Abstract

Multi-view architecting connects the system design to customer context and life cycle context. We teach an architecting method based on many views and fast iteration of the views. Visual models, functional models, and mathematical models in all views are the means to communicate about the system, to discuss specification and design choices, to reason about consequences, and to make decisions.

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.

March 11, 2015
status: preliminary draft
version: 1.3
Abstract

This module introduces Architectural Reasoning using Conceptual Modeling.
Abstract

The SEMA course System Modeling and Analysis is a 5 day course. Core of the course is Architectural Reasoning Using Conceptual Modeling. This course uses the CAFCR+ model with 6 views. Qualities connect all views. Threads-of-reasoning capture the architectural reasoning across views and qualities. Conceptual models visualize and capture the context, the system and its design. Quantification is a means to make problem and solution space tangible.
<table>
<thead>
<tr>
<th>Day</th>
<th>Course Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1</td>
<td>Introduction to Modeling</td>
</tr>
<tr>
<td></td>
<td>Exploring the Case</td>
</tr>
<tr>
<td>Day 2</td>
<td>Sample Customer Space</td>
</tr>
<tr>
<td></td>
<td>Functions and Parts</td>
</tr>
<tr>
<td>Day 3</td>
<td>Customer Space Analysis</td>
</tr>
<tr>
<td></td>
<td>Quantification and Concepts</td>
</tr>
<tr>
<td>Day 4</td>
<td>Business and Life Cycle</td>
</tr>
<tr>
<td></td>
<td>Integration and Reasoning</td>
</tr>
<tr>
<td>Day 5</td>
<td>Modeling</td>
</tr>
<tr>
<td></td>
<td>Wrap-up</td>
</tr>
</tbody>
</table>
During the SEMA course you work in teams of about 3 persons. Smaller teams (even single persons) are acceptable as well.

Every team preferably works on a real part of a system with some real development that goes on.

We start to model the status quo of the system and then we will model and analyze a change or addition that is being considered.

As preparation for the course I ask you the following:

- Look if the other participants are working on similar systems, such that you can work as team.
- Pick as team a system/component/function/project you will use during the course.
- For this system/component/function/project collect information about: who is the customer, what does the customer need, how is the system used, what technologies are used in the system, what are the main technological challenges et cetera. You do not have to be an expert when you come to the course, but you need to have some feeling for the system you will be working on during the course and presumably also in the 10 week project.
- If you are preparing your master project, then the master project case is probably a good option. This will boost your master project.
Assignments during the Course

<table>
<thead>
<tr>
<th>Customer objectives</th>
<th>Application</th>
<th>Functional</th>
<th>Conceptual</th>
<th>Realization</th>
<th>Life cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. elevator</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. exploring the case</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. story telling</td>
<td>4. use case</td>
<td></td>
<td></td>
<td>5. dynamic behavior</td>
<td></td>
</tr>
<tr>
<td>6. block diagram</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7. context and workflow</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8. customer key driver graph</td>
<td></td>
<td>9. budget based design</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10. concept selection</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11. business plan</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12. change analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13. line of reasoning</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14. thread of reasoning</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15. quantified chain of models</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16. credibility and accuracy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>core</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SEMA System Modeling and Analysis Course</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="http://www.gaudisite.nl/info/SEMAcourse.info.html">http://www.gaudisite.nl/info/SEMAcourse.info.html</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SEMA Basic Philosophy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="http://www.gaudisite.nl/info/SEMAbasics.info.html">http://www.gaudisite.nl/info/SEMAbasics.info.html</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Physical Models of an Elevator</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="http://www.gaudisite.nl/info/ElevatorPhysicalModel.info.html">http://www.gaudisite.nl/info/ElevatorPhysicalModel.info.html</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>optional</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Teaching conceptual modeling at multiple system levels using multiple views</td>
</tr>
<tr>
<td>Understanding the human factor by making understandable visualizations</td>
</tr>
<tr>
<td><a href="http://www.gaudisite.nl/info/UnderstandingHumanFactorVisualizations.info.html">http://www.gaudisite.nl/info/UnderstandingHumanFactorVisualizations.info.html</a></td>
</tr>
<tr>
<td>Dynamic Range of Abstraction Levels in Architecting</td>
</tr>
<tr>
<td><a href="http://www.gaudisite.nl/info/DynamicRangeAbstractionLevels.info.html">http://www.gaudisite.nl/info/DynamicRangeAbstractionLevels.info.html</a></td>
</tr>
</tbody>
</table>
Course Material CAFCR Scan

**core**

SEMA Method Overview
http://www.gaudisite.nl/info/SEMAmethodOverviewSlides.pdf
Short introduction to basic "CAFCR" model
http://www.gaudisite.nl/info/BasicCAFCR.info.html
InitialCAFCRscan
http://www.gaudisite.nl/info/InitialCAFCRscan.info.html

**optional**

Architectural Reasoning Explained
Architectural Reasoning
http://www.gaudisite.nl/ArchitecturalReasoning.html
Iteration How To
http://www.gaudisite.nl/info/IterationHowTo.info.html
Modeling and Analysis: Iteration and Time-boxing
http://www.gaudisite.nl/info/MAiterationAndTimeboxing.info.html
<table>
<thead>
<tr>
<th><strong>core</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Story How To</td>
</tr>
<tr>
<td><a href="http://www.gaudisite.nl/info/StoryHowTo.info.html">http://www.gaudisite.nl/info/StoryHowTo.info.html</a></td>
</tr>
<tr>
<td>Use Case How To</td>
</tr>
<tr>
<td><a href="http://www.gaudisite.nl/info/UseCases.info.html">http://www.gaudisite.nl/info/UseCases.info.html</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>optional</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Story Telling in Medical Imaging</td>
</tr>
<tr>
<td><a href="http://www.gaudisite.nl/info/MIstories.info.html">http://www.gaudisite.nl/info/MIstories.info.html</a></td>
</tr>
<tr>
<td><strong>core</strong></td>
</tr>
<tr>
<td>----------------</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>optional</strong></th>
<th>Basic Working Methods of a System Architect</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SubSea Modeling Example</td>
</tr>
</tbody>
</table>
**core**
Methods to Explore the Customer Perspective
http://www.gaudisite.nl/info/MethodsToExploreTheCustomerPerspective.info.html

Key Drivers How To
http://www.gaudisite.nl/info/KeyDriversHowTo.info.html

**optional**
Medical Imaging Workstation: CAF Views
http://www.gaudisite.nl/info/MIviewsCAF.info.html
<table>
<thead>
<tr>
<th><strong>core</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Modeling and Analysis: Budgeting</td>
</tr>
<tr>
<td><a href="http://www.gaudisite.nl/info/MAbudgeting.info.html">http://www.gaudisite.nl/info/MAbudgeting.info.html</a></td>
</tr>
<tr>
<td>Concept Selection, Set Based Design and Late Decision Making</td>
</tr>
<tr>
<td><a href="http://www.gaudisite.nl/info/ConceptSelectionSetBased.info.html">http://www.gaudisite.nl/info/ConceptSelectionSetBased.info.html</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>optional</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>The Tool Box of the System Architect</td>
</tr>
<tr>
<td>core</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>Simplistic Financial Computations for System Architects.</td>
</tr>
<tr>
<td><a href="http://www.gaudisite.nl/info/SimplisticFinancialComputations.info.html">http://www.gaudisite.nl/info/SimplisticFinancialComputations.info.html</a></td>
</tr>
<tr>
<td>Modeling and Analysis: Life Cycle Models</td>
</tr>
<tr>
<td><a href="http://www.gaudisite.nl/info/MAlifeCycle.info.html">http://www.gaudisite.nl/info/MAlifeCycle.info.html</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>How to present architecture issues to higher management</td>
</tr>
<tr>
<td><a href="http://www.gaudisite.nl/info/ArchitectManagementInteraction.info.html">http://www.gaudisite.nl/info/ArchitectManagementInteraction.info.html</a></td>
</tr>
</tbody>
</table>
core

Qualities as Integrating Needles
http://www.gaudisite.nl/info/QualityNeedles.info.html

Threads of Reasoning
http://www.gaudisite.nl/info/ThreadsOfReasoning.info.html

Threads of reasoning illustrated by medical imaging case
http://www.gaudisite.nl/PresentationMITORSslides.pdf
## Course Material Modeling

<table>
<thead>
<tr>
<th><strong>core</strong></th>
<th></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>optional</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ASP Python Exercise</td>
<td><a href="http://www.gaudisite.nl/info/ASPpythonExercise.info.html">http://www.gaudisite.nl/info/ASPpythonExercise.info.html</a></td>
</tr>
</tbody>
</table>
Course Material Wrap-up

core

Consolidating Architecture Overviews
SEMA Homework Assignment
http://www.gaudisite.nl/info/SEMAhomeworkAssignmentSlides.pdf

optional

Guidelines for Visualization
http://www.gaudisite.nl/info/VisualizationGuidelines.info.html
Granularity of Documentation
http://www.gaudisite.nl/info/DocumentationGranularity.info.html
Light Weight Review Process
http://www.gaudisite.nl/info/LightWeightReview.info.html
Cookbook A3 Architecture Overview by Daniel Borches
http://www.gaudisite.nl/BorrentesCookbookA3architectureOverview.pdf
How to Create an Architecture Overview
http://www.gaudisite.nl/info/OverviewHowTo.info.html
Abstract

This presentation explains the basic philosophy behind the SEMA course. The SEMA course in the first place is a course that provides an approach to architectural reasoning. Core to architectural reasoning is the ability to make conceptual models and to use them in conjunction. The course discusses how to make conceptual models, how to get input, and how to use them for analysis. Modeling is put in broader perspective, such as model evolution, simulation, and validation.
You will mostly be working!

