Systems Research Forum
Vol. 3, No. 2 (2009) 1–14

SYSTEM AND CONTEXT MODELING — THE ROLE OF TIME-BOXING AND MULTI-VIEW ITERATION

GERRIT MULLER
Buskerud University College and Embedded Systems Institute
Pb. 235, 3603 Kongsberg, Norway
TU/e Campus, Laplace Building 0.10, Den Dolech 2
P. O. Box 513, 5600 MB Eindhoven, Netherlands
Gerrit.muller@embeddedsystems.nl

System architects perform Modeling of Systems and their context to support communication with stakeholders, to facilitate reasoning about system requirements and design, to support decision-making, and in general to create and maintain understanding, insight, and overview. The challenge in modeling is to find an appropriate abstraction level, and to make sufficient progress.

In this paper we discuss how time-boxing and iteration over multiple views and models help to address both challenges. Time-boxing and iteration fit in a broader modeling method that we will discuss briefly to provide the background.

We will illustrate the approach with models made as part of the System Modeling and Analysis course at Buskerud University College. An especially interesting finding is that very short time-boxes of 5 to 15 minutes provide sufficient time to create usable first instances of models. Later iterations with somewhat longer time-boxes improve the quality of the initial models.

Keywords:

1. Introduction

IEEE1471 (IEEE1471, 2000, Hilliard, 2001) introduces the notion that architecture descriptions consists of models. Models capture one or more specific views. The combination of stakeholder and a specific concern is a view. Following this line of reasoning many models are needed to create an architecture description. Standards such as DoDAF (DoD, 2003) provide many predefined views.

Currently the term model is also fashionable as a means to streamline the engineering process. *Model-Based Design* (MathWorks, 2009), *Model Driven Architecture* (OMG, 2009), or *Model Driven Engineering* are approaches where models are used to specify, but also to design and engineer. These approaches are propagated by commercial tool vendors, such as The MathWorks and IBM. Tools support the creation and management of models. These models are also used to generate the final realization details, without further human intervention.

In this paper we discuss the use of models as a means of communication and discussion between stakeholders and as a means to reach design decisions that fit in the context and lifecycle of the system-of-interest. These models follow the idea behind IEEE1471, where models relate to stakeholders and concerns. These models serve mostly to help human stakeholders to understand problem and solution space and to reason about alternative options.

The emphasis of this paper is on the need to iterate over views, representations and abstraction levels. Cernosek explains in (Cernosek and Naiburg, 2004) that the value of modeling is to *better understand the situation*, to *craft a better system*, to *build and design a system architecture*, and to *create visualizations of the implementation.* In Fig. 1 of the same paper it is shown that modeling can be used at many levels of abstractions and in many stages of the development lifecycle. This figure, however, also suggests that models with increasing abstraction levels are made later in the lifecycle. As opposed to this, this paper advocates to make more abstract models early in the lifecycle.

2. Research Methodology

The modeling and analysis methods and techniques presented here are researched by applying the methods in the classroom and in workshops in industrial practice. The students or industrial employees are guided through the modeling approach in typically three to five days. The author of this paper is the teacher c.q. the facilitator as well as the observer and the data collector in a simultaneous way. The main purpose in this phase of research is to collect case material and anecdotal material about successes and failures.

This paper is based on the 2007, 2008, and 2009 editions of this course and on workshops derived from this course in companies in healthcare, semiconductor equipment, food preparation, and in defense systems. From the more than ten different cases and domains we use the manufacturing facility case, modeled in the 2007 class, as illustration for this paper on time-boxing and multi-view iteration. So far about 20 students participated in these courses and about 70 industrial employees participated in the workshops.

3. Case Description

This paper illustrates the modeling approach and especially the time-boxing and multi-view iteration by showing the results of one group of students who modeled a manufacturing facility for the course *Modeling and Analysis*. This course, taught at Buskerud University College (BUC, 2008), is a one-week course followed by a ten-week project. The master students have to model an actual system in its contexts, in a group of two to four students. This actual system is preferably the system they are working on in daily practice, with some actual challenging problem.

This case study, the manufacturing facility, manufactures metal components that require lots of (high precision, high quality) machining and processing. The facility consists of a set of large halls with huge machines, lots of operators and supervisors, and stocking areas. Due to its historic growth the facility is a maze. The combination of high precision machines and experienced operators and process engineers allows the factory to produce the high quality that is demanded by their customers.

