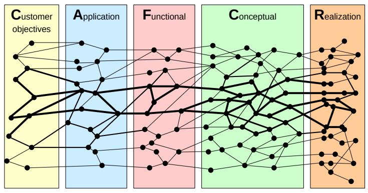
Threads of Reasoning

-



Gerrit Muller

University of South-Eastern Norway-NISE

Hasbergsvei 36 P.O. Box 235, NO-3603 Kongsberg Norway

gaudisite@gmail.com

Abstract

A method of reasoning is described, which addresses cross-cutting issues. The basis is fast iteration in the problem and solution space.
A thread of reasoning is a set of highly relevant related issues, which are addressed by articulating the problem in terms of tension and analyzing it in the CAFCR framework.

Distribution
This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

All Gaudí documents are available at:
http://www.gaudisite.nl/

version: 2.4 status: finished June 21, 2020

1 Introduction

The submethods provide generic means to cope with a limited part of the system architecture. The CAFCR model and the qualities provide a framework to position these results. The story telling is a means to do analysis and design work on the basis of concrete and specific facts. In this chapter a reasoning method is discussed to integrate all previous submethods. This reasoning method covers both the high level and the detailed views and covers the relation between multiple submethods and multiple qualities. The method is based on the identification of the points of tension in the problem and potential solutions.

The reasoning approach is explained as a 5 step approach. Section 2 provides an overview of the approach and gives a short introduction to each step. Section 3 describes the actual reasoning over multiple viewpoints: how to maintain focus and overview in such a multi-dimensional space? How to communicate and document? Section 4 explains how the threads of reasoning fit in the complete method.

2 Overview of Reasoning Approach

Fast exploration of the problem and solution space improves the quality of the specification and design decisions, as explained in Chapter **??**. It is essential to realize that such an exploration is highly concurrent, it is neither top-down, nor bottom-up, see viewpoint hopping and decision making in Sections **??** and **??**. In practice many designers find it difficult to make a start. In fact this does not have to be difficult: most starting points can be used, as long as the method is used with a sufficient open mind (that means that the starting point can be changed, when the team discovers that more important specification or design decisions are needed).

Figure 1 shows an overview of the entire reasoning approach. Step 1 is to *select a starting point*. After step 1 the iteration starts with step 2 *create insight*. Step 3 is *deepening the insight* and step 4 is *broadening the insight* with the questions. The next iteration is prepared by step 5 *refining or selecting the next need or problem*.

During this iteration continuous effort is required to *communicate with the stakeholders* to keep them up to date, to *consolidate in simple models* that are used during analysis and discussions and to *refactor the documentation* to keep it up to date with the insights obtained.

2.1 Selecting a Starting Point

As stated earlier it is more important to get started with the iteration than to spend a lot of time trying to find the most ideal starting point. A very useful starting point is to take a need or problem that is very hot at the moment. If this issue turns out to be important and critical then it needs to be addressed anyway. If it turns out to

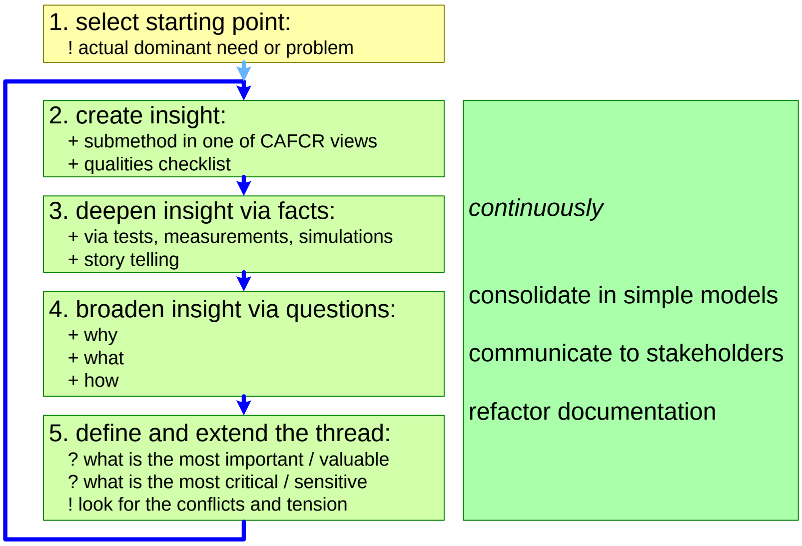


Figure 1: Overview of reasoning approach

be not that important, then the outcome of the first iteration serves to diminish the worries in the organization, enabling it to focus on the important issues.