One Case during the course and the home work assignment
Work in teams if possible
Select a case close to your day-to-day practice

Learning by Doing
Some theory, apply on case

Case = System of interest + developing organization + some innovative change

Choice of case is critical!
Our Primary Interest

developing organization

architect

system of interest
Context, Zoom-out and Zoom-in

- customer organization
- developing organization
- architect
- supplier organization

- super system
- system of interest
- subsystems
Adding the Time Dimension

past  current  future

customer organization

past super system  super system  future super system

developing organization

past system of interest  system of interest  future system of interest

architect

knowledge  innovation

supplier organization

past subsystems  subsystems  future subsystems

based on TRIZ
Challenges

past   current   future

customer organization
past system of interest
heterogeneity
size & complexity
unknowns

architect
system of interest
ambiguity

organization
past subsystems
knowledge
innovation

architect
future system
future of interest
uncertainties

architect
future subsystems
based on TRIZ

legacy constraints

version: 0.2
March 11, 2015
SEMABchallenges
From Theory to Practice

Theory: typical SE workflow: V-model, requirements management, “top-down”

- requirements
  specification as input to the design, documented
  **SMART**
  Specific, Measurable, Acceptable, Realistic, Traceable

- requirements engineering
  the flow down of the requirements through the V.

Practice: Finite knowledge and wisdom causes late disruptions

- size & complexity
- heterogeneity
- ambiguity
- unknowns
- uncertainties
- legacy constraints
- innovation and new territory require *learning, e.g. experimenting, exploring, failing, discovering*
  complement with “bottom-up”

**Verification** of result against specification

specification

- system design
- subsystem design
- component design
- component realization

- component test
- subsystem test
- system test
- verification
- validation
Recommendations as Red Thread

**principles**
- use feedback
- work incremental
- work evolutionary
- be explicit
- make issues tangible

**objectives**
- support communication
- facilitate reasoning
- support decision making
- create understanding
- maintain insight
- overview

**recommendations**
- Time-box
- Iterate
- Quantify early
- Measure and validate
- Multiple levels of abstraction
- (Simple) mathematical models
- Analysis of accuracy and credibility
- Multi-view
- System and its context
- Visualize

help to achieve

translate into

translate into
Final Delivery: Presentation to Top Management

- Societal trends
- Competition trends
- Customers/stakeholders
- Business/market needs
- Product/project needs
- Design and concepts
- Specific aspects
- Technology
- Summary and conclusions
- Why choices are appropriate

- Business quantification
- Risk analysis
- Conclusions and recommendations
Case Selection

Determine the system of interest

Define your organization

Determine an innovative change to be architected
Physical Models of an Elevator

by Gerrit Muller    HBV-NISE

e-mail: gaudisite@gmail.com

www.gaudisite.nl

Abstract

An elevator is used as a simple system to model a few physical aspects. We will show simple kinematic models and we will consider energy consumption. These low level models are used to understand (physical) design considerations. Elsewhere we discuss higher level models, such as use cases and throughput, which complement these low level models.

---

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.

March 11, 2015
status: preliminary
draft
version: 0.4
Learning Goals

To understand the need for

- various views, e.g. physical, functional, performance
- mathematical models
- quantified understanding
- assumptions (when input data is unavailable yet) and later validation
- various visualizations, e.g. graphs
- understand and hence model at multiple levels of abstraction
- starting simple and expanding in detail, views, and solutions gradually, based on increased insight

To see the value and the limitations of these conceptual models

To appreciate the complementarity of conceptual models to other forms of modeling, e.g. problem specific models (e.g. structural or thermal analysis), SysML models, or simulations
This presentation starts with a trivial problem.

Have patience!

Extensions to the trivial problem are used to illustrate many different modeling aspects.

Feedback on correctness and validity is appreciated
inhabitants want to reach their destination fast and comfortable

building owner and service operator have economic constraints: space, cost, energy, ...
Elementary Kinematic Formulas

\[ S_t = \text{position at time } t \]

\[ v_t = \text{velocity at time } t \]

\[ a_t = \text{acceleration at time } t \]

\[ j_t = \text{jerk at time } t \]

\[ \frac{dS}{dt} = v \]

\[ \frac{dv}{dt} = a \]

\[ \frac{da}{dt} = j \]

Position in case of uniform acceleration:

\[ S_t = S_0 + v_0 t + \frac{1}{2} a_0 t^2 \]
Initial Expectations

What values do you expect or prefer for these quantities? Why?

- $t_{\text{top floor}} = \text{time to reach top floor}$
- $v_{\text{max}} = \text{maximum velocity}$
- $a_{\text{max}} = \text{maximum acceleration}$
- $j_{\text{max}} = \text{maximum jerk}$

This slide presents a diagram showing a building with an elevator shaft, indicating the distance to the top floor as 40 meters.
Google "elevator" and "jerk":

- $v_{\text{max}} \approx 2.5 \text{ m/s}$
- $a_{\text{max}} \approx 1.2 \text{ m/s}^2$ (up)
- $j_{\text{max}} \approx 2.5 \text{ m/s}^3$

12% of gravity; weight goes up

humans feel changes of forces
high jerk values are uncomfortable

t_{\text{top floor}} \approx 16 \text{ s}

relates to motor design and energy consumption
relates to control design

numbers from: http://www.sensor123.com/vm_eva625.htm
CEP Instruments Pte Ltd Singapore
Exercise Time to Reach Top Floor Kinematic

**input data**

S₀ = 0m  \[ S_t = 40m \]

v_max = 2.5 m/s

a_max = 1.2 m/s² (up)

j_max = 2.5 m/s³

**elementary formulas**

\[ v = \frac{dS}{dt} \quad a = \frac{dv}{dt} \quad j = \frac{da}{dt} \]

Position in case of uniform acceleration:

\[ S_t = S_0 + v_0 t + \frac{1}{2} a_0 t^2 \]

**exercises**

\[ t_{\text{top floor}} \text{ is time needed to reach top floor without stopping} \]

Make a model for \( t_{\text{top floor}} \) and calculate its value

Make 0ᵉ order model, based on constant velocity

Make 1ᵉ order model, based on constant acceleration

What do you conclude from these models?
Models for Time to Reach Top Floor

**Input Data**
- \( S_0 = 0m \)  \( S_{\text{top floor}} = 40m \)
- \( v_{\text{max}} = 2.5 \text{ m/s} \)
- \( a_{\text{max}} = 1.2 \text{ m/s}^2 \) (up)
- \( j_{\text{max}} = 2.5 \text{ m/s}^3 \)

**Elementary Formulas**
\[
\begin{align*}
  v &= \frac{dS}{dt} \\
  a &= \frac{dv}{dt} \\
  j &= \frac{da}{dt}
\end{align*}
\]

Position in case of uniform acceleration:
\[
S_t = S_0 + v_0 t + \frac{1}{2} a_0 t^2
\]

**0th Order Model**
- \( S_{\text{top floor}} = v_{\text{max}} \cdot t_{\text{top floor}} \)
- \( t_{\text{top floor}} = S_{\text{top floor}} / v_{\text{max}} \)
- \( t_{\text{top floor}} = 40/2.5 = 16 \text{s} \)

**1st Order Model**
- \( t_a \approx 2.5/1.2 \approx 2 \text{ s} \)
- \( S(t_a) \approx 0.5 \cdot 1.2 \cdot 2^2 \)
- \( S(t_a) \approx 2.4 \text{ m} \)
- \( t_v \approx (40-2*2.4)/2.5 \)
- \( t_v \approx 14 \text{s} \)
- \( t_{\text{top floor}} \approx 2 + 14 + 2 \)
- \( t_{\text{top floor}} \approx 18 \text{s} \)
Conclusions

$v_{\text{max}}$ dominates traveling time

The model for the large height traveling time can be simplified into:

$$t_{\text{travel}} = \frac{S_{\text{travel}}}{v_{\text{max}}} + (t_a + t_j)$$
Exercise Time to Travel One Floor

**input data**

- $S_0 = 0 \text{m}$
- $S_{\text{top floor}} = 40 \text{m}$
- $v_{\text{max}} = 2.5 \text{ m/s}$
- $a_{\text{max}} = 1.2 \text{ m/s}^2$ (up)
- $j_{\text{max}} = 2.5 \text{ m/s}^3$

**elementary formulas**

\[
\begin{align*}
    v &= \frac{dS}{dt} \\
    a &= \frac{dv}{dt} \\
    j &= \frac{da}{dt}
\end{align*}
\]

Position in case of uniform acceleration:

\[
S_t = S_0 + v_0 t + \frac{1}{2} a_0 t^2
\]

**exercise**

Make a model for one floor and calculate it

What do you conclude from this model?
2nd Order Model Moving One Floor

2nd order model

- $S_{one floor} = 3m$
- $v_{max} = 2.5 \text{ m/s}$
- $a_{max} = 1.2 \text{ m/s}^2$ (up)
- $j_{max} = 2.5 \text{ m/s}^3$

$t_{one floor} = 2 \ t_a + 4 \ t_j$

- $t_j = \frac{a_{max}}{j_{max}}$
- $S_1 = \frac{1}{6} \ j_{max} t_j^3$
- $v_1 = 0.5 \ j_{max} t_j^2$
- $S_2 = S_1 + v_1 t_a + 0.5 \ a_{max} t_a^2$
- $v_2 = v_1 + a_{max} t_a$
- $S_3 = S_2 + v_2 t_j + 0.5 \ a_{max} t_j^2 - \frac{1}{6} j_{max} t_j^3$

$S_3 = 0.5 \ S_t$

- $t_j \sim= 1.2/2.5 \sim= 0.5s$
- $S_1 \sim= 1/6 \times 2.5 \times 0.5^3 \sim= 0.05m$
- $v_1 \sim= 0.5 \times 2.5 \times 0.5^2 \sim= 0.3\text{ m/s}$

et cetera
1st Order Model Moving One Floor

**1st order model**

- **Input data**
  - $S_{\text{one floor}} = 3\, \text{m}$
  - $v_{\text{max}} = 2.5\, \text{m/s}$
  - $a_{\text{max}} = 1.2\, \text{m/s}^2$ (up)
  - $j_{\text{max}} = 2.5\, \text{m/s}^3$

- **Equations**
  - $S(t_a) = \frac{1}{2} \cdot a_{\text{max}} \cdot t_a^2$
  - $t_a = \sqrt{\frac{S(t_a)}{0.5 \cdot a_{\text{max}}}}$
  - $t_{\text{one floor}} = 2 \cdot t_a = 2 \sqrt{\frac{S(t_a)}{0.5 \cdot a_{\text{max}}}}$
  - $v(t_a) = a_{\text{max}} t_a$
  - $v(t_a) \approx 1.2 \cdot 1.6 \approx 1.9\, \text{m/s}$

- **Coarse 2nd order correction**
  - $t_{\text{one floor}} \approx 2 \cdot t_a + 2 \cdot t_j$
  - $t_j \approx 0.5\, \text{s}$
  - $t_{\text{one floor}} \approx 2 \cdot 1.6 + 2 \cdot 0.5 \approx 4\, \text{s}$
Conclusions

$a_{\text{max}}$ dominates travel time

The model for small height traveling time can be simplified into:

$$t_{\text{travel}} = 2 \sqrt{S_{\text{travel}}/0.5 \ a_{\text{max}}} + t_j$$
exercise

Make a model for $t_{\text{top floor}}$
Take door opening and docking into account
What do you conclude from this model?
Elevator Performance Model

**functional model**
- close doors
- undock elevator
- move elevator
- dock elevator
- open doors elevator

**performance model**
\[ t_{\text{top floor}} = t_{\text{close}} + t_{\text{undock}} + t_{\text{move}} + t_{\text{dock}} + t_{\text{open}} \]

**assumptions**
- \( t_{\text{close}} \sim t_{\text{open}} \sim 2 \text{s} \)
- \( t_{\text{undock}} \sim 1 \text{s} \)
- \( t_{\text{dock}} \sim 2 \text{s} \)
- \( t_{\text{move}} \sim 18 \text{s} \)

**outcome**
- \( t_{\text{top floor}} \sim 2 + 1 + 18 + 2 + 2 \)
- \( t_{\text{top floor}} \sim 25 \text{s} \)
Conclusions

The time to move is dominating the traveling time.