The factory produces many different types of components. However, the equipment is often shared in the production line for different components. The organization is somewhat machine centric, partially caused by the high investment cost of the equipment. The IT infrastructure for logistics (ERP) and configuration and data management (PDM) is centralized.

The problem that needs to be addressed is the production planning for a new class of components, to be produced in the near future. This requires extension of the production capacity, but it also requires increased capabilities in size and quality.

Two out of three students of the group who did the modeling actually worked at this facility. During the course both author and students visited the facility, which helped them tremendously to understand the system (the manufacturing facility) and the challenges it is facing.

4. Modeling Approach

Figure 1 shows the objectives of modeling as taught in the course and the principles underlying the course (for all course material see `http://www.gaudisite.nl/MA.html`). The main objectives of modeling a system and its context

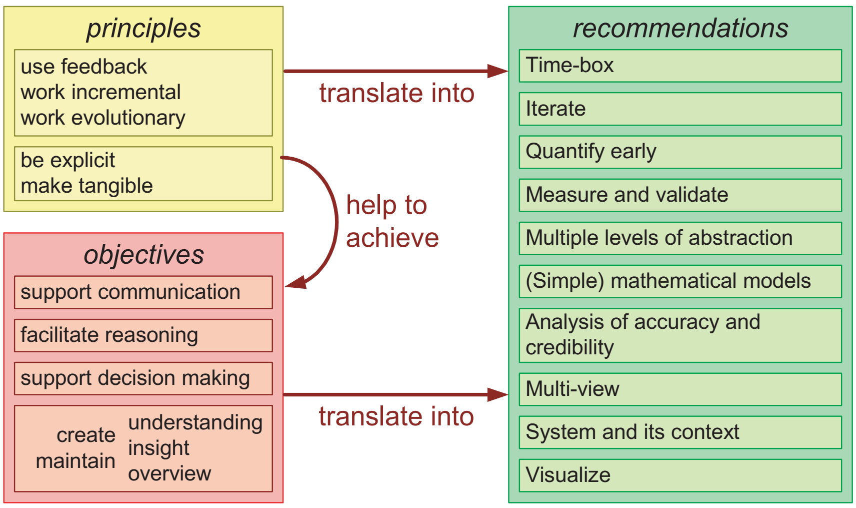

Fig. 1. Guidelines from the course System Modeling and Analysis, showing the objectives, the underlying principles, and the derived recommendations for the modeling approach.

are the following:

- to support communication with stakeholders
- to facilitate reasoning about choices in problem space and solution space
- to support decision-making and
- to create and to maintain understanding, insight, and overview

All of these objectives belong to the core of systems architecting and address problems occurring everywhere in engineering projects of complex systems. We apply a number of principles throughout the course:

- use feedback (a very common principle in many engineered systems, but also very common in biological systems!)
- work incrementally
- work evolutionary (see the book by Gilb about incremental and evolutionary methods (Gilb, 2005))
- be explicit and
- make issues tangible

In complex projects, stakeholders and engineers sometimes use the complexity as an argument to stay vague and to keep many options open. This tendency camouflages problems and limits the insight. The principles *be explicit* and *make issues tangible* are countermeasures for this behavior. The modeling approach offers means to be more explicit and to make issues more tangible.

The objectives of modeling and the underlying principles are translated into ten recommendations at the right-hand side of Fig. 1. In this paper we zoom in on the *time-boxing* and *iteration*. However, the success of the approach itself requires the combined application of all recommendations. In (Muller, 2008) the approach is described in detail, including the other recommendations.

5. CAFCR+: A Framework for Views

One of the essential messages of the modeling course is that the system-of-interest has to be modeled in its broader context. In the course we use the CAFCR+ model as framework. Note that other frameworks, such as DoDAF, could also be used. The main purpose of the framework is to provide guidance for *what* to model, and especially to help designers and engineers to broaden their horizon beyond the system-of-interest.

