the Dutch Championship ICT Architecture `http://www.nkictarchitecture.nl`. The initiators, Jan Turk, Hans Bot, and Gert Florijn, of this event stimulate the development of the architecting discipline and as such triggered the creation of this article, and provided valuable input and feedback. Erik Philippus sent me the link for the ”plain English” guide, the Plain English Campaign `http://www.plainenglish.co.uk/` has given permission to use their list of recommendations.

References

[1] Architecture Working Group (AWG). *IEEE Recommended Practice for Architectural Description of Software-Intensive Systems*. The Institute of Electrical and Electronics Engineers, Inc., 2000.

[2] Dana Bredemeyer. Definitions of software architecture. http://www.bredemeyer.com/definiti.htm, 2002. large collection of definitions of software architecture.

[3] Derek K. Hitchins. Putting systems to work. http://www.hitchins.co.uk/, 1992. Originally published by John Wiley and Sons, Chichester, UK, in 1992.

[4] Carnegie Mellon Software Engineering Institute. How do you define software architecture? http://www.sei.cmu.edu/architecture/definitions.html, 2002. large collection of definitions of software architecture.

[5] Gerrit Muller. The system architecture homepage. http://www.gaudisite.nl/index.html, 1999.

[6] Gerrit Muller. CAFCR: A multi-view method for embedded systems architecting: Balancing genericity and specificity. http://www.gaudisite.nl/ThesisBook.pdf, 2004.

[7] Gerrit Muller and Eirik Hole. Architectural descriptions and models. http://www.architectingforum.org/whitepapers/SAF_WhitePaper_2006_1.pdf, 2006.

History

Version: 1.2, date: July 20, 2006 changed by: Gerrit Muller

- minor textual changes

Version: 1.1, date: July 5, 2006 changed by: Gerrit Muller

- added context to approach
- added customer operations to stakeholder diagram

Version: 1.0, date: July 5, 2006 changed by: Gerrit Muller

- added top level IEEE 1471 diagram
- added diagrams with forms and mediums
- added diagram and text about structure
- changed status to draft

Version: 0.1, date: July 4, 2006 changed by: Gerrit Muller

- Got permission to use list of recommendations from Plain English Campaign
- changed status to preliminary draft
- choose a logo

Version: 0, date: June 27, 2006 changed by: Gerrit Muller

- Created, no changelog yet