Docking and door handling is significant part of the traveling time.

\[ t_{\text{top floor}} = t_{\text{travel}} + t_{\text{elevator overhead}} \]
Measured Elevator Acceleration

The graph shows the measured elevator acceleration with time. The acceleration is measured in m/s² and the time is in seconds. The graph is reproduced from http://www.sensor123.com/vm_eva625.htm. CEP Instruments Pte Ltd Singapore.
What did we ignore or forget?

acceleration: up <> down 1.2 m/s² vs 1.0 m/s²

slack, elasticity, damping et cetera of cables, motors....

controller impact

.....
exercise

Make a model for $t_{\text{one floor}}$
Take door opening and docking into account
What do you conclude from this model?
Elevator Performance Model

**functional model**
- close doors
  - undock elevator
  - move elevator
  - dock elevator
  - open doors elevator

**performance model one floor (3m)**

\[
t_{\text{one floor}} = t_{\text{close}} + t_{\text{undock}} + t_{\text{move}} + t_{\text{dock}} + t_{\text{open}}
\]

**assumptions**
- \( t_{\text{close}} \sim t_{\text{open}} \sim 2\text{s} \)
- \( t_{\text{undock}} \sim 1\text{s} \)
- \( t_{\text{dock}} \sim 2\text{s} \)
- \( t_{\text{move}} \sim 4\text{s} \)

**outcome**
- \( t_{\text{one floor}} \sim 2 + 1 + 4 + 2 + 2 \)
- \( t_{\text{one floor}} \sim 11\text{s} \)
Conclusions

Overhead of docking and opening and closing doors is dominating traveling time.

Fast docking and fast door handling has significant impact on traveling time.

\[ t_{\text{one floor}} = t_{\text{travel}} + t_{\text{elevator overhead}} \]
Exercise Time Line

**Exercise**

Make a time line of people using the elevator. Estimate the time needed to travel to the top floor. Estimate the time needed to travel one floor. What do you conclude?

```
start

action 1

action 2

action 3

ready

time
```
**Time Line; Humans Using the Elevator**

- **Dock open doors**
- **Wait for leaving people**
- **Walk in select floor**
- **Wait for elevator**
- **Wait for leaving people**
- **Walk out**

### Assumptions: Human Dependent Data

- $t_{\text{wait for elevator}} = [0..2 \text{ minutes}]$ depends heavily on use
- $t_{\text{wait for leaving people}} = [0..20 \text{ seconds}]$ idem
- $t_{\text{walk in}} \approx t_{\text{walk out}} \approx 2 \text{ s}$
- $t_{\text{select floor}} \approx 2 \text{ s}$

### Outcome

- $t_{\text{one floor}} = t_{\text{minimal waiting time}} + t_{\text{walk out}} + t_{\text{travel one floor}} + t_{\text{wait}}$
- $t_{\text{top floor}} = t_{\text{minimal waiting time}} + t_{\text{walk out}} + t_{\text{travel top floor}} + t_{\text{wait}}$
- $t_{\text{one floor}} \approx 8 + 2 + 11 + t_{\text{wait}} \approx 21 \text{ s} + t_{\text{wait}}$
- $t_{\text{top floor}} \approx 8 + 2 + 25 + t_{\text{wait}} \approx 35 \text{ s} + t_{\text{wait}}$

### Assumptions: Additional Elevator Data

- $t_{\text{minimal waiting time}} \approx 8 \text{ s}$
- $t_{\text{travel top floor}} \approx 25 \text{ s}$
- $t_{\text{travel one floor}} \approx 11 \text{ s}$
Overview of Results for One Elevator

**top floor**

- **0th order time to move elevator 40m**: 16s
- **1st order correction elevator docking and doors**: 2s
- **7s**
- **1st order correction**: 3s
- **Human related waiting time**: 10s

Total: \(35s + t_{\text{wait}}\)

**one floor**

- **0th order time to move elevator 40m**: 16s
- **1st order correction elevator docking and doors**: 2s
- **7s**
- **1st order model**: 3s
- **Human related waiting time**: 10s

Total: \(21s + t_{\text{wait}}\)
Conclusions

The human related activities have significant impact on the end-to-end time.

The waiting times have significant impact on the end-to-end time and may vary quite a lot.

$$t_{\text{end-to-end}} = t_{\text{human activities}} + t_{\text{wait}} + t_{\text{elevator travel}}$$
**Exercise**

Estimate the energy consumption and the average and peak power needed to travel to the top floor.

What do you conclude?
Energy and Power Model

**input data**

- $S_0 = 0 \text{m}$
- $S_t = 40 \text{m}$
- $v_{\text{max}} = 2.5 \text{ m/s}$
- $a_{\text{max}} = 1.2 \text{ m/s}^2 \text{ (up)}$
- $j_{\text{max}} = 2.5 \text{ m/s}^3$
- $g = 10 \text{ m/s}^2$
- $m_{\text{elevator}} = 1000 \text{ Kg (incl counter weight)}$
- $m_{\text{passenger}} = 100 \text{ Kg}$

**elementary formulas**

- $E_{\text{kin}} = \frac{1}{2} m v^2$
- $E_{\text{pot}} = mgh$
- $W = \frac{dE}{dt}$
- $E_{\text{kin max}} = \frac{1}{2} m v_{\text{max}}^2$
  \[ \approx 0.5 \times 1100 \times 2.5^2 \]
  \[ \approx 3.4 \text{ kJ} \]
- $W_{\text{kin max}} = m v_{\text{max}} a_{\text{max}}$
  \[ \approx 1100 \times 2.5 \times 1.2 \]
  \[ \approx 3.3 \text{ kW} \]
- $E_{\text{pot}} = mgh$
  \[ \approx 100 \times 10 \times 40 \]
  \[ \approx 40 \text{ kJ} \]
- $W_{\text{pot max}} \approx E_{\text{pot}}/t_v$
  \[ \approx 40/16 \]
  \[ \approx 2.5 \text{ kW} \]
## Conclusions

- $E_{\text{pot}}$ dominates energy balance
- $W_{\text{pot}}$ is dominated by $v_{\text{max}}$
- $W_{\text{kin}}$ causes peaks in power consumption and absorption
- $W_{\text{kin}}$ is dominated by $v_{\text{max}}$ and $a_{\text{max}}$

### Energy and Power Calculations

\[
E_{\text{kin max}} = \frac{1}{2} m v_{\text{max}}^2 \\
\approx 0.5 \times 1100 \times 2.5^2 \\
\approx 3.4 \text{ kJ}
\]

\[
W_{\text{kin max}} = m v_{\text{max}} a_{\text{max}} \\
\approx 1100 \times 2.5 \times 1.2 \\
\approx 3.3 \text{ kW}
\]

\[
E_{\text{pot}} = mgh \\
\approx 100 \times 10 \times 40 \\
\approx 40 \text{ kJ}
\]

\[
W_{\text{pot max}} \approx \frac{E_{\text{pot}}}{t_v} \\
\approx 40/16 \\
\approx 2.5 \text{ kW}
\]
Exercise Qualities and Design Considerations

**Exercise**

What other qualities and design considerations relate to the kinematic models?
Examples of other qualities and design considerations

- Safety: $v_{\text{max}}$
- Acoustic noise: $v_{\text{max}}, a_{\text{max}}, j_{\text{max}}$
- Mechanical vibrations: $v_{\text{max}}, a_{\text{max}}, j_{\text{max}}$
- Air flow: ?
- Operating life, maintenance: duty cycle, ?

Obstacles cause vibrations.
applicability in other domains

kinematic modeling can be applied in a wide range of domains:
transportation systems (trains, busses, cars, containers, ...)
waffer stepper stages
health care equipment patient handling
material handling (printers, inserters, ...)
MRI scanners gradient generation
...
Exercise

Assume that a group of people enters the elevator at the ground floor. On every floor one person leaves the elevator.

What is the end-to-end time for someone traveling to the top floor?

What is the desired end-to-end time?

What are potential solutions to achieve this?