Figure 2 shows the views in the CAFCR+ model. The *Customer Objectives* and *Application* views model the *what* and *how* respectively of the customer: What does the customer want to achieve and how does the customer achieve these objectives? The *Functional* view is the black box view of the system, the *what* of the system, describing all system requirements: its functionality, its interfaces, but also its quantified performance figures. The *Conceptual* and *Realization* views describe the *how* of the system. The *how* is described at the more re-usable conceptual level, and

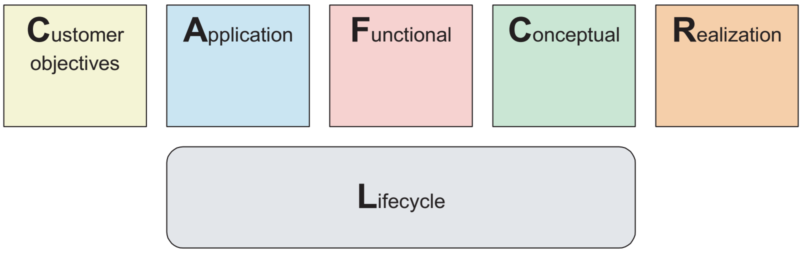

Fig. 2. Views in the CAFCR+ framework.

at the faster changing realization level. Note that many quantified requirements in the *Functional* view can only be verified against a realization. Finally, the *Lifecycle* view describes the *what* and *how* of the lifecycle of the system, from creation, production and maintenance, up to disposal.

The CAFCR+ framework provides six main views. However, in every main view there are many relevant views again. Muller (2004) provides a collection of submethods per main view.

6. Multi-View Iteration

We emphasize the need to iterate quickly over different views as well as the need to zoom-in and to zoom-out. Zooming in and out actually is iteration over different abstraction levels. The idea behind fast iteration is that the *why* of some need, problem or solution is often either in the broader context, e.g., a view with a broader scope, or in some detail, e.g., a view with less scope, or zooming in more in the same view.

Our experience is that many designers tend to work out the same view until they consider it "ready". We, on the other hand, force designers to switch views much more rapidly, by the use of time-boxes. We assert that designers need an understanding of the *why* to make sensible views. After all, how can designers "finish" a view, when they do not understand its *why*?

Figure 3 shows all the flip-overs made by the modeling team working on the manufacturing facility. These flip-overs have been positioned relative to the CAFCR+ model, and the order of creation is denoted by the numbers.

The use of flip-overs early in the process facilitates fast drawing and discussions. It also prevents participants losing themselves in tools or details. Flip-overs invite participants to sketch rather than make the perfect diagram. The flip-overs are taped to the walls of the classroom. The team gradually builds an understanding of the system and its context on the wall. The flip-overs serve as the common memory of the team. Team members can easily remember and refer to the material, since they created it together.

Figure 4 shows how these flip-overs are attached to the wall. Here one participant is explaining what they did during the last time-box. All flip-overs at the

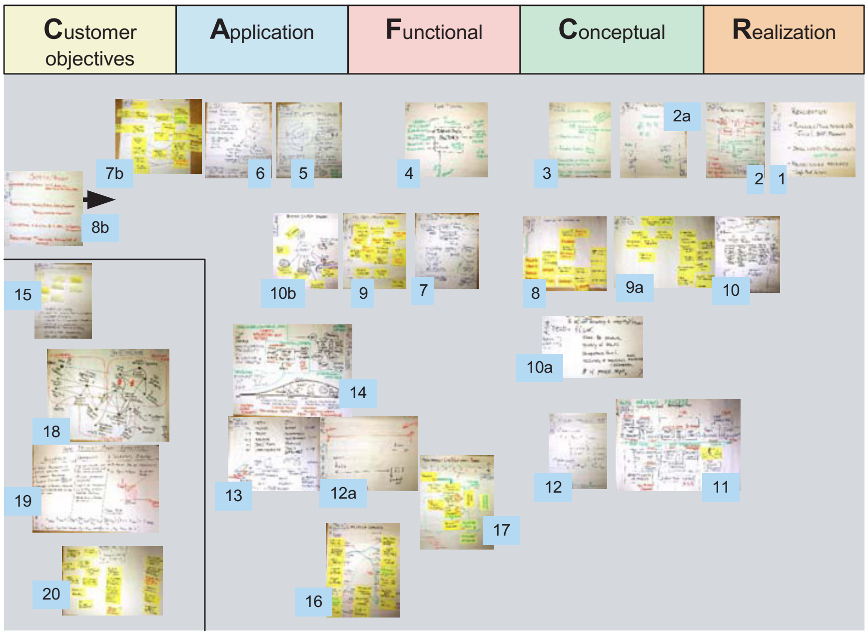

Fig. 3. All flip-overs of the manufacturing team after the one-week course.

