# Challenges in Teaching Conceptual Modeling for Systems Architecting

Gerrit Muller

HBV, Kongsberg, Norway
Gerrit.muller@hbv.no

**Abstract.** Systems architecting requires systems architects that are able to understand, reason, communicate and make decisions about system specification and design. Systems architects use multiple views on a system and its context to achieve that. Systems architecting uses conceptual modeling as tool for understanding, reasoning, communication and decision making.
After ten years of teaching, we reflect on the challenges of teaching the conceptual modeling ability to practitioners in companies. We find that we have to stretch participants multifold to let them evolve from designer into systems architect and conceptual modeler.

**Keywords:** systems architecting, system context, competence, teaching

## 1 Introduction

Many companies need systems architects who are able to lead development projects to ensure that specifications, design, and implementation satisfy stakeholders' needs and expectations. In the past ten years, we have taught systems architecting courses in companies and at universities. These systems architecting courses teach conceptual modeling as means for systems architecting.
Many of these companies realize that the systems architecting competence requires hard and soft skills and significant experience. Typically, these companies compose an educational program consisting of some technical depth, systems architecting methods and mindset, and soft skills. Systems architecting courses teaching conceptual modeling typically fit in such broader program.
We observe that the entire systems architecting training program continuously stretches the majority of participants. We have to challenge participants to get them out of their comfort zone and out of their original mental paradigms.
In this paper, we discuss the challenges of teaching conceptual modeling in this context. Section 2 explains the broad meaning of systems architecting. Section 3 elaborates how conceptual modeling supports the systems architecting efforts. Section 4 discusses the teaching challenges.

## 2 Systems architecting and Systems architecting Challenges

Rechtin and Maier [1] provide a foundation for Systems architecting in the book "The Art of Systems Architecting". The systems architecting activity has as purpose to create and maintain an systems architecture that ensures that systems and their designs fulfill stakeholder needs and expectations. Figure 1 visualizes a top-view on systems architecture, where we decomposed stakeholder needs in customer value proposition and business proposition. These propositions drive the system requirements that in turn drive the system design. Vice versa, design and requirements enable customer value and business propositions.

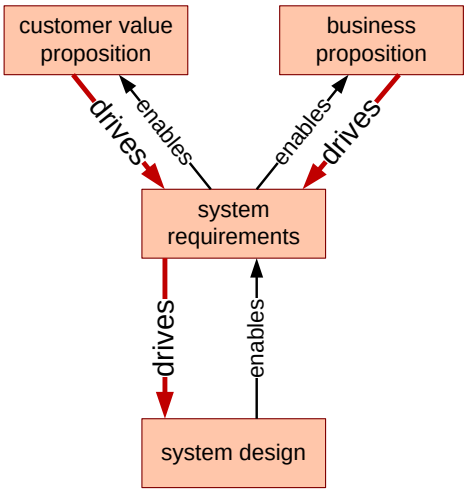


**Fig. 1.** Systems architecture top-view, relating customer value and business proposition to system requirements to system design

TRIZ (Altshuller 2000) positions the system of interest in a 3 by 3 matrix, as shown in Fig.2 right-hand side. In this matrix, the vertical direction is the system scope. The system of interest is part of a larger super system (shown above). The system-of-interest itself is partitioned in subsystems (shown below). The horizontal dimension I the time axis, running from past to future. Fig. 2 adds at the left-hand side the organization that dominates the system scope. For the system-of-interest that is the developing organization, while the customer organization owns and operates the super system.

A typical challenge in developing systems is that we have knowledge from past systems that we like to re-use for future systems. However, innovation may require other and new knowledge than the knowledge from the past.

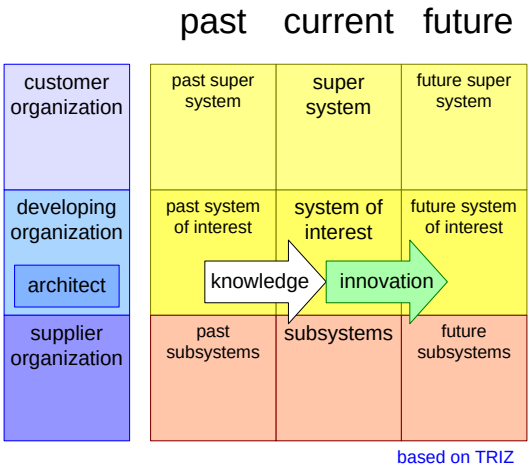


**Fig. 2.** The current system of interest in broader perspective

Fig. 3 overlays Fig.2 with systems architecture that captures and guides past, current, and future, and super system, system, and subsystems. All organizations contribute to the systems architecting activity, with the systems architect as owner and conductor of the systems architecting activity.