What are the main parameters of the design space?
Multiple Users Model

**elevator data**
- $t_{\text{min wait}} \sim= 8$ s
- $t_{\text{one floor}} \sim= 11$ s
- $t_{\text{walk out}} \sim= 2$ s
- $n_{\text{floors}} = 40 \text{ div } 3 + 1 = 14$
- $n_{\text{stops}} = n_{\text{floors}} - 1 = 13$

**outcome**

\[
t_{\text{end-to-end}} = n_{\text{stops}} (t_{\text{min wait}} + t_{\text{one floor}}) + t_{\text{walk out}} + t_{\text{wait}} \\
\sim= 13 \times (8 + 11) + 2 + t_{\text{wait}} \\
\sim= 249 \text{ s} + t_{\text{wait}}
\]

\[
t_{\text{non-stop}} \sim= 35 \text{ s} + t_{\text{wait}}
\]
Considerations

desired time to travel to top floor $\sim< 1$ minute

note that $t_{\text{wait next}} = t_{\text{travel up}} + t_{\text{travel down}}$

if someone just misses the elevator then the waiting time is

$$t_{\text{end-to-end}} \sim= 249 + 35 + 249 = 533 \text{s} \sim= 9 \text{ minutes}!$$

desired waiting time $\sim< 1$ minute
Design of a system with multiple elevator requires a different kind of models: oriented towards logistics.
Exceptional Cases

- non-functioning elevator
- maintenance, cleaning of elevator
- elevator used by people moving household
- rush hour
- special events (e.g. party, new years eve)
- special floors (e.g. restaurant)
- many elderly or handicapped people
- playing children
Make a list of all *visualizations* and *representations* that we used during the exercises
Summary of Visualizations and Representations

**Physical Models of an Elevator**

65  Gerrit Muller

**version:** 0.4  
**March 11, 2015**  
**EPMsummary/Visualizations**

- **Mathematical Formulas**
  
  \[ s_t = s_0 + v_0 t + \frac{1}{2} a_0 t^2 \]
  
  \[ t_{\text{top floor}} = t_{\text{close}} + t_{\text{undock}} + t_{\text{move}} + t_{\text{dock}} + t_{\text{open}} \]

- **Schematic Graphs**

- **Measurement Graph**

- **Functional Model**
  - close doors
  - undock elevator
  - move elevator
dock elevator
  - open doors elevator

- **Functional Model Diagram**

- **Timeline, Concurrency**

- **Quantification**
  - \[ t_{\text{top floor}} \approx 25 \text{s} \]
  - \[ t_{\text{top floor}} = 21 \text{s} + t_{\text{wait}} \]

- **Other People Entering**
  - minimal waiting time
  - \[ 4 \text{s} \]

- **Human Related**
  - \[ 7 \text{s} \]

- **Elevator Docking and Doors**
  - \[ 11 \text{s} \]

- **Waiting Time**
  - \[ 21 \text{s} \]

- **1st Order Model**

- **2nd Order Correction**

**Notes:**

- Graph reproduced from: [http://www.sensor123.com/vm_eva625.htm](http://www.sensor123.com/vm_eva625.htm)

- CEP Instruments Pte Ltd Singapore
Architecting Scope and Challenges

Scope

- **customer organization**
  - past system
  - super system
  - future super system

- **developing organization**
  - past system of interest
  - system of interest
  - future system of interest

- **architect**
  - past subsystems
  - subsystems
  - future subsystems

- **supplier organization**
  - past subsystems (based on TRIZ)

Challenges

- **customer organization**
- **developing organization**
- **architect**
- **supplier organization**

- **heterogeneity**
- **ambiguity**
- **size & complexity**
- **unknowns**
- **uncertainties**
- **legacy constraints**

Recommendations

- **principles**
  - use feedback
  - work incremental
  - work evolutionary
  - be explicit
  - make issues tangible

- **objectives**
  - support communication
  - facilitate reasoning
  - support decision making
  - create maintain
  - understanding insight overview

- **recommendations**
  - Time-box
  - Iterate
  - Quantify early
  - Measure and validate
  - Multiple levels of abstraction
  - (Simple) mathematical models
  - Analysis of accuracy and credibility
  - Multi-view
  - System and its context
  - Visualize

Final Top-Down Delivery

- **summary how solution answers needs**
- **business quantification**
  - risk analysis
  - conclusions and recommendations
  - product project
    - system functions
    - key performance
  - design and concepts
    - functional, physical
    - quantified
  - specific aspects
    - functional, physical
    - quantified
  - technology
    - critical or new

Summary Module Architectural Reasoning Introduction

version: 0.4
March 11, 2015

Gerrit Muller
Introduction Conceptual Modeling

Zooming Out

Complementary Visualizations and Representations

intentionally left blank

intentionally left blank
Abstract

This module introduces the case exploration used in the course Architectural Reasoning using Conceptual Modeling.
Abstract

This presentation provides an overview of the SEMA course: Architectural Reasoning Using Conceptual Modeling. This course uses the CAFCR+ model with 6 views. Qualities connect all views. Threads-of-reasoning capture the architectural reasoning across views and qualities. Conceptual models visualize and capture the context, the system and its design. Quantification is a means to make problem and solution space tangible.
From vague notions to articulate and structured architecture description:

- Articulated
- Structured

Problem and solution know-how

Vague notion of the problem

Vague notion of potential solutions

Basic methods:
- Decision making
- Modeling and analysis
- Time-boxing and iteration

Architecting method:
- Framework
- Submethods
- Integration methods

Architecture description:
- Articulated
- Structured

Problem and solution

Know-how

Architecting
Overview of architecting method

**method outline**

<table>
<thead>
<tr>
<th>Customer objectives</th>
<th>Application</th>
<th>Functional</th>
<th>Conceptual</th>
<th>Realization</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>framework</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>+ key drivers</td>
<td>+ stakeholders</td>
<td>+ use case</td>
<td>+ construction</td>
</tr>
<tr>
<td></td>
<td>+ value chain</td>
<td>+ context diagram</td>
<td>+ commercial, logistics</td>
<td>decomposition</td>
</tr>
<tr>
<td></td>
<td>+ business models</td>
<td>+ entity relationship</td>
<td>+ decompositions</td>
<td>functional</td>
</tr>
<tr>
<td></td>
<td>+ supplier map</td>
<td>+ models</td>
<td>+ mapping technical</td>
<td>decomposition</td>
</tr>
<tr>
<td></td>
<td>+ dynamic models</td>
<td>+ functions</td>
<td>+ information model</td>
<td>and several more</td>
</tr>
</tbody>
</table>

**submethods**

+ use case
+ key drivers
+ value chain
+ business models
+ supplier map
+ stakeholders
+ context diagram
+ entity relationship
+ models
+ dynamic models
+ use case
+ commercial, logistics
+ decompositions
+ mapping technical
+ functions
+ and several more
+ construction
+ decomposition
+ functional
+ decomposition
+ information model
+ and many more
+ budget
+ benchmarking
+ performance
+ analysis
+ safety analysis
+ and many more

**integration**

via qualities

**explore**

specific details

reasoning

**method visualization**
Purpose of Modeling

- Facts from research
- Measurements
- Assumptions

Uncertainties, unknowns, errors

Modeling

Analysis

Results

Project

Specification
Verification
Decisions

Risk
Customer satisfaction
Time, cost, effort
Profit margin

Accuracy
Working range
Credibility
What to Model?

- business: profit, etc.
- operational costs
- stakeholder benefits
- workload
- risks

- key performance: throughput, response
- reliability
- availability
- scalability
- ...

- (emerging?) properties: resource utilization
- load
- latency, throughput
- quality, accuracy
- ...

- and their mutual relations

Diagram:

- usage context
  - enterprise & users
  - requirements
  - black box view

- system
  - design
  - realization
  - technology

- life cycle context
  - creation
  - life cycle business
Overview of Modeling Approach

- **facts from research**
  - measurements
  - assumptions

- **modeling**
  - analysis

- **results**

- **decision making**
  - decisions
    - project man.
    - business spec.
    - design

**collect input data**

**model and analyse relevant issues**

**for different stakeholders & concerns**

**integration and reasoning**
Short introduction to basic “CAFCR” model

by Gerrit Muller  Buskerud University College

e-mail: gaudisite@gmail.com

www.gaudisite.nl

Abstract

The basic “CAFCR” reference model is described, which is used to describe a system in relation to its context. The main stakeholder in the context is the customer. The question “Who is the customer?” is addressed.
The “CAFCR” model

**What** does Customer need in Product and **Why**?

- **Customer**
  - What
  - How
  - Objectives

- **Application**
  - Functional

- **Product**
  - Conceptual
  - Realization

- Drives, justifies, needs
- Enables, supports
Integrating CAFCR

**What** does Customer need in Product and **Why**?

**Customer**
- **What**: Customer objectives
- **How**: Application

**Product**
- **What**: Functional, Conceptual
- **How**: Realization

- Context: Understanding
- Intention: Objective-driven
- Opportunities: Knowledge-based
- Constraint: Awareness

Short introduction to basic “CAFCR” model

Gerrit Muller
CAFCR can be applied recursively

Value Chain
larger scope has smaller
influence on architecture

System
(producer)

Consumer

Drives

Enables

Customer's
Customer Business

Customer

Drives

Enables

Business

Drives

Enables

Consumer Drives
Enables

Value Chain
larger scope has smaller
influence on architecture

version: 0.4
March 11, 2015
CAFCRrecursion
## Market segmentation

<table>
<thead>
<tr>
<th>Segmentation axis</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>geographical</td>
<td>USA, UK, Germany, Japan, China</td>
</tr>
<tr>
<td>business model</td>
<td>profit, non profit</td>
</tr>
<tr>
<td>economics</td>
<td>high end versus cost constrained</td>
</tr>
<tr>
<td>consumers</td>
<td>youth, elderly</td>
</tr>
<tr>
<td>outlet</td>
<td>retailer, provider, OEM, consumer direct</td>
</tr>
</tbody>
</table>
Example of a small buying organization

Who is the customer?