Fig. 4. Presentation of intermediate models in the classroom.

right-hand side belong to one single case, the flip-overs at the wall at the back are from another case.

Figure 5 shows the iterations more clearly as a flow in different sessions. The first session is a fast scan of all views. Flip-over 8b is a list of most relevant qualities that

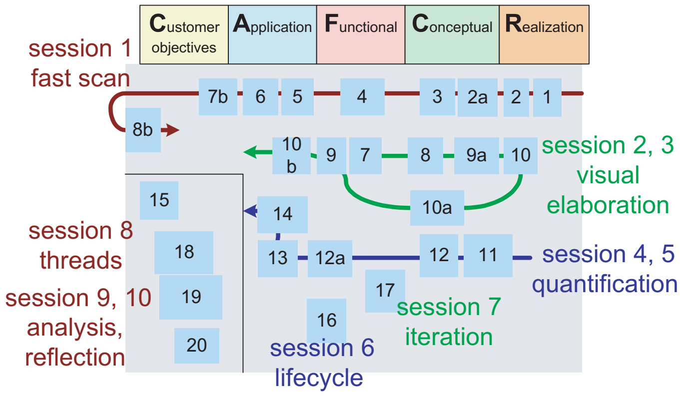

Fig. 5. The same flip-overs as before, annotated with the iterations over the half-day sessions. During the course three iterations have been made over the system in its context, plus some integrating and reflective sessions.

connects all five CAFCR views. The first session is a bottom-up scan, starting at the realization. The idea is that most designers have sufficient concrete knowledge about the system itself. Our experience is that starting top-down, with customer objectives, tends to result in too academic results. By working bottom-up the participants are working upwards by repeatedly asking *why* questions. At the end they present their results top-down to the other teams. Often many questions pop-up in filling in the later views (e.g., customer objectives and application) about the design (*how* is the system realizing this?). These questions will have to be addressed in later iterations.

Sections 2 and 3 start in the middle and work in two directions. In this iteration the views are better visualized and further elaborated. In Secs. 4 and 5 quantifications are added. Quantifications force designers to be sharp and concrete. Section 6 focuses on the lifecycle view. Sections 7 and 8 are attempts to integrate the results and to get more to the "big picture". Sections 9 and 10 analyze the results of the modeling and reflect on the modeling and its results.

In one week the participants have gone at least three times over all views. This provided them with a shared overview of problem and solution space.

7. Time-Boxes, Duration Guidelines

The duration of working on one view must be limited in order to make the iteration principle work. The natural tendency of teams is the need to "finish" a view before going on to the next view. This behavior potentially means that teams might need a large amount of time for a single view, certainly when the view (nearly) matches the expertise area of one of the team members. To counteract this we recommend the use of time-boxes: before hand set the duration of working on a view. The duration of time-boxes can be as short as 5 minutes in early iterations and may increase to days for substantial models later during product creation. The main purpose of time-boxes is to ensure that the designers see other perspectives early, and to prevent the designers getting lost in depth.

The most common situation in Product Creation is that a new product is derived from previous product(s), building on previous knowledge and experience. The course focuses on this mode of incremental innovation rather than on the rare occasion of green-field product creation from scratch. Both managers and designers tend to assume that the team working on the new development shares a common understanding of the previous product(s). Our experience contradicts this assumption: team members have approximately the same understanding, but biased by, and limited to, individual perspectives.

The very first iteration, the quick scan over all views, has as its main goal to create a shared understanding in the team, and to create an overview. Getting a shared understanding is needed at the beginning of the creation process. We have been using time-boxes of 15 minutes for this first iteration many times. Our experience is that 15 minutes is sufficient. Most often the facilitator has to chase the participants to switch to the next view in time. In one occasion, with very experienced designers, less time was used in this first iteration.

For the next iterations time-boxes of typical 40 minutes are used. Such a longer time-box, in combination with a more specific focus, allows a discussion in more depth. Our experience is that deepening an issue can be done in 40 minutes, while giving more time than 40 minutes is often not effective. Teams tend to start to repeat previous discussions after about 30–45 minutes. So we either plan a reporting session after 40 minutes or we provide a new focus.