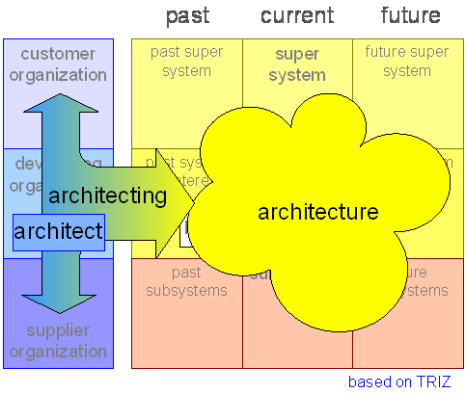


**Fig. 3.** The systems architecture as overarching description

Fig. 4 annotates Fig. 2 with challenges for the systems architecting activity. The past systems bring legacy constraints. The broad context of the system in the super system and its related organizations are complex and large; size and complexity are significant challenges for systems architecting. The broader world of super systems is heterogeneous in stakeholders, and their concerns and needs. These stakeholders tend to express themselves ambiguous. Moving into the future opens a new unknown world full of uncertainties.

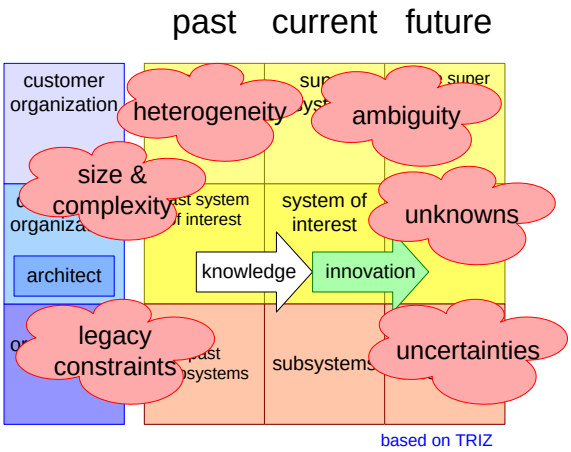


**Fig. 4.** Systems architecting challenges

## 3 How Conceptual Modeling supports Systems architecting

Systems architects have a need for understanding, communication, reasoning, and making decisions about problem and solution space. Conceptual modeling is a good tool to support these needs [3, 4]. Bonnema [5] describes how systems architecting benefits from modeling, using key drivers and quantified technical budgets. Borches [6] combines the use of conceptual modeling with A3s to create compact A3 Architecture Overviews. Muller, Wee, and Moberg [7] describe an example of the results of teaching conceptual modeling and A3s.

Bonnema [5] and Borches [6] describe three core views in systems architecture descriptions: parts, dynamics (functionality), and quantified characteristics, see Fig. 5. We observe that many stakeholders in the developing organization think and act on the structure (the parts and the interfaces). However, the prime interests of customers are the capabilities they get from the system. Capabilities are (quantified) performance characteristics and the related dynamic behavior. Architecture relates these views.

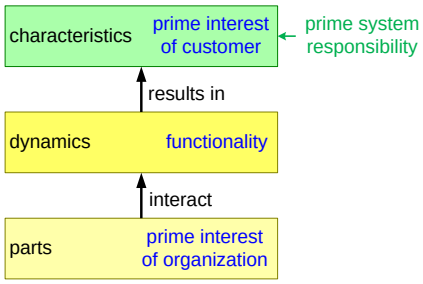


**Fig. 5.** Systems architecture = Structure (parts and interfaces) + dynamics + characteristics; Conceptual Modeling captures these relations

Since 2005, we have been teaching courses in architectural reasoning at companies and universities. Fig. 6 shows the core principle, objectives, and recommendations in these courses. The main principles behind the approach are feedback (do we move in the proper direction, does our solution solve the problem) and being explicit (make issues tangible by being explicit, for example by quantifying).