- CFO: Chief Financial Officer
- CIO: Chief Information Officer
- CMO: Chief Marketing Officer
- CEO: Chief Executive Officer
- CTO: Chief Technology Officer
- purchaser
- decision maker(s)
- department head
- user
- maintainer
- operator

CEO: Chief Executive Officer
CFO: Chief Financial Officer
CIO: Chief Information Officer
CMO: Chief Marketing Officer
CTO: Chief Technology Officer
CAFCR+ model; Life Cycle View

Customer objectives

Application

Functional

Conceptual

Realization

Life cycle
operations
maintenance
upgrades

development
manufacturing
installation

sales, service, logistics, production, R&D

Short introduction to basic “CAFCR” model
81 Gerrit Muller

version: 0.4
March 11, 2015
BCAFCRplusLifeCycle
Abstract

This presentation guides a team through a quick CAFCR scan. Such quick scan with typically 15 minutes per view helps to build an initial overview of the problem and solution space.
make a bottom-up analysis of your product:

1. realization
2. conceptual
3. functional
4. application
5. customer objectives
6. qualities

use time boxes of 15 minutes per view

show the most dominant decomposition of that view, as diagram or as a list; some more guidance will be given per step.
<table>
<thead>
<tr>
<th>Do</th>
<th>Do not</th>
<th>Because</th>
</tr>
</thead>
<tbody>
<tr>
<td>start sketching/drawing as soon as possible</td>
<td>write long texts</td>
<td>sketches stimulate sharing and discussion</td>
</tr>
<tr>
<td>use shared large sheets of paper (e.g. flip-over)</td>
<td>immediately capture electronic</td>
<td>sharing and discussion help to explore faster</td>
</tr>
<tr>
<td>number the flip-overs and add a title</td>
<td>have nice but volatile discussions</td>
<td>remembering the order gets challenging</td>
</tr>
<tr>
<td>annotate (add notes) during discussions</td>
<td>write with pen or pencil</td>
<td>information and insight is quickly lost</td>
</tr>
<tr>
<td>use yellow note stickers and flip-over markers</td>
<td>Do not stick to the first solution</td>
<td>stickers are easily (re)moved</td>
</tr>
<tr>
<td>be open for ideas and surprises</td>
<td></td>
<td>you hopefully discover a lot; increased insight will change problem and solution</td>
</tr>
</tbody>
</table>
Step 1: Realization View

Choose 1 or 2 items from below

How

- GPS
- gyros
- DL2128V

HW block diagram

- CAN
- CAN master
- ARM CPU
- 256MB NAND Flash
- 8 12 bit DA

SW layer diagram

- applications
  - view
  - PIP
  - adjust
  - TXT

- services toolboxes
  - view
  - menu
  - audio
  - video
  - TXT
  - etc.

- driver
  - viewport
  - menu
  - drivers
  - scheduler

- hardware
  - tuning
  - frame-buffer
  - MPEG
  - DSP
  - CPU
  - RAM
  - etc.

2D layout of system internals

- primary engine
- transmission
- batteries
- fuel tank

3D sketch of system internals

Annotate/mark most critical technologies or characteristics
Step 2: Conceptual View

Chose 1 or 2 items from below

- **What**
  - sense position
  - sense orientation
  - sense speed
  - determine trajectory
  - determine setpoints
  - control rudders
  - control engine

- **functional model**

- **information model**

- **subsystem decomposition**

- **performance model**

---

Chose 1 or 2 items from below

- subsystem decomposition
- functional model
- information model
- performance model

---

**What**

- sense position
- sense orientation
- sense speed
- determine trajectory
- determine setpoints
- control rudders
- control engine

---

**functional model**

**information model**

**subsystem decomposition**

**performance model**

---

**math**

\[
 t_{\text{recon}} = t_{\text{filter}}(n_{\text{raw-x}}, n_{\text{raw-y}}) + t_{\text{FFT}}(n_{\text{raw-x}}, n_{\text{raw-y}}) + t_{\text{control-overhead}}
\]

\[
 t_{\text{FFT}} = c_{\text{FFT}} \cdot n \cdot \log(n)
\]
Step 3: Functional View; Top level Spec

system seen as black box
inputs
functions
quantified characteristics

restrictions, prerequisites
boundaries, exceptions
standards, regulations
outputs

interfaces
Step 4: Application View

Chose 1 or 2 items from below

- system context
  - stakeholders and concerns (who)
  - 2D map (where)

- work flow & time line (what, when)

Initial CAFCR scan

Gerrit Muller

version: 0.2
March 11, 2015
SEMAbottomUpScanApplication
Step 5: Customer Objectives View; Value Chain
Step 6: Qualities

**usable**
- usability
- attractiveness
- responsiveness
- image quality
- wearability
- storability
- transportability

**dependable**
- safety
- security
- reliability
- robustness
- integrity
- availability

**effective**
- throughput or productivity

**interoperable**
- connectivity
- 3rd party extendible

**liable**
- liability
- testability
- traceability
- standards compliance

**efficient**
- resource utilization
- cost of ownership

**consistent**
- reproducibility
- predictability

**serviceable**
- serviceability
- configurability
- installability

**future proof**
- evolvability
- portability
- upgradeability
- extendibility
- maintainability

**down to earth attributes**
- cost price
- power consumption
- consumption rate (water, air, chemicals, et cetera)
- size, weight, accuracy

**logistics friendly**
- manufacturability
- logistics flexibility
- lead time

**ecological**
- ecological footprint
- contamination
- noise
- disposability

**ecological**
- ecological footprint
- contamination
- noise
- disposability
Present the results top-down

Use two to three flip charts of the six that have been created.

Explain in five minutes the needs of the customer, the system, and the major design choices.
Method Overview

Architecting Method Overview

- **method outline**
  - framework
    - **Customer objectives**
    - Application
    - Functional
    - Conceptual
    - Realization
  - submethods
    - use case
    - decomposition
    - mapping technical functions
    - functional decomposition
    - information model
    - and many more
    - construction decomposition
  - integration
    - via qualities
    - exploration specific details
    - submethods
    - framework
    - reasoning

- **method visualization**

Modeling Method Overview

- project
  - specification
  - verification
  - decisions
  - accuracy
  - working range
  - credibility
  - risk
  - customer satisfaction
time, cost, effort
profit margin

- facts from research
- measurements
- assumptions
- modeling
- results
- analysis

- uncertainties
- unknowns
- errors
- modeling
- analysis
- results
- project

Modeling Scope

- business: profit, etc.
- operational costs
- stakeholder benefits
- workload
- risks

- key performance: throughput, response reliability
- availability
- scalability
- (emerging?) properties: resource utilization
- load
- latency, throughput
- quality, accuracy

usage context
- enterprise & users
- requirements
- black box view

system
- design
- realization
- technology

life cycle business
- creation

life cycle context
... and their mutual relations

intentionally left blank

Summary Module Architectural Reasoning Case Exploration

Gerrit Muller

version: 0.2
March 11, 2015
CAFCR views

Customer objectives
Application
Functional
Conceptual
Realization

Integrate and Iterate

What does Customer need in Product and Why?

Customer

Product

What

How

What

How

What does Customer need in Product and Why?

drives, justifies, needs
enables, supports

Product How

Customer How

Customer

Product

Application

Functional

Conceptual

Realization

Sketch on Flips, Use Note stickers

Do

- start sketching/drawing as soon as possible
- use shared large sheets of paper (e.g., flip-over)
- number the flip-overs and add a title
- annotate (add notes) during discussions
- use yellow note stickers and flip-over markers
- be open for ideas and surprises

Do not

- write long texts
- immediately capture electronic
- have nice but volatile discussions
- write with pen or pencil
- Do not stick to the first solution

Because

- sketches stimulate sharing and discussion
- sharing and discussion help to explore faster
- remembering the order gets challenging
- information and insight is quickly lost
- stickers are easily (re)moved
- you hopefully discover a lot; increased insight will change problem and solution

Plus Life Cycle view

Customer objectives
Application
Functional
Conceptual
Realization

Life cycle

development manufacturing installation

operations maintenance upgrades

sales, service, logistics, production, R&D
Abstract

This module introduces Customer Space Sampling as part of the course Architectural Reasoning using Conceptual Modeling.
Abstract

A story is an easily accessible story or narrative to make an application live. A good story is highly specific and articulated entirely in the problem domain: the native world of the users. An important function of a story is to enable specific (quantified, relevant, explicit) discussions.
From story to design

What does Customer need in Product and Why?

Customer What
Customer How
Product What
Product How

Customer objectives
Application
Functional
Conceptual
Realization

market vision
story
a priori solution knowledge

analyze design
analyze design

version: 1.1
March 11, 2015
SHTfromStoryToDesign
A day in the life of Bob

bla blah bla, rabarber music
bla blah composer bla bla
qwewewy, zops.

nja nja njet njippe est quo
vadis? Pjotr jaleski bla bla
bla bmeo fgtf gis higr
mmmm bas engel heeft een
interessant excusus, lex stelt
voor om vanavond door te
werken.

In the middle of the night he
is awake and decides to
cchange the world forever.

The next hour the great
event takes place:

This brilliant invention will change the world forever
because it is so unique and
valuable that nobody believes the feasibility. It is
great and WOW at the same time.

Vtables are seen as the solution for an
direction problem. The invention of Bob will
obsolete all of this in one incredibke move, which will make him famous forever.

He opens his PDA, logs in and enters his private secure non trivial password,
followed by a thorough authentication. The PDA asks for the fingerprint of this little left
toe and to pronounce the word shit. After passing this test Bob can continue.
Points of attention

- purpose
- scope
- viewpoint, stakeholders
- visualization
- size (max 1 A4)
- recursive decomposition, refinement
Criteria for a good story

- accessible, understandable
  "Do you see it in front of you?"
- valuable, appealing
  attractive, important
  "Are customers queuing up for this?"
- critical, challenging
  "What is difficult in the realization?"
  "What do you learn w.r.t. the design?"
- frequent, no exceptional niche
  "Does it add significantly to the bottom line?"
- specific
  names, ages, amounts, durations, titles, ...
Betty is a 70-year-old woman who lives in Eindhoven. Three years ago her husband passed away and since then she lives in a home for the elderly. Her 2 children, Angela and Robert, come and visit her every weekend, often with Betty’s grandchildren Ashley and Christopher. As so many women of her age, Betty is reluctant to touch anything that has a technical appearance. She knows how to operate her television, but a VCR or even a DVD player is way too complex.

When Betty turned 60, she stopped working in a sewing studio. Her work in this noisy environment made her hard-of-hearing with a hearing-loss of 70dB around 2kHz. The rest of the frequency spectrum shows a loss of about 45dB. This is why she had problems understanding her grandchildren and why her children urged her to apply for hearing aids two years ago. Her technophobia (and her first hints or arthritis) inhibit her to change her hearing aids’ batteries. Fortunately her children can do this every weekend.

This Wednesday Betty visits the weekly Bingo afternoon in the meetingplace of the old-folk’s home. It’s summer now and the tables are outside. With all those people there it’s a lot of chatter and babble. Two years ago Betty would never go to the bingo: “I cannot hear a thing when everyone babbles and clatters with the coffee cups. How can I hear the winning numbers?!”. Now that she has her new digital hearing instruments, even in the bingo cacophony, she can understand everyone she looks at. Her social life has improved a lot and she even won the bingo a few times.