Figure 6 shows an example of typical iterations and durations of the time-boxes. In this example three iterations are done in two days. The story telling and use case iteration zooms in on one specific use case, providing an in-depth sample of problem and solution space.

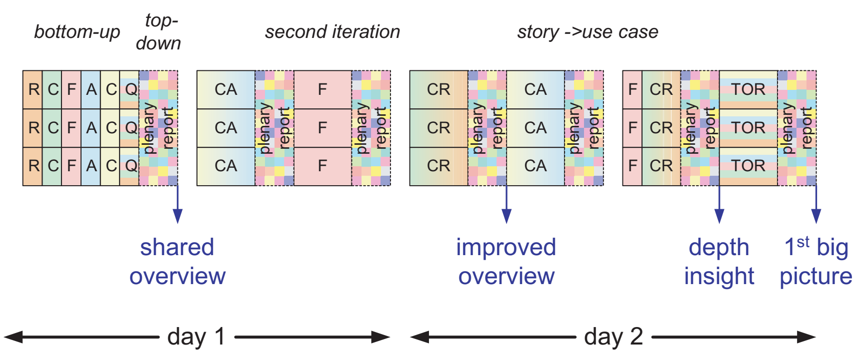

Fig. 6. **Example of Timing of Iterations**. In this example three teams work in parallel. The first morning 15 minute time-boxes are used. Later time-boxes are typical 40 minutes. Plenary reporting typical takes between 30 and 60 minutes.

These short time-boxes work well in the classroom, but we have also applied them successfully in workshops for actual projects. After several fast iterations the time-box duration will have to increase to achieve sufficient depth of modeling and analysis. However, we recommend to keep on using time-boxes also during later design and engineering, with a maximum duration of few days.

8. Case Results

The manufacturing facility was originally modeled as a simple 2D map and process flow, see Fig. 7. Note that this is a cleaned-up version of the original flip-over sketches. Both diagrams have annotations that surfaced during the discussion, e.g., the fact that transportation requires a truck and the number of people involved at the different locations.

Sometimes hot debates emerge about terms, representations and consistency. The facilitator has to stop these discussions early. First a common understanding has to grow, before formalization and standardization of these aspects can be done. Representations and terms will gradually crystallize during the process.

Figure 7 might be perceived as being too simplistic. However, it does serve as a starting point for many other discussions. It also helped the team to see the entire factory as the system-of-interest, rather than as one machine or department. This drawing was made in the quick scan in one of the 15 minute time-boxes.

We skip many of the other models that have been made by this team. After the first two iterations the team identified the lead time and its predictability, from raw material to finished product, as key performance parameters. They also identified that the introduction of new components to be manufactured and the impact of ramping up new production was a hot issue. As a consequence they decided to model the lead time further.

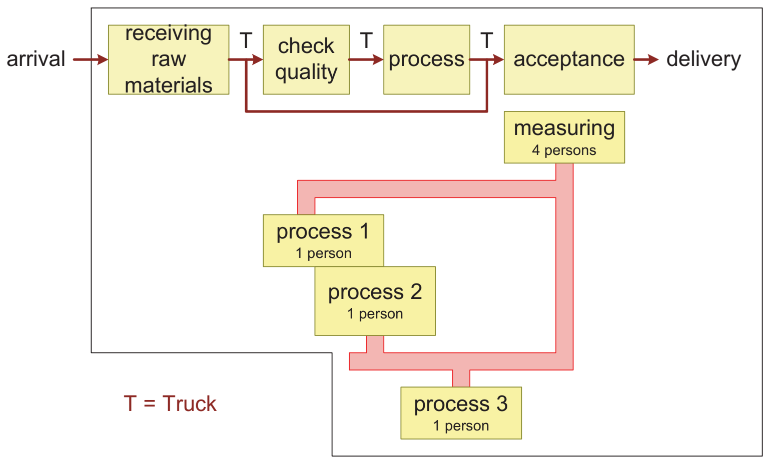

Fig. 7. Initial 2D model and simplified process flow of the factory.

$$t_{\text{lead time}} = t_{\text{processing total}} + t_{\text{handling}}$$

$$t_{\text{processing total}} = \sum_{\text{all processes}} t_{\text{processing}_{\text{process}}}$$

$$\text{e.g. } t_{\text{drill 1..n}} + t_{\text{grind 1..m}} + \ldots$$

Fig. 8. Simple lead time model for finished products.