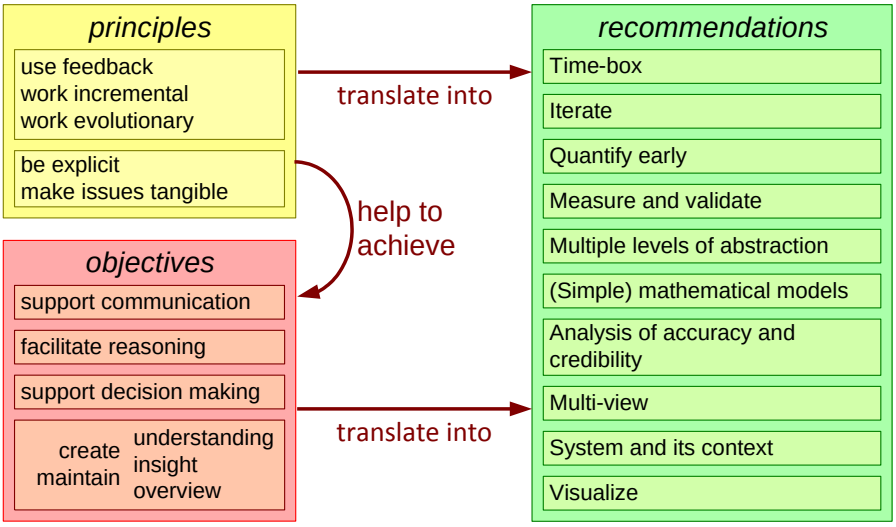


**Fig. 6.** Recommendation for Conceptual Modeling

These principles help to achieve our main objectives of understanding, communication, reasoning, and decision making. The principles and recommendations translate into a concrete set of recommendations.

**Time-box and iterate.** We translate the principle of feedback is into an approach where modelers spend a limited amount of time on a topic. However, they revisit the topic many times (iteration) to include insights gained from other topics; see [8].

**Quantify early.** We translate the principle of being explicit into early quantification. The idea is that quantification forces designers to be explicit; discussing a number brings sharpness in the discussion. However, stakeholders should be aware of the limitations of quantification in early phases. Numbers may and probably will change, and they need validation.

**Measure and validate.** The early quantification benefits from measurements (what is current practice) and validation (what evidence and arguments do we have for the numbers).

**Multiple levels of abstraction** are countermeasures for size and complexity. In fact, abstraction is the core of conceptual modeling. However, a challenge is to connect high-level abstractions to lower-level abstractions, such that the systems architecture offers concrete guidance to designers and engineers.

**Simple mathematical models** are instrumental for being explicit and for understanding and reasoning about problems and solutions. Mathematical formulas capture rela-

tions between parts and components in such way that we can reason about these relations.

**Analysis of accuracy and credibility** is an additional consequence of early quantification and the need for validation. Systems architects and their stakeholders need to understand the accuracy and credibility of the information they are acting on.

**Multi-view.** Systems architects need many views to describe systems architecture. ISO/IEC/IEEE 42010 [9] defines a view as the combination of a stakeholder and a concern. This definition results in a large number of potential views. In practice, systems architects use 8 to 12 views to describe systems architecture [10], since more views are not manageable and stakeholders are unable to cope with too many views.

**System in its context.** Understanding and reasoning about a system requires understanding of its context. The rationale of systems architecture decisions is often in the context. Consequently, the conceptual modeling has to cover the context too.

**Visualize** enhancing understanding, reasoning, communication, and decision-making. A picture says more than a thousand words is a saying that illustrates the value of visualizations. From research in modeling [11.12.13.14], we know that visualizations stimulate discussions between stakeholders.

## 4 The current educational approach

The course format has evolved over the past 10 years. The core of the format, however, has stayed the same over all these years. The course primarily stimulates participants to experience the views and various modeling techniques by applying them in the classroom with guidance from the teacher. The teacher offers small nuggets of theory followed by an assignment of 15 to 40 minutes (time-boxes). The participants preferably use a case from their daily work environment, since these cases suffer from "real-world" complexity as shown in Fig. 4. The only exception is the first assignment, where we use an elevator example explaining essential aspects of conceptual modeling, as described in [15], such as the need for complementary models. Fig. 7 shows the various conceptual models and visualizations that we use in this elevator example. The parts, dynamics, and (quantified) characteristics are all present.