That same night, together with her friend Janet, she attends Mozart’s opera The Magic Flute. Two years earlier this would have been one big low rumbly mess, but now she even hears the sparkling high piccolos. Her other friend Carol never joins their visits to the theaters. Carol also has hearing aids, however hers only “work well” in normal conversations. “When I hear music it’s as if a butcher’s knife cuts through my head. It’s way too sharp!”. So Carol prefers to take her hearing aids out, missing most of the fun. Betty is so happy that her hearing instruments simply know where they are and adapt to their environment.
Value and Challenges in this story

Value proposition in this story:
quality of life:
  active participation in different social settings
usability for nontechnical elderly people:
  "intelligent" system is simple to use
  loading of batteries

Challenges in this story:
Intelligent hearing instrument
Battery life — at least 1 week
No buttons or other fancy user interface on the hearing instrument, other than a robust On/Off method
The user does not want a technical device but a solution for a problem
Instrument can be adapted to the hearing loss of the user
Directional sensitivity (to prevent the so-called cocktail party effect)
Recognition of sound environments and automatic adaptation (adaptive filtering)

source: Roland Mathijssen, Embedded Systems Institute, Eindhoven
Create a story

as tekst + sketch or as cartoon

Use the criteria

be highly specific!

envision the future value proposition

Enjoy!
Abstract

Use cases are frequently used in Software Engineering. Use cases support specification and facilitate design, analysis, verification and testing. Many designers, unfortunately, apply use cases in a rather limited way. This presentation provides recommendations for effective use cases.
Why Use Cases?

Supports or is part of specification
by providing specific data in user perspective

Facilitates analysis and design

Facilitates verification and testing
Example Time Shift recording

20:00 21:00 22:00 23:00

- start movie
- broadcast
- end movie

- view
- talk
- play
- record
- phone rings
- pause viewing
- finish conversation
- resume viewing
- view
Construction limits intrude in User Experience

- number of tuners
- number of simultaneous streams (recording and playing)
- amount of available storage
- management strategy of storage space
What if?

1. programmed recording of other station
2. very long phone call
3. Dad zaps

Use Case How To
107 Gerrit Muller
version: 0.1 March 11, 2015
Content of a Use Case

**Use Case**

- User or system specified functionality
- Behavior
- Interfaces
- Qualities (NFR's)

**Input Data**
- Format
- Size
- Content

**Output Data**
- Format
- Size
- Content

**Sub)System or Component**

**Context**

**Interaction**
### Example personal video recorder use case contents

<table>
<thead>
<tr>
<th>typical use case(s)</th>
<th>worst case, exceptional, or change use case(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>interaction flow (functional aspects)</strong></td>
<td><strong>functional</strong></td>
</tr>
<tr>
<td>select movie via directory</td>
<td>multiple inputs at the same time</td>
</tr>
<tr>
<td>start movie</td>
<td>extreme long movie</td>
</tr>
<tr>
<td>be able to pause or stop</td>
<td>directory behaviour in case of</td>
</tr>
<tr>
<td>be able to skip forward or backward</td>
<td>extreme many short movies</td>
</tr>
<tr>
<td>set recording quality</td>
<td></td>
</tr>
<tr>
<td><strong>performance and other qualities (non-functional aspects)</strong></td>
<td><strong>non-functional</strong></td>
</tr>
<tr>
<td>response times for start / stop</td>
<td>response time with multiple inputs</td>
</tr>
<tr>
<td>response times for directory browsing</td>
<td>image quality with multiple inputs</td>
</tr>
<tr>
<td>end-of-movie behaviour</td>
<td>insufficient free space</td>
</tr>
<tr>
<td>relation recording quality and storage</td>
<td>response time with many directory entries</td>
</tr>
<tr>
<td></td>
<td>replay quality while HQ recording</td>
</tr>
</tbody>
</table>
Example of Quantification of Typical Use Case

3 examination rooms connected to 1 medical imaging workstation + printer

examination room: average 4 interleaved examinations / hour

image production: 20 $1024^2$ 8 bit images per examination

film production: 3 films of 4k*5k pixels each

high quality output (bi-cubic interpolation)
Recommendations for working with use cases

+ combine related functions in one use case
- do not make a separate use case for every function
+ include non-functional requirements in the use cases

+ minimise the amount of required *worst case* and *exceptional use cases*
- excessive amounts of use cases propagate to excessive implementation efforts
+ reduce the amount of these use cases in steps
- a few well chosen *worst case* use cases simplifies the design
Use Case Exercise

Make specification overview with ~10 SMART Key Performance Parameters (or functions or interfaces)

determine at least one use case

- **S**pecific (quantified)
- **M**easurable (verifiable)
- **A**chievable (Attainable, Action oriented, Acceptable, Agreed-upon, Accountable)
- **R**ealistic (Relevant, Result-Oriented)
- **T**ime-bounded (Timely, Tangible, Traceable)

use case
typical use with relevant context data (quantified!)
Use Case Summary

Customer Language

Customer
What does Customer need in Product and Why?

Customer
How
Product
What
Product
How

What does Customer need in Product and Why?

Story and Use Case Summary

Accesible and Specific to Learn

• accessible, understandable
  “Do you see it in front of you?”

• valuable, appealing
  attractive, important
  “Are customers queuing up for this?”

• critical, challenging
  “What is difficult in the realization?”
  “What do you learn w.r.t. the design?”

• frequent, no exceptional niche
  “Does it add significantly to the bottom line?”

• specific
  names, ages, amounts, durations, titles, ...

Use Cases include Quantification

3 examination rooms connected to 1 medical imaging workstation + printer

examination room: average 4 interleaved examinations / hour

image production: 20 1024² 8 bit images per examination

film production: 3 films of 4k*5k pixels each high quality output (bi-cubic interpolation)

Typical and Worst case

typical use case(s)

interaction flow (functional aspects)
select movie via directory
start movie
be able to pause or stop
be able to skip forward or backward
set recording quality

performance and other qualities (non-functional aspects)
response times for start / stop
response times for directory browsing
end-of-movie behaviour
relation recording quality and storage

worst case, exceptional, or change use case(s)

functional
multiple inputs at the same time
extreme long movie
directory behaviour in case of extreme many short movies

non-functional
response time with multiple inputs
image quality with multiple inputs insufficient free space
response time with many directory entries
replay quality while HQ recording

Summary Module Architectural Reasoning Customer Space Sampling

version: 0.1
March 11, 2015

Gerrit Muller
Abstract

This module discusses fundamental design methods and techniques, especially partitioning, interface, behavior, and quantified performance design.
Abstract

The fundamental concepts and approach system partitioning are explained. We look at physical decomposition and functional decomposition in relation to supply chain, lifecycle support, project management, and system specification and design.
Parts, Dynamics, Characteristics

- **Characteristics** interact with **Dynamics** resulting in **Functionality**.
- **Functionality** is the **Prime Interest of Customer**.
- **Functionality** also serves as the **Prime Interest of Organization**.
- **Prime System Responsibility** is involved in this process.
Engineering

- engineering knowledge
- system specification
- system design
- source data

- parts data base
- production procedures
- qualification procedures
- system documentation

- procurement
- production
- installation
- quality assurance
- lifecycle support

- knowledge DB
- doc DB
- CAD
- SCM
- ERP
- PDM

- past experience
- project documents
- mechanical electrical design database
- source code management
- resource planning, e.g. SAP
- product data management

System Partitioning Fundamentals
118  Gerrit Muller

version: 0.2
March 11, 2015
SPFEngineering
Example Physical Decomposition

System Partitioning Fundamentals
119  Gerrit Muller

version: 0.2
March 11, 2015
REPLIsystemsAll
Partitioning is Applied Recursively

- subsystem 1
  - subsub system A
  - atomic part
  - subsub system B
  - atomic part
  - subsub system N
  - atomic part
- subsystem 2
  - subsub system A
  - subsub system B
  - subsub system N
- subsystem n
  - subsub system A
  - subsub system B
  - subsub system N
Software plus Hardware Decomposition

System Partitioning Fundamentals

121  Gerrit Muller
the part is cohesive

  functionality and technology belongs together

the coupling with other parts is minimal

  minimize interfaces

the part is selfsustained for production and qualification

  can be in conflict with cost or space requirements

clear ownership of part

  e.g. one department or supplier
How much self-sustained?

control SW  application SW  HMI SW  control electronics  control interface

cooling  EMC shielding  main function  qualification support  adjustment support

power stabilization  power conversion  power distribution  production support  mechanical package

How self sustained should a part be?

trade-off:

cost/speed/space optimization  logistics/lifecycle/production flexibility clarity
Decoupling via Interfaces

System Partitioning Fundamentals
124  Gerrit Muller

version: 0.2
March 11, 2015
SPFInterfaceDecoupling
System is composed

by using standard interfaces

limited catalogue of variants (e.g. cost performance points)
System Partitioning Fundamentals

126  Gerrit Muller

version: 0.2
March 11, 2015
SPFsystemCreation

System Creation

stakeholder needs → architecting
business objectives → architecting

architecture guidelines

specification

top-level design rationale

design

partitioning interfaces
design allocation

engineering

documentation

system and parts data

procedures

procurement

production

installation

quality assurance

lifecycle support
Simplistic Functional SubSea Example

- Prevent blow-outs
- Regulate flow and pressure
- Combine multiple streams
- Separate gas, oil, water, sand
- Increase well pressure
- Transport to top-side
- Measure pressure, temp, flow
- Control pressure, temp, flow

Sensor signals: pressure, temp, flow

Sensor data: settings

Hydrocarbons from well: water, sand

System Partitioning Fundamentals
127 Gerrit Muller

version: 0.2
March 11, 2015
Functional Decomposition

How does the system work and operate?
Functions describe *what* rather than *how*.
Functions are *verbs*.
Input-Process-Output paradigm.
Multiple kinds of flows:
  - physical (e.g. hydrocarbons)
  - information (e.g. measurements)
  - control
At lower level one part \( \sim = \) one function
  - pump pumps, compressor compresses, controller controls
At higher level functions are complex interplay of physical parts
  - e.g. regulating constant flow, pressure and temperature
<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>2.4m * 0.7m * 1.3m</td>
</tr>
<tr>
<td>Weight</td>
<td>1450 Kg</td>
</tr>
<tr>
<td>Cost</td>
<td>30000 NoK</td>
</tr>
<tr>
<td>Reliability</td>
<td>MTBF 4000 hr</td>
</tr>
<tr>
<td>Throughput</td>
<td>3000 l/hr</td>
</tr>
<tr>
<td>Response time</td>
<td>0.1 s</td>
</tr>
<tr>
<td>Accuracy</td>
<td>+/- 0.1%</td>
</tr>
</tbody>
</table>

Many characteristics of a system, function or part can be quantified. Note that quantities have unit.
How about the **accuracy** of the **fuse** when performing **printing**?