Figure 8 shows the lead time model as simple mathematical formula. This formula is the outcome of one 40 minute time-box. The formula in itself does not tell us much yet. We need quantifications to make the formula come alive. Participants often have to pass a threshold to write such formula. Uncertainty about the actual situation is often used as an excuse to keep the model implicit and qualitative. Some pressure from the facilitator is often required to pass this threshold. The lesson to be learned by the participants is that articulating the formula allows them to fill in numbers and to compare numbers and behavior with reality. Playing with the formula and (measured) numbers creates insight, and most often triggers many new questions. This playing with the formula is done in another time-box of 40 minutes.

The simplicity of the formula is somewhat misleading. Before this formula is reached many discussions take place, such as:

- What is the core decomposition to be used (e.g., processing steps in this case)?
- What is the granularity (also processing steps of minutes to many hours in this case)?
- Is there any concurrency (in this case the processing pipeline is entirely sequential)?

Lots of discussions pop up and questions are raised once numbers are substituted in the formula. Unfortunately, we are not allowed to show actual numbers for competitive reasons. It is our experience that the step from qualitative models to quantitative models helps tremendously to make discussions more sharp and concrete. Qualitative models do allow for nice, but non-committal discussions, while quantifications make escapes much more difficult.

Also for quantification the facilitator has to apply some pressure to get past a threshold. Uncertainty about actual performance often prevents designers to call actual numbers. However, it is crucial to quantify, since the quantification is so beneficial to the sharpness of the discussion. In this case the original quantification was more than a factor two off from actual performance. That is not too bad, as in other cases we also have seen original estimates that were an order of magnitude off.

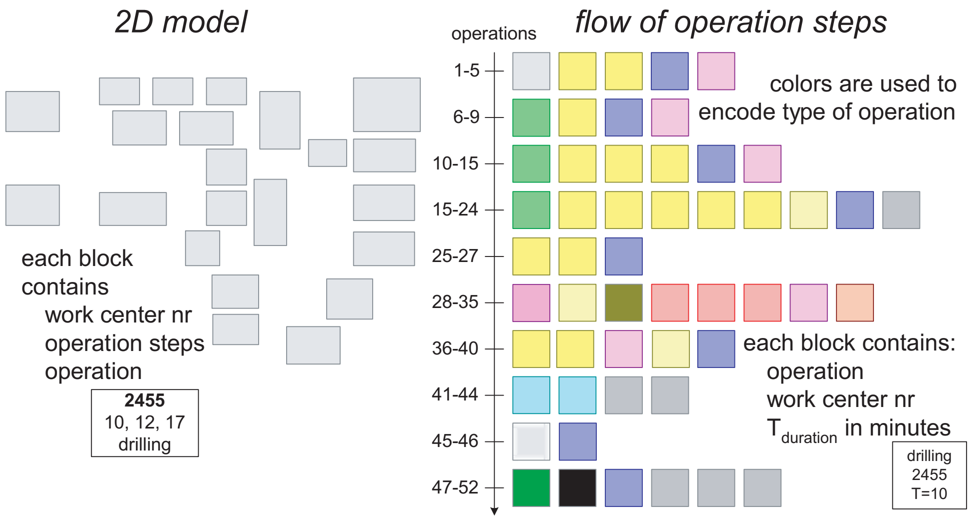

Fig. 9. Next iteration of 2D model and process flow with more detailed annotations of location, duration, and processing.

The following iteration, made as homework after the one-week course, resulted in the models shown in Fig. 9. Much more detail has been added about locations, the processing itself and the durations. These models are less abstract than the initial models in Fig. 7. The more elaborated model allows more detailed fact finding. The duration needed to get this type of drawing with meaningful data is in the order of one or a few days. In fact this elaboration is a validation of the earlier simpler models.

Throughout the course we also emphasize that models are made for a reason. We mentioned before that lead time and predictability of the lead time are key performance parameters. One of the techniques taught to identify models to be made is to find the tensions or conflicts in needs and realizations. In this case the introduction of the production of new components has tension with the lead time and especially with the predictability of the lead time. However, when we look at the lifecycle of the factory, then many factors pop up that impact the lead time and its predictability. Figure 10 shows a model of the lifecycle aspects that impact the lead time. Note that in this initial model qualitative color coding is used to visualize the impact. Such traffic light color coding is an intuitive way to get a quick insight in the most relevant factors. At the same time it helps to broaden the insight to identify all of the contributing lifecycle aspects.