In practice there are many hot issues that after some iterations turn out to be non-issues. This is often causes by non-rational fears, uncertainty, doubt, rumors, lack of facts et cetera. Going through the iteration, which includes fact finding, quickly positions the issues. This is of great benefit to the organization as a whole.

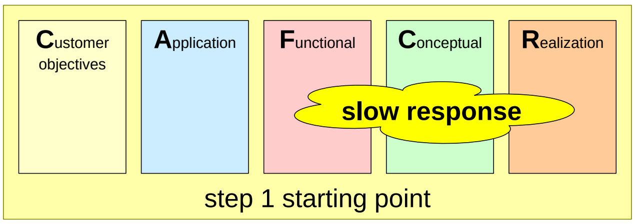


Figure 2: Example of a starting point: a slow system response discussed from the designer's viewpoint

The actual dominant needs or problems can be found by listening to what is mentioned with the greatest loudness, or which items dominate in all discussions and meetings. Figure 2 shows the response time as starting point for the iteration. This starting point was triggered by many design discussions about the cause of a slow system response and about potential concepts to solve this problem.

2.2 Building up Insight

The selected issue can be modeled by means of one of the many submethods as described in the CAFCR chapters. Doing this, it will quickly become clear what is known (and can be consolidated and communicated) and what is unknown, and what needs more study and is hence input for the next step. Figure 3 shows the *response time model* as potential submethod.

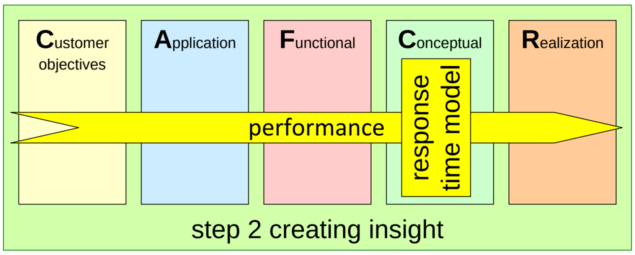


Figure 3: Example of creating insight: to study the required performance a response model of the system is made

An alternative approach is to look at the issue from the perspective of quality. One then has to identify the most relevant qualities, by means of the checklist in Figure **??**. These qualities can be used to sharpen the problem statement. Figure 3 shows the *performance* as quality to be used to understand the response time issue.

2.3 Deepening the Insight

The insight is deepened by gathering specific facts. This can be done by simulations, or by tests and measurements on existing systems. At the customer side story telling helps to get the needs sufficiently specific, as illustrated by Figure 4.

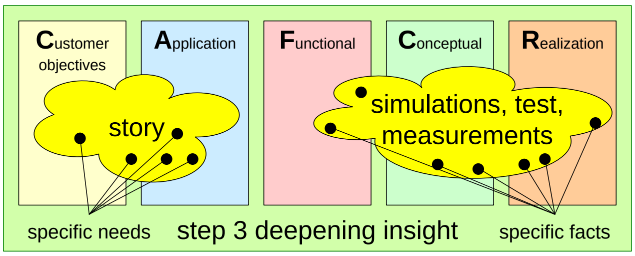


Figure 4: Deepening the insight by articulating specific needs and gathering specific facts by simulations, tests and simulations

It is important in this phase to *sample* specific facts and not to try to be complete. A very small subset of specific facts can already provide lots of insight. The speed of iteration is much more important than the completeness of the facts. Be aware that the iteration will quickly zoom in on the core design problems, which will result in sufficient coverage of the issues anyway.