What is the **accuracy** of the **fuse** when **printing**?

Example from a high volume printer
Example Technical Budget

- **Process Overlay Budget**
  - 80 nm
  - Matched Machine 60 nm
  - Single Machine 30 nm
  - Process Dependency Sensor 5 nm

- **Lenses**
  - Matching 25 nm

- **Stage Overlay**
  - Accuracy 12 nm

- **Position Accuracy**
  - 7 nm

- **Alignment Repro**
  - 5 nm

- **Metrology Stability**
  - 5 nm

- **Stage Al. Pos. Meas. Accuracy**
  - 4 nm

- **System Adjustment Accuracy**
  - 2 nm

- **Global Alignment Accuracy**
  - 6 nm

- **Interferometer Stability**
  - 1 nm

- **Tracking**
  - Error WS 2 nm
  - Error RS 1 nm

- **Tracking Error X, Y**
  - 2.5 nm

- **Tracking Error Phi**
  - 75 nrad

- **Off Axis Sensor Repro**
  - 3 nm

- **Blue Align Sensor Repro**
  - 3 nm

- **Tracking Error Phi**
  - 75 nrad

- **Off Axis Pos. Meas. Accuracy**
  - 4 nm

- **Global Alignment Accuracy**
  - 6 nm
Example of A3 overview

A3 architecture overview of the Metal Printer (all numbers have been removed for competitive sensitivity)

1. Close doors
2. Align
3. Move to proximity
4. Process
5. Move substrate unloading position
6. Open doors

System Partitioning Fundamentals
Gerrit Muller
March 11, 2015
LEANOverviewA3
Exercise Dynamic Behavior

Capture the **dynamic behavior** of the **internals** of your system in **multiple** diagrams.

Diagrams that capture dynamic behavior are among others:

- Functional flow (of control or information, material or goods, or energy)
- Activity or sequence diagrams (e.g. with “swimming lanes”)
- State diagrams
Make a set of **block diagrams** capturing the **static parts** and **interfaces**.

Ensure coverage of the entire system, e.g. including service, training, production, etc.

Show both **hardware** and **software**

Good block diagram have in the order of 10 to 20 blocks
Design Fundamentals

Parts, Dynamics, Characteristics

- **characteristics** prime interest of customer
- **dynamics** functionality
- **parts** prime interest of organization

Decoupling via Interfaces

- **power interface**
- **control interface e.g. CAN**
- **part e.g. pipe**
- **part e.g. pressure and flow regulator**
- **part e.g. pipe**
- **other part with same interfaces can replace original**

Dynamic Behavior

- **increase well pressure**
- **prevent blow-outs**
- **regulate flow and pressure**
- **combine multiple streams**
- **separate gas, oil, water, sand**
- **transmit to top-side**
- **hydrocarbons from well**
- **water sand**

Question Generator

- How about the **<characteristic>** of the **<component>** when performing **<function>**?
- What is the **accuracy** of the **fuse** when **printing**?

Example from a high volume printer

Summary Module Architectural Reasoning Design Fundamentals
135 Gerrit Muller

version: 0.2 March 11, 2015
Module 34, Architectural Reasoning Customer Space Analysis

by Gerrit Muller    HBV-NISE

e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

This module provides methods and techniques to analyze the customer space.
Abstract

This presentation provides a set of techniques to explore the customer perspective. The main purpose is for an organization to understand its customer sufficiently. Architects need this level of understanding to guide specification and design.
## Overview of methods

<table>
<thead>
<tr>
<th>what</th>
<th>story telling, scenario</th>
<th><a href="http://www.gaudisite.nl/info/StoryHowTo.info.html">http://www.gaudisite.nl/info/StoryHowTo.info.html</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>who</td>
<td>stakeholders and concerns</td>
<td>humans organizations autonomous behavior emotions</td>
</tr>
<tr>
<td>how</td>
<td>system context diagram human-made artifacts</td>
<td></td>
</tr>
<tr>
<td>workflow</td>
<td></td>
<td></td>
</tr>
<tr>
<td>when</td>
<td>timeline from seconds to years</td>
<td></td>
</tr>
<tr>
<td>where</td>
<td>map from nanometers to kilometers</td>
<td></td>
</tr>
<tr>
<td>why</td>
<td>customer key driver graph productivity model</td>
<td><a href="http://www.gaudisite.nl/info/KeyDriversHowTo.info.html">http://www.gaudisite.nl/info/KeyDriversHowTo.info.html</a></td>
</tr>
<tr>
<td>financial</td>
<td>cost of ownership model money flow</td>
<td></td>
</tr>
</tbody>
</table>
Scenario: Patient George

- Patient George has continuous headache.
- His family doctor has send him to the Neurologist.
- The Neurologist wants to exclude the possibility of a tumor and requests an MRI examination.
- The Radiologists does not see any indication for a tumor.
- The Radiologist sends his report to the Neurologist.
- The Neurologist discusses his findings with the patient and sends a report to the family doctor.
From Complaint to Diagnosis

Methods to Explore the Customer Perspective
140  Gerrit Muller

version: 0
March 11, 2015
MRradiologyPatientAndPhysicians
Methods to Explore the Customer Perspective

Gerrit Muller

version: 0  
March 11, 2015
MECPcontextDiagram
weeks view: from Complaint to Diagnosis

functional flow
- call family doctor
- visit family doctor
- call neurology department
- visit neurologist
- call radiology department
- examination itself
- diagnosis by radiologist
- report from radiologist to neurologist
- visit neurologist

Methods to Explore the Customer Perspective
145  Gerrit Muller

version: 0
March 11, 2015
MRendToEndTimeLine
half hour view: Examination

George arrives at radiology department
Nurse explains the procedure
George is waiting in the dressing room
Prepare George for the examination
(a.o. RF coils)
Position Imaging
View away
View away
George leaves exam room
Examination of previous patient
15 minute time slot

14:00
14:15
14:30

Methods to Explore the Customer Perspective
Gerrit Muller
version: 0
March 11, 2015
MRneuroTypicalTimeline
5 minute view: Patient Preparation (1 operator)

**functional procedure**

- walk from dressing room to table
- position patient on table
- move table upwards
- position coils and connect
- move table and patient into magnet
- make plan scan

<table>
<thead>
<tr>
<th>walk</th>
<th>position on table</th>
<th>table up</th>
<th>talk</th>
<th>coils</th>
<th>in magnet</th>
<th>walk</th>
<th>talk</th>
<th>plan scan</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14:15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14:20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Patient Preparation Work Flow

1. Get patient
2. Patient on table
3. Get RF coil
4. Position RF coil
5. Move patient in magnet
6. Plan scan

preparation work flow
1. Get patient
2. Patient on table
3. Get RF coil
4. Position RF coil
5. Move patient in magnet
6. Plan scan
Productivity and Cost models

**Productivity model**
- Use
- Events
- Configuration
- Working conditions
- Production rate

**Cost Of Ownership model**
- Radiologist
- Nurse
- Security
- Administration
- Operator
- Personnel
- Consumables
- Service
- Facilities
- Financing
The financial context of the radiology department

Methods to Explore the Customer Perspective
151  Gerrit Muller
Make a **context diagram**, showing the **systems** and their **relations** in the **customer space**

- typically, tens of systems are relevant for customers

Capture one or a few main **workflows** in the customer space
Abstract

The notion of "business key drivers" is introduced and a method is described to link these key drivers to the product specification.
Example Motorway Management Analysis

Key-drivers
- Safety
  - Reduce accident rates
  - Enforce law
  - Improve emergency response
- Effective Flow
  - Reduce delay due to accident
  - Improve average speed
  - Improve total network throughput
  - Optimize road surface
  - Speed up target groups
  - Anticipate on future traffic condition
- Smooth Operation
  - Ensure traceability
  - Ensure proper alarm handling
  - Ensure system health and fault indication
- Environment
  - Reduce emissions

Derived application drivers
- Early hazard detection with warning and signaling
- Maintain safe road condition
- Classify and track dangerous goods vehicles
- Detect and warn noncompliant vehicles
- Enforce speed compliance
- Enforce red light compliance
- Enforce weight compliance

Requirements
- Automatic upstream accident detection
- Weather condition dependent control
- Traffic speed and density measurement
- Cameras
- Deicing
- Traffic condition dependent speed control

Note: The graph is only partially elaborated for application drivers and requirements.
Method to create Key Driver Graph

- Define the scope specific. in terms of stakeholder or market segments

- Acquire and analyze facts extract facts from the product specification and ask why questions about the specification of existing products.

- Build a graph of relations between drivers and requirements where requirements may have multiple drivers by means of brainstorming and discussions

- Obtain feedback discuss with customers, observe their reactions

- Iterate many times increased understanding often triggers the move of issues from driver to requirement or vice versa and rephrasing
<table>
<thead>
<tr>
<th>Recommendation</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limit the number of key-drivers</td>
<td>minimal 3, maximal 6</td>
</tr>
<tr>
<td>Don’t leave out the obvious key-drivers</td>
<td>for instance the well-known main function of the product</td>
</tr>
<tr>
<td>Use short names, recognized by the customer.</td>
<td></td>
</tr>
<tr>
<td>Use market-/customer- specific names, no generic names</td>
<td>for instance replace “ease of use” by “minimal number of actions for experienced users”, or “efficiency” by “integral cost per patient”</td>
</tr>
<tr>
<td>Do not worry about the exact boundary between Customer Objective and Application</td>
<td>create clear goal means relations</td>
</tr>
</tbody>
</table>
Transformation of Key Drivers into Requirements

Customer What
- Customer objectives

Key (Customer) Drivers

Customer How
- Derived Application Drivers

Product What
- Functional
- Goal
  - means
  - may be skipped or articulated by several intermediate steps
- Functions
  - interfaces
  - performance figures

 REQfromDriverToRequirement

version: 0.2
March 11, 2015

Key Drivers How To
Gerrit Muller
Make a **customer key driver graph**

Use yellow note stickers

Start at the right hand side

![Customer Key Driver Graph]

- why
- 5 m/s
- why
- <200 Kg
- 5 hrs
Analysis Methods and Techniques

Stakeholders and Concerns (Who)