The assignments guide the participants through three iterations for their case to let them experience the growth and evolution of their case insight. The first iteration, covering supersystem, system, and subsystem level, makes 6 steps of 15 minutes to explore the "playing field" with a focus on the current situation. Following two iterations make 5 to 6 steps of 20 to 40 minutes with a focus on extending the system in functionality or performance. These longer time-boxes allow some elaboration per model. The participants work during all assignments on flip charts, which they attach to the wall. In this way, the case "grows" on the wall, helping participants to see all views and the increasing insight in many related aspects; see Fig. 8 for a typical classroom setting.

The final step is integration and convergence of all steps into a limited set of lines of reasoning, to create overview. This last step is quite challenging for the participants. All earlier steps create some information or insight. After so many steps and 3

iterations, the amount and variety of information tends to overwhelm participants. In this last step, they have to recover the overview by selecting the most relevant information form systems architecting perspective.

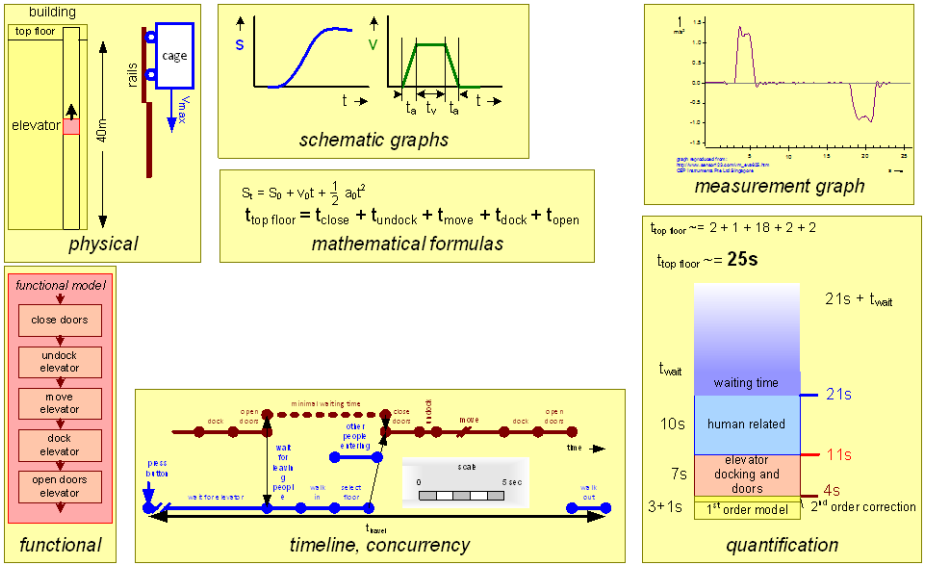


**Fig. 7.** Example of Conceptual Models and Visualizations from an Elevator

**Fig. 8.** A typical class room during the course

In the homework after the course, participants elaborate the models they made in the classroom. However, outside the classroom, they can easier access knowledge in their company and validate or correct earlier assumptions. They transform most models from flip chart to electronic diagrams, for example in PowerPoint or Visio. Again, the most challenging step is ordering the models in a “T-shape" presentation. A T-shape presentation is a presentation that guides the audience top-down through the case , starting broad in setting the context, explaining solution options in depth, and con-

cluding broad by showing impact and recommendations. The challenge in this last is to make the mental shift from chaotic divergence to structured convergence.

## 5 Challenges in Education

In the 10 years of teaching in company specific programs for among others semiconductors, telecom, defense, health care, and electronics, and for master programs in systems engineering, we observe a number of re-occurring challenges, described in the following subsections

### 5.1 From mono-discipline to multi-disciplinary

The first step many participants have to take is the step from a single discipline to a multi-disciplinary world. Main hurdle is that various disciplines use other languages and mental paradigms. For example, mechanical engineering is a concrete, well-established engineering discipline, used to a physical world with its physical laws and surprises. At the other hand computer scientists and software engineers live in a virtual world, where systems behave according to the rules that we define in an abstract formalism.

### 5.2 From multi-disciplinary to system

Initially, engineers have the impression that systems are simply the sum of the various contributing disciplines. However, once they have experienced emergence of behavior and properties, they start to see that systems engineering is a discipline in itself with its own methods and techniques.

### 5.3 From system to customer and life cycle context

Next step is that designers and engineers have to zoom out further and enter the context of the system. We distinguish the customer context (the context where the system is in operational use) and the life cycle context from conception to decommissioning. The evolution of a system throughout its lifecycle tends to be a revelation. The exploration in the customer context brings a confrontation with all non-technical aspects, especially human factors.

