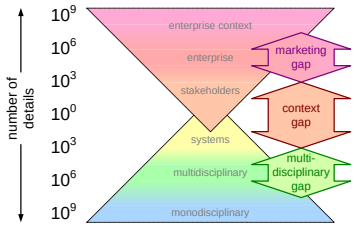
Dynamic Range of Abstraction Levels in Architecting

-



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

This paper has been integrated in the book "Systems Architecting: A Business Perspective", http://www.gaudisite.nl/SABP.html, published by CRC Press in 2011.

Abstract

One of the challenges in architecting is to span many orders of magnitude in the level of abstraction. The system of interest itself can be viewed on many levels of abstraction. However, the context of customers, life cycle, and related products adds a few more orders of magnitude to be spanned.

version: 0.1 status: draft September 6, 2020

1 Introduction

System architects need the capability to “zoom in” and “zoom out”. A tremendous dynamic range of abstraction has to be covered from high level business and customer objectives to detailed design decisions at engineering level. The system-of-interest itself spans many abstraction levels. However the architect has to look beyond the system-of-interest itself, towards the customer context, the life cycle, and to related products.

2 From System-of-Interest to Context

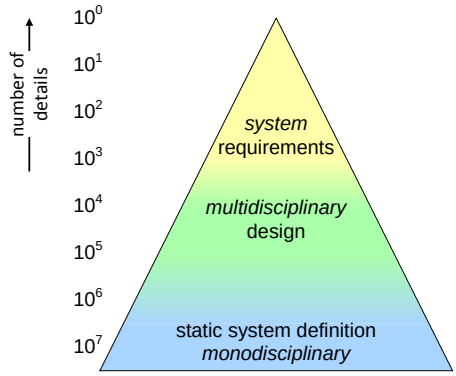


Figure 1: Connecting System Specification to Detailed Design

The translation of the product specification of the system-of-interest into detailed mono-disciplinary design decisions spans many orders of magnitude. The few statements of performance, cost and size in the system requirements specification ultimately result in millions of details in the technical product description: million(s) of lines of code, connections, and parts. The technical product description is the accumulation of *mono-disciplinary* formalizations. Figure 1 shows this dynamic range as a pyramid with the system at the top and the millions of technical details at the bottom.

The current system-of-interest is most often part of a broader set of products that evolves over time: the product family or portfolio. The aggregate amount of details in the product family or portfolio can be several orders of magnitude larger than the amount of details for one system. Figure 2 shows the increase of the dynamic range from system to portfolio.

Architects also have to take the context of the system into account, from both customer as well as business perspective. We can transform the portfolio pyramid from Figure 2 into Figure 3 to show the number of details of a portfolio in its

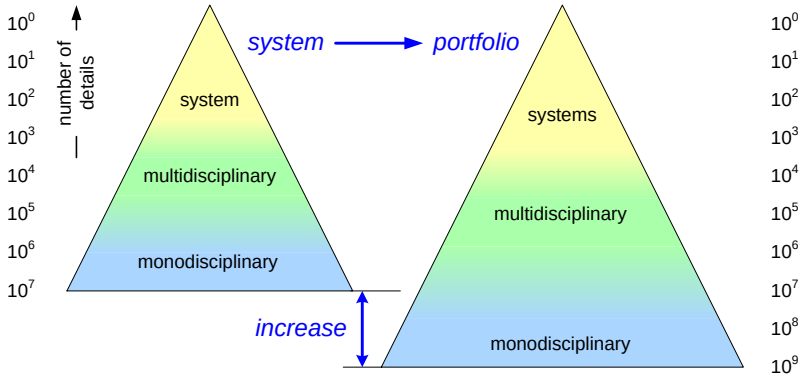


Figure 2: From system to Product Family or Portfolio

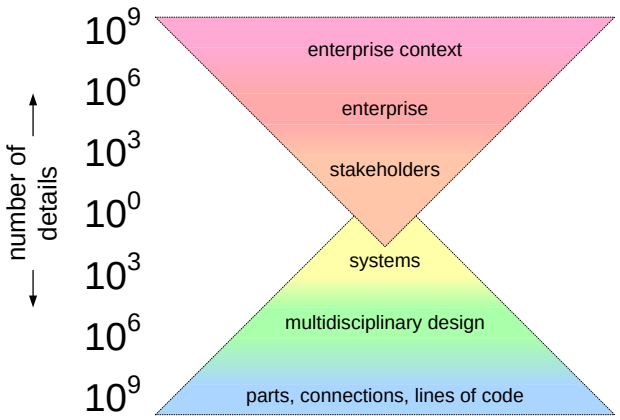


Figure 3: Product Family in Context

context. The context is also shown as a pyramid, representing the fact that in the outside world, where systems are actually used, can be viewed at many levels of abstractions.