2.4 Broadening the Insight

Needs and problems are never nicely isolated from the context. In many cases the reason why something is called a problem is because of the interaction between the function and the context. The insight is broadened by relating the need or problem to the other views in the CAFCR model. This can be achieved by the *why*, *what* and *how* questions as described in Section **??** and shown in Figure 5.

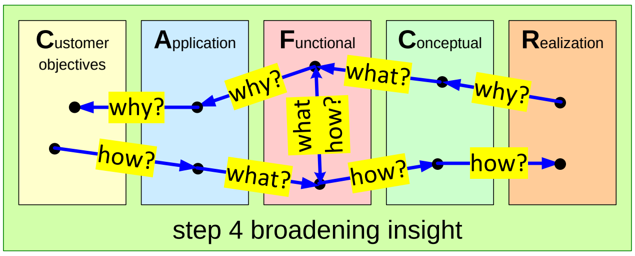


Figure 5: Broadening the insight by repeating why, what and how questions

The insight in the quality dimension can also be broadened by looking at the interaction with related qualities: what happens with safety, when we increase the performance?

2.5 Define and Extend the Thread

During the study and discussion of the needs and problems many new questions and problems pop up. A single problem can trigger an avalanche of new problems. Key in the approach is not to drown in this infinite ocean full of issues, by maintaining focus on important and critical issues. The most progress can be made by identifying the specification and design decisions that seem to be the most conflicting, i.e. where the most tension exists between the issues.

The relevance of a problem is determined by the *value* or the *importance* of the problem for the customer. The relevance is also determined by how challenging a problem is to solve. Problems that can be solved in a trivial way should immediately be solved. The approach as described is useful for problems that require some critical technical implementation. The implementation can be critical because it is difficult to realize, or because the design is rather sensitive[1] or rather vulnerable

[1] for instance in MRI systems the radius of the gradient coil system and the cost price were related with $(r_{magnet} - r_{gradientcoil})^5$. 1 cm more patient space would increase the cost dramatically, while at the same time patient space is crucial because of claustrophobia.

(for example, hard real-time systems with processor loads up to 70%).

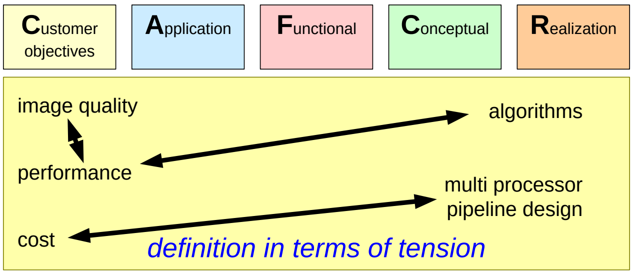


Figure 6: Example definition of the thread in terms of tension for a digital TV

Figure 6 shows the next crucial element to define the thread: identification the tension between needs and implementation options. The problem can be formulated in terms of this tension. A clearly articulated problem is half of the solution.

The example in Figure 6 shows the tension between the customer objectives and the design options. The image quality objective requires good algorithms that require a lot of processing power. Insufficient processing power lowers the system performance. The processing power is achieved by a pipeline of multiple processors. The cost of the number crunching capacity easily exceeds the cost target.

3 Reasoning

The reasoning by the architect is based on a combination of subjective intuition and objective analysis. The intuition is used to determine the direction and to evaluate results. The analysis partially validates the direction and partially helps the architect to develop his intuition further.

The assessment of the solutions is done by means of criteria. An objective ranking of the solutions can be made based on these criteria. The architect (and the other stakeholders) have their own subjective ranking based on intuition. By comparing the objective and subjective rankings a better understanding is achieved of both problem and solutions. This is shown in Figure 7. The increased understanding of the problem is used to improve the criteria. The increased understanding of the problem and the solutions influences the intuition of the architect (for instance this type of function is more expensive than expected). The increased understanding of the solution will trigger new solution(s).