### 5.4 From static to dynamic

Many stakeholders with the developing organization think about the system in static terms: its structure of parts and interfaces. For some of them the static understanding is sufficient, e.g. for purchasing. However, for anyone who needs to understand merging system behavior and performance, the dynamic interaction is crucial. We experience that understanding and reasoning about dynamic behavior is challenging for a significant amount of the participants.

### 5.5 From qualitative to quantitative

Many participants hesitate to quantify. Uncertainty (we do not know yet) partially drives the hesitation. Partially, fear for misuse by stakeholders drives the avoidance of quantification. Participants have to learn to make assumptions (and validate them later) to make progress in understanding and reasoning. Participants who stay qualitative may discover too late that they have been working in the wrong direction.

### 5.6 From well-defined to ill-defined

Every step from designer to systems architect decreases the degree of certainty in the problem definition. Mentally, participants need to grow an acceptance to act in a problem and solution space that is ill defined in many aspects.

### 5.7 From technical to human and nature

When moving from the world of technical design to the context with humans and physical environment, humans and nature confront the participants with major surprises. Especially the humans open a world where participants need other (softer) methods and techniques.

### 5.8 From reactive to proactive and critical

Lastly, the role of systems architect (and conceptual modeler) requires a proactive and critical attitude. Good systems architects are obsessed with the need to understand and the ability to reason. Systems architects need to challenge assumptions and question requirements. Additional challenge is to be critical in a constructive way, such that guides and leads the team.

## 6 Conclusions

Conceptual modeling is a natural tool for systems architecting. Systems architecture descriptions consist of a collection of conceptual models. Teaching conceptual modeling to (potential) systems architects is necessary. However, participants have to take many hurdles to make the step from designer to systems architect and conceptual modeler

## References

1. Rechtin, E., Maier M.: The Art of Systems Architecting. CRC Press, Boca Raton (1997

2. Altshuller, G.: The Innovation Algorithm; TRIZ, systematic innovation and technical creativity. Technical Innovation Center, Worcester, MA, (2000). Translated, edited and annotated by Lev Shulyak and Steven Rodman.
3. Robinson S., Conceptual Modelling: Who Needs It? SCS M&S Magazine 2010 / n2 (April) http://www.scs.org/magazines/2010-04/index_file/Files/Robinson.pdf
4. Davies I., Green P., Rosemann M., Indulska M., and Gallo S. How do practitioners use conceptual modeling in practice? Data Knowledge and Engineering July 2005.
5. Bonnema, G.M., FunKey Architecting - An Integrated Approach to System Architecting Using Functions, Key Drivers and System Budgets, PhD thesis University of Twente, 2008
6. Borches, D. A3 architecture overviews: a tool for effective communication in product evolution, PhD thesis University of Twente, 2010
7. Muller, G., Wee, D., Moberg M.: Creating an A3 Architecture Overview; a Case Study in SubSea Systems, INCOSE 2015 in Seattle
8. Muller, G.: System and Context Modeling -- The Role of Time-boxing and Multi-view Iteration. Systems Research Forum Vol. 3, No. 2 (2009) p139-152
9. ISO/IEC/IEEE 42010:2011 - Systems and software engineering - Architecture description Iso.org. 2011-11-24. Retrieved 2013-08-06
10. System Architecture Forum whitepaper: Architectural Descriptions and Models http://architectingforum.org/whitepapers/SAF_WhitePaper_2006_2.pdf, 2006
11. Engebakken, E., Muller, G., and Pennotti, M., Supporting the System architect: Model-assisted Communication; Systems Research Forum Vol 4, No2 (2010) pages 173-188
12. Rypdal, R.W., Muller, G., and Pennotti, M., Developing the Modeling Recommendation Matrix: Model-Assisted Communication at Volvo Aero, INCOSE 2012 in Rome
13. Polanscak, E., Supporting Product Development: A3 as a tool to capture and visualize Architecture knowledge sharing/A3-assisted Communication and Documentation, master project paper at HBV, 2011
14. B. Stalsberg and G. Muller, "Increasing the value of model-assisted communication: Modeling for understanding, exploration and verification in production line design projects" Proc. INCOSE 2014, Las Vegas, July 2014.
15. Muller, G.: Teaching conceptual modeling at multiple system levels using multiple views, CIRP 2014 in Milano