Figure 11 shows in the different views what models have been made or what components and factors have been discussed during the course. It shows that in limited time lots of subjects are included or at least touched.

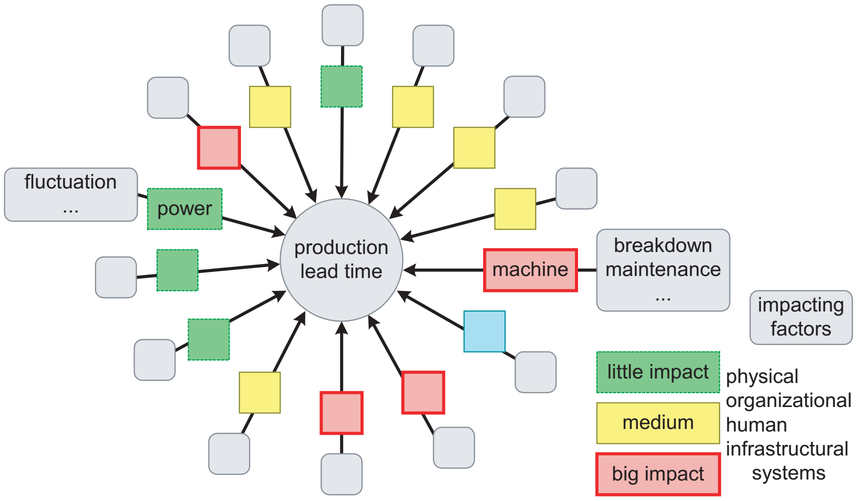

Fig. 10. Lifecycle model showing lifecycle systems and factors that may impact the production of lead time. The systems are shaded according to the severity of impact.

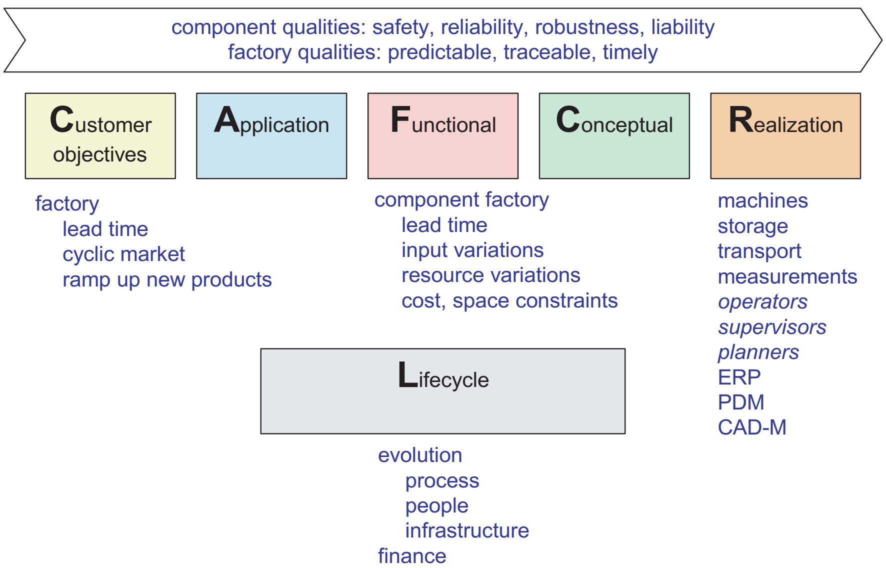

Fig. 11. The CAFCR+ model annotated with models or components of models that have been discussed during the course.

9. Future Work

In this paper we provided one case study and we focused on time-boxing and multi-view iteration. We are working on the consolidation of other case studies and we will use these to report on the other recommendations for modeling and analysis, such as the use of multiple views, visualization, etc.

In the long term the challenge is to move from observational research to a more theoretical foundation: what are modeling prerequisites, what are (well-founded) guidelines for modeling, and what is the value of modeling? Especially challenging is the validation, since many soft factors impact the modeling outcome. A separate branch of research has to address methodological issues in researching and validating this combination of hard engineering and soft sciences.