3 Architecture and Architecting

The challenge of developing an architecture is to capture the essence of both the systems to be build as well as the contexts where systems are being created and used. Figure 4 shows that most of an architecture covers the higher abstraction levels. An architecture needs to abstract from most details to facilitate the capture of the essence. Only a simplified description or model can be used at system level

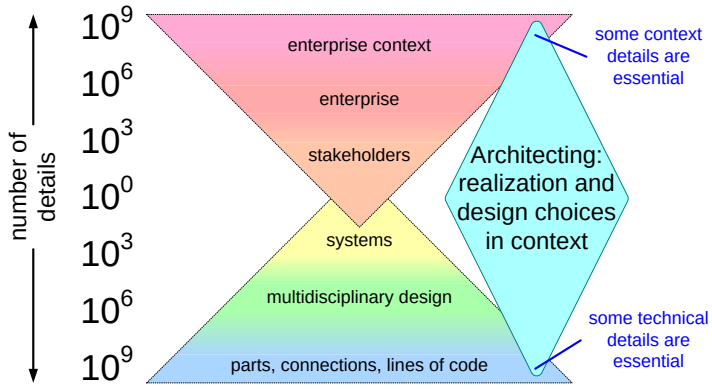


Figure 4: Architecture: the Essence of System and Context

to reason and facilitate communication.

However, some crucial details either from mono-disciplinary area or from the customer or business contexts might have to be included. Quite often the devil is in the detail. Hence known crucial details are part of an architecture description or model.

Note that architectures do have a scope:

System architecture captures the essence of a system in its context. Note that the *system context* includes the product family or portfolio. However, the focus of the *system architecture* is on the system itself, and as such will position this system in the broader portfolio.

Family architecture captures the essence of the family of systems and its context. The focus is now on the family, explaining how different products can support specific market needs, and providing guidance to harvest synergy between products.

Portfolio architecture is similar to family, but at an higher aggregation level.

Architecting involves all activities to create an architecture: exploring details in system(s) and context, communication, design, specification, making decisions et cetera. In other words architecting combines *external* zoom-in and zoom-out (fact gathering and communication) with *internal* zoom-in and zoom-out (specification, design, integration).

4 Revisiting Design and Engineering

We can revisit the terms *design* and *engineering* based on the dynamic range of abstraction levels, as shown in Figure 5.

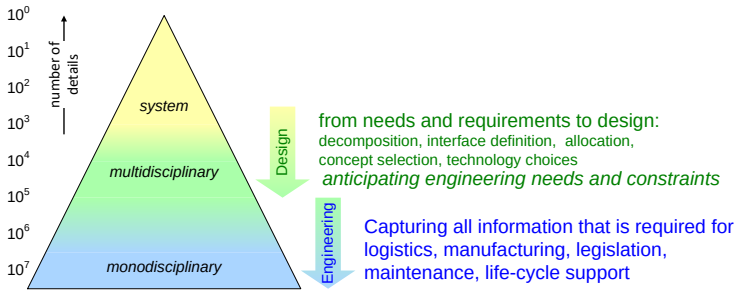


Figure 5: Positioning *design* and *engineering* in the dynamic range of abstraction levels

Designing is the activity to get from needs and requirements to a design: decomposition, interface definition, allocation, concept selection, technology choices, etc. The design has to anticipate the engineering needs and constraints.

Engineering is capturing all information that is required for the Customer Oriented Process, such as logistics, manufacturing, legislation, maintenance, life cycle support.

Engineering and design mostly takes place internally in the organization, with the exception of the communication with external suppliers.

5 Architecting and Design in Practice

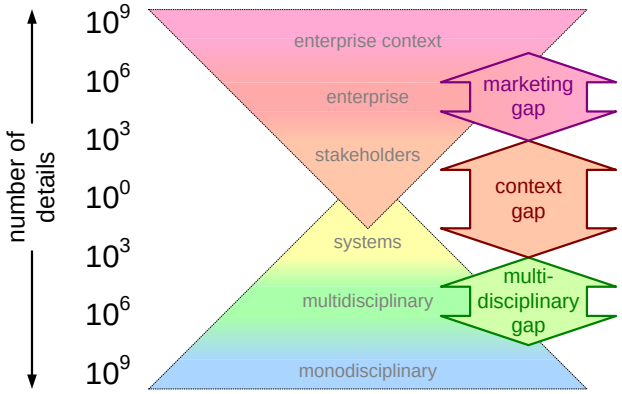


Figure 6: Frequently observed gaps in practice

In practice, several problems can be observed in most organizations that can

be explained by "gaps". Figure 6 shows some gaps that can be observed in many organizations:

Multi-disciplinary gap is the gap between product specification and detailed design decisions.

Context gap is the gap between stakeholders and product specification.

Marketing gap is the gap between the detailed outside world with billions of individuals and our abstracted understanding in terms of stakeholders, concerns, and needs.

Architects have a core role in closing and preventing the multi-disciplinary and the context gaps. In practice, the marketing managers do have the irresponsibility for the marketing gap with their knowledge of stakeholders, enterprises, and enterprise contexts.

The multi-disciplinary gap, from specification to detailed design, is often bridged by experience: older engineers make decisions based on their past experiences. Note that these decisions are often right. The problem is that the implicit nature of these decisions does not facilitate communication, review, or discussion. Worse is that this knowledge gradually disappears from the organization, making further evolution even less transparent.

The context gap, how marketing research information relates to choices in the product specification, requires an extrovert focus of architects. Early in their careers many architects look inward (to design and engineering) and too little outward (to customers and other stakeholders in the Customer Oriented Process). Architects make major development steps when they start to address both gaps in a balanced way.

References

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

History

Version: 0.1, date: February 7, 2011 changed by: Gerrit Muller

- added third gap
- merged changes from SABP

Version: 0, date: July 7, 2010 changed by: Gerrit Muller

- created by integrating fragments from other papers