During the reasoning a network of related issues emerges, as shown in Figure 8. Figure 8 visualizes the network as a graph, where a dot represents a specification or a design decision and a line represents a relation. Such a relation can be: *is implemented by*, *is detailed by*, *is conflicting with*, *enables or supports* et cetera.

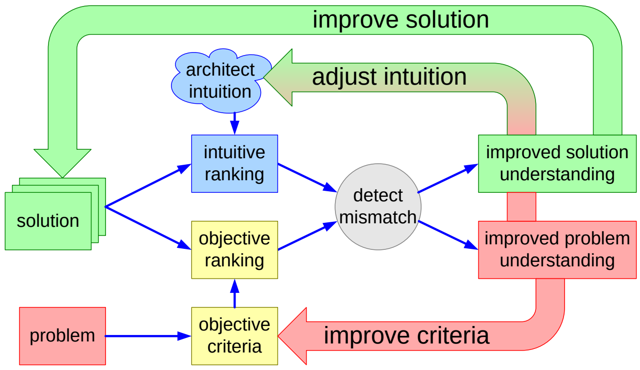


Figure 7: Reasoning as a feedback loop that combines intuition and analysis

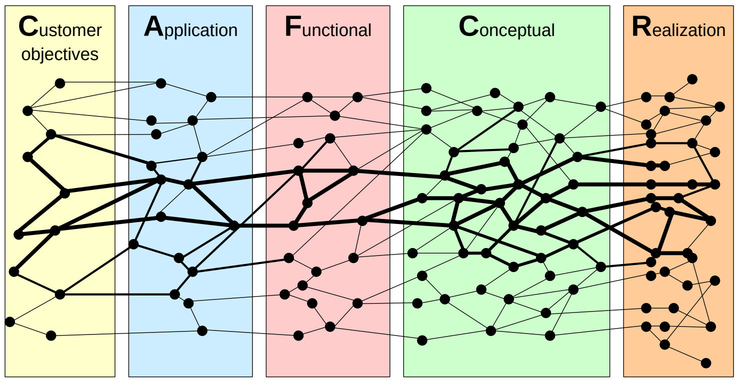


Figure 8: One *thread of reasoning* showing related issues. The line thickness is an indication for the weight of the relation.

The thickness of the line indicates the weight of the relation (thin is weak, thick is strong).

This graph is a visualization of the thread of reasoning followed by an architect. Crucial in such a thread is that it is sufficiently limited to maintain overview and to enable discussion and reasoning. A good thread of reasoning addresses relevant problem(s), without drowning in the real world complexity.

A continuous concern is to communicate with the stakeholders and to consolidate the findings, for instance in documentation. Figure 9 shows the more structured way to document and communicate these findings. The architect needs several iterations to recognize the structure in the seeming chaotic thread of reasoning. This example discusses the thread that has been shown in Figure 6. This single thread of reasoning addresses three key drivers as shown in Figure 9: *IQ (Image Quality)*, *cost* and *performance*. Most information in the thread of reasoning addresses

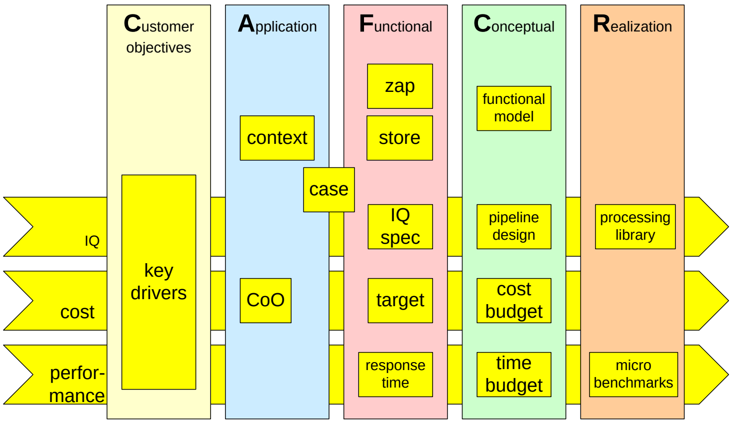


Figure 9: Example of the documentation and communication for a digital TV. The thread is documented in a structured way, despite the chaotic creation path. This structure emerges after several iterations.

these key drivers, however some additional information emerges too, such as the *context* of the digital TV at home, the functionality of the *zap* and *store* functions and the internal *functional models*.