10. Summary and Conclusion

Figure 12 summarizes the modeling recommendations as discussed in the beginning of the paper. The main conclusion of this paper is that time-boxes and multi-view iteration help to gradually build a "big picture" in limited time. This big picture and its constituting models are incrementally improved by successive iterations. The duration of time-boxes can be as short as 5 minutes in early iterations and may increase to days for substantial models later during product creation. The main purpose of time-boxes is to ensure that the designers see other perspectives early, and to prevent that designers get lost in depth.

The prerequisite for this way of working is that a sufficient broad set of viewpoints is used to understand the system-of-interest. The models that are made use complementary representations: visualizations like maps, block diagrams and flow models, (simple) mathematical formulas, and numeric tables and graphs. Regular evaluation of relevance in terms of value and risk helps to determine where to zoom

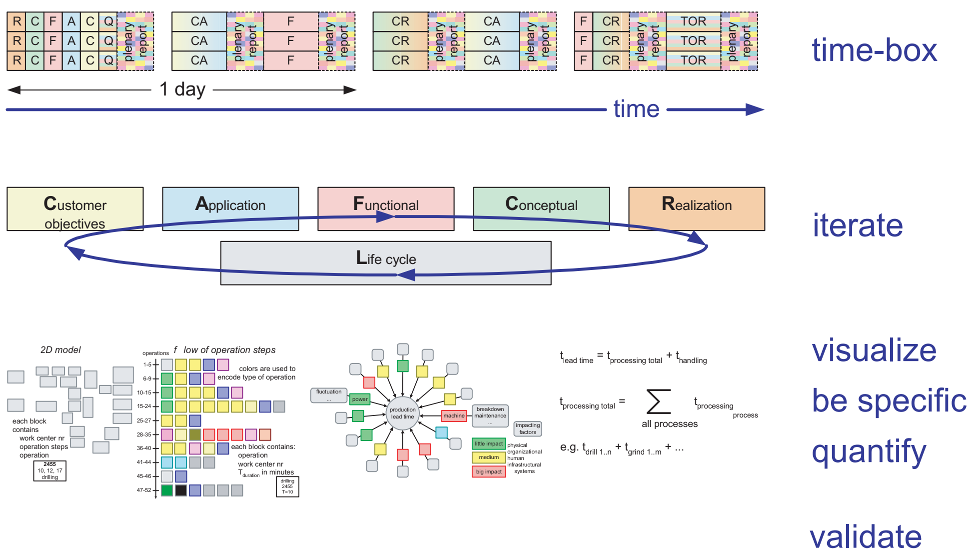

Fig. 12. Summary of the modeling recommendations.

in and where to spend more time. Last but not least, models need validation, thus we recommend early and frequent validation.

Acknowledgments

The participation of the students during the course and their project home work provided crucial input for this paper. Roelof Hamberg and Teun Hendriks participated during this course and became teachers themselves. They also served as discussion partners for this article and provided valuable input and feedback.

References

BUC Buskerud University College, Course: System Modeling and Analysis (SEMA 6201), 2008, http://www.hibu.no/sfiles/35/32/1/file/description_system_modeling_and_analysis_4.doc.

Cernosek, G. and Naiburg, E., The value of modeling, White paper (2004), ftp://ftp.software.ibm.com/software/rational/web/whitepapers/ValueOfModeling.pdf.

DoD, DoD Architecture Framework, Volume 1: Definitions and Guidelines, Version 1, US Dept. of Defence (2003).

Gilb, T., *Competitive Engineering* (Elsevier, 2005).

Hilliard, R., IEEE Std 1471 and Beyond, *SEI's First Architecture Representation Workshop* (2001).

Muller, G. J., CAFCR: A multi-view method for embedded systems architecting; balancing genericity and specificity, Ph.D. thesis, June 2004, http://www.gaudisite.nl/ThesisBook.pdf.

Muller, G. J., System modeling and analysis: A practical approach, http://www.gaudisite.nl/SystemModelingAndAnalysisBook.pdf.

OMG, Model driven architecture, accessed 17 August 2009, http://www.omg.org/mda/.

The Institute of Electrical and Electronics Engineers, Inc., IEEE recommended practice for architectural description of software-intensive systems, IEEE Std 1471, The Institute of Electrical and Electronics Engineers, Inc. (2000).

The MathWorks, About Model-Based Design (2009), http://www.mathworks.com/applications/controldesign/description/mbd.html.