4 Outline of the complete method

The *threads of reasoning* are the integration means of the overall method. In this section a short description is given how the *threads of reasoning* are are combined with the *submethods*, *quality checklists* and *story telling* to form a complete method. The steps in the description refer to Figure 7. Note that this aspect is speculative, because it has not been applied and therefore cannot be evaluated at this moment. Only an outline can be given now. A more detailed description of the method has to wait until further research is due.

The starting point (step 1) of a product creation is often a limited product specification, belonging in the *Functional view*. The next step is to explore (step 2) the customer context (*Customer Objectives* and *Application* views) and to explore the technical merits (*Conceptual* and *Realization* views). This exploration is used to identify a first set of customer-side opportunities and to identify the biggest technical challenges. During the exploration the *submethods* and *quality checklists* are used as a source of inspiration, for instance to determine the opportunity in the business model of the customer. Next (step 3) a *story* must be created that addresses the most important and valuable opportunities and the biggest technical challenges. The *story* is used to derive a first *use case* and to do a more thorough exploration (step 4) of the specification and the design. At this moment the first thread of reasoning is already visible (step 5), connecting a coarse product specification with customer opportunities and technical challenges. From this moment onwards the steps are repeated over and over, extending the thread of reasoning and creating one or two more threads of reasoning if needed. The submethods and the qualities are used during these iterations as a toolbox to describe specific parts of this creation process.

5 Summary

The reasoning approach is a means to integrate the CAFCR views and the qualities to design a system that fits entirely in the customer needs. The *threads of reasoning* approach is described by five steps. The result can be visualized as a graph of many related customer needs, specification issues and design issues. In this graph the core reasoning can be indicated around a limited set of key drivers or quality needs. In Chapter **??** the graph will be visualized for the Medical Imaging Workstation case.

References

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

History

Version: 2.5, date: April 6, 2004 changed by: Gerrit Muller

- replaced minus in intuition figure by descriptive text.
- corrected reference to section questions
- added references to basic working methods
- reformulated text about intuition and analysis
- small textual changes
- changed status to finished

Version: 2.4, date: March 16, 2004 changed by: Gerrit Muller

- added section "Outline of the complete method"
- small textual changes
- changed status to finished

Version: 2.3, date: March 11, 2004 changed by: Gerrit Muller

- added section "Summary"

Version: 2.2, date: February 27, 2004 changed by: Gerrit Muller

- small textual improvement

Version: 2.1, date: October 28, 2003 changed by: Gerrit Muller

- fine tuning
- added sentence about *tension* to the introduction
- used the same example in the figure for tension and documentation
- changed the figure of thread definition and extension into a figure about the tension in a thread.
- changed status to concept

Version: 2.0, date: October 28, 2003 changed by: Gerrit Muller

- fine tuning
- changed status to draft
- removed iteration figure
- split the figures woth step 1, 2, 3, and 4.
- rephrased step 5 into define and extend the thread.

Version: 1.1, date: September, 2003 changed by: Gerrit Muller

- fine tuning

Version: 1.0, date: May 26 2003 changed by: Gerrit Muller

- complete rewrite
- changed status to preliminary draft

Version: 0.3, date: May 21 2003 changed by: Gerrit Muller

- moved "problem solving approach" to chapter "Basic working methods of an architect"

Version: 0.2, date: April 28 2003 changed by: Gerrit Muller

- added text about problem solving approach
- added text about reasoning
- changed the order of the sheets

Version: 0.1, date: April 25 2003 changed by: Gerrit Muller

- added diagram exploration, problem identification and decision flow

Version: 0, date: April 3 2002 changed by: Gerrit Muller

- Created, no changelog yet