- government
  - cost of care
- financial dir.
  - cash flow
  - cost of op.
- insurance
  - cost of care
- administration
  - patient id
  - invoice
- general practitioner
  - patient
- patient
  - comfort
  - health
- family
  - support
- IT dep.
  - conformance
  - security
- facility man.
  - space
  - service supp.
- maintainer
  - accessibility
  - safety
- cleaner
  - accessibility
  - safety
- nurse
  - diagnosis
  - reimbursement
- radiologist
  - diagnosis
- operator
  - ease of use
- inspection
  - quality
- operator
  - ease of use
- cleaner
  - accessibility
  - safety
- safety
- accessibility
- inspection
- quality
- accessibility
- support

Workflow (what dynamics)

- request exam
- schedule exam
- perform exam
- send report
- receive patient
- prepare patient
- examine patient
- release patient
- patient
  - undresses
  - move patient
  - position on table
  - attach coils
  - move into magnet

Context Diagram (what systems)

- hospital ERP
- reception clients
- clinical clients
- primary care
- insurance portal
- patient portal
- RIS (radiology)
- patient info report
- patient info schedule
  - status
- MRI images
- CT images
- Xray images
- PACS (Picture Archiving and Communication)
- IT infrastructure (communication, gateways, servers, storage, ...)
- physician workstation
- administrative
- clinical
- imaging
- patient
- support
- external stakeholders

Information Flow

- richness
- clinical value
- acquire images
  - prepare diagnosis
  - diagnosis
  - report authorise
- archive
- medical imaging workstation
- clinical review
- education
- research
- treatment planning
- demonstration
- time
More Analysis Methods and Techniques

Timeline (when, what, who)

1. Call family doctor
2. Visit family doctor
3. Call neurology department
4. Visit neurologist
5. Call radiology department
6. Examination itself
7. Diagnosis by radiologist
8. Report from radiologist to neurologist
9. Visit neurologist

2D or 3D map (where)

Annotated map (where, what)

Preparation work flow:
1. Get patient
2. Patient on table
3. Get RF coil
4. Position RF coil
5. Move patient in magnet
6. Plan scan

Cost Models

Cost Of Ownership model:
- Radiologist
- Nurse
- Security
- Administration
- Facilities
- Operator
- Personnel
- Consumables
- Service
- Financing

Production rate:
- Typical
- Use
- Events
- Configuration
- Working conditions

Productivity model:
- Version: 0.2
- March 11, 2015
Customer Key Driver Graph

Focus on Customer World

Key-drivers
Safety
- Reduce accident rates
- Enforce law
- Improve emergency response
- Maintain safe road condition
- Optimise road surface
- Speed up target groups
- Anticipate on future traffic condition

Effective Flow
- Reduce delay due to accidents
- Improve average speed
- Improve total network throughput
- Ensure traceability
- Ensure proper alarm handling
- Ensure system health and fault indication

Smooth Operation
- Reduce emissions

Derived application drivers
Early hazard detection with warning and signaling
- Speed up target groups

Requirements
- Automatic upstream accident detection
- Weather condition dependent control
- Traffic speed and density measurement
- Cameras

Note: the graph is only partially elaborated for application drivers and requirements

Specific Scope, Fact Based

- Define the scope specific.
- Acquire and analyze facts and ask why questions about the specification of existing products.
- Build a graph of relations between drivers and requirements by means of brainstorming and discussions where requirements may have multiple drivers.
- Obtain feedback discuss with customers, observe their reactions.
- Iterate many times increased understanding often triggers the move of issues from driver to requirement or vice versa and rephrasing.

3 to 6 Key driver, Capture Tensions

- Limit the number of key-drivers minimal 3, maximal 6
- Don’t leave out the obvious key-drivers for instance the well-known main function of the product
- Use short names, recognized by the customer.
- Use market-/customer- specific names, no generic names for instance replace “ease of use” by “minimal number of actions for experienced users”, or “efficiency” by “integral cost per patient”.
- Do not worry about the exact boundary between Customer Objective and Application create clear goal means relations

intentionally left blank

Summary Module Architectural Reasoning Customer Space Analysis
161 Gerrit Muller
version: 0.2 March 11, 2015
Module 31, Architectural Reasoning Conceptual Design

by Gerrit Muller       HBV-NISE

e-mail: gaudisite@gmail.com

www.gaudisite.nl

Abstract

This module conceptual design methods, such as budgeting and concept selection.
Abstract

This presentation addresses the fundamentals of budgeting: What is a budget, how to create and use a budget, what types of budgets are there. What is the relation with modeling and measuring.
content of this presentation

What and why of a budget

How to create a budget (decomposition, granularity, inputs)

How to use a budget
What is a Budget?

A budget is

*a quantified instantiation of a model*

A budget can

*prescribe or describe the contributions*

*by parts of the solution*

*to the system quality under consideration*
Why Budgets?

- to make the design explicit
- to provide a baseline to take decisions
- to specify the requirements for the detailed designs
- to have guidance during integration
- to provide a baseline for verification
- to manage the design margins explicitly
Visualization of Budget Based Design Flow

can be more complex than additions

t_{proc} + t_{over} + t_{disp} + t_{over} = model

SRS
\[
\begin{array}{|c|}
\hline
\text{t_{boot}} & 0.5s \\
\text{t_{cap}} & 0.2s \\
\hline
\end{array}
\]
spec

measurements
existing system

design estimates; simulations

feedback

budget

<table>
<thead>
<tr>
<th>t_{proc}</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>t_{over}</td>
<td>20</td>
</tr>
<tr>
<td>T_{proc}</td>
<td>30</td>
</tr>
<tr>
<td>t_{disp}</td>
<td>5</td>
</tr>
<tr>
<td>t_{over}</td>
<td>20</td>
</tr>
<tr>
<td>T_{disp}</td>
<td>25</td>
</tr>
<tr>
<td>T_{total}</td>
<td>55</td>
</tr>
</tbody>
</table>

new (proto)

system

tuning

micro benchmarks
aggregated functions
applications
profiles
traces

Modeling and Analysis: Budgeting
Gerrit Muller

version: 1.0
March 11, 2015
EAAbudget
<table>
<thead>
<tr>
<th>step</th>
<th>example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1A measure old systems</td>
<td>micro-benchmarks, aggregated functions, applications</td>
</tr>
<tr>
<td>1B model the performance starting with old systems</td>
<td>flow model and analytical model</td>
</tr>
<tr>
<td>1C determine requirements for new system</td>
<td>response time or throughput</td>
</tr>
<tr>
<td>2 make a design for the new system</td>
<td>explore design space, estimate and simulate</td>
</tr>
<tr>
<td>3 make a budget for the new system:</td>
<td>models provide the structure measurements and estimates provide initial numbers specification provides bottom line</td>
</tr>
<tr>
<td>4 measure prototypes and new system</td>
<td>micro-benchmarks, aggregated functions, applications profiles, traces</td>
</tr>
<tr>
<td>5 Iterate steps 1B to 4</td>
<td></td>
</tr>
</tbody>
</table>
## Budgets Applied on Medical Workstation Memory Use

### memory budget in Mbytes

<table>
<thead>
<tr>
<th></th>
<th>code</th>
<th>obj data</th>
<th>bulk data</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>shared code</td>
<td>11.0</td>
<td></td>
<td></td>
<td>11.0</td>
</tr>
<tr>
<td>User Interface process</td>
<td>0.3</td>
<td>3.0</td>
<td>12.0</td>
<td>15.3</td>
</tr>
<tr>
<td>database server</td>
<td>0.3</td>
<td>3.2</td>
<td>3.0</td>
<td>6.5</td>
</tr>
<tr>
<td>print server</td>
<td>0.3</td>
<td>1.2</td>
<td>9.0</td>
<td>10.5</td>
</tr>
<tr>
<td>optical storage server</td>
<td>0.3</td>
<td>2.0</td>
<td>1.0</td>
<td>3.3</td>
</tr>
<tr>
<td>communication server</td>
<td>0.3</td>
<td>2.0</td>
<td>4.0</td>
<td>6.3</td>
</tr>
<tr>
<td>UNIX commands</td>
<td>0.3</td>
<td>0.2</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>compute server</td>
<td>0.3</td>
<td>0.5</td>
<td>6.0</td>
<td>6.8</td>
</tr>
<tr>
<td>system monitor</td>
<td>0.3</td>
<td>0.5</td>
<td>0</td>
<td>0.8</td>
</tr>
<tr>
<td><strong>application SW total</strong></td>
<td><strong>13.4</strong></td>
<td><strong>12.6</strong></td>
<td><strong>35.0</strong></td>
<td><strong>61.0</strong></td>
</tr>
<tr>
<td>UNIX Solaris 2.x</td>
<td></td>
<td></td>
<td></td>
<td>10.0</td>
</tr>
<tr>
<td>file cache</td>
<td></td>
<td></td>
<td></td>
<td>3.0</td>
</tr>
<tr>
<td><strong>total</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>74.0</strong></td>
</tr>
</tbody>
</table>
Power Budget Visualization for Document Handler

- Scanner and feeder
- Procédu
- Paper path
- Finisher
- Power supplies
- Cooling
- UI and control
- Paper input module

Legend:
- Physical layout
- Size proportional to power

Modeling and Analysis: Budgeting
Gerrit Muller

version: 1.0
March 11, 2015
MDMpowerProportions
Alternative Power Visualization

- Power supplies
- UI and control
- Paper input module
- Paper path
- Finisher
- Electrical power
- Cooling
- Paper
- Procedé
### Evolution of Budget over Time

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>fact finding through details</td>
<td>aggregate to end-to-end performance</td>
</tr>
<tr>
<td>aggregate to end-to-end performance</td>
<td>search for appropriate abstraction level(s)</td>
</tr>
<tr>
<td>from coarse guesstimate</td>
<td>to reliable prediction</td>
</tr>
<tr>
<td>from typical case</td>
<td>to boundaries of requirement space</td>
</tr>
<tr>
<td>from static understanding</td>
<td>to dynamic understanding</td>
</tr>
<tr>
<td>from steady state</td>
<td>to initialization, state change and shut down</td>
</tr>
<tr>
<td>from old system</td>
<td>to prototype</td>
</tr>
<tr>
<td></td>
<td>to actual implementation</td>
</tr>
</tbody>
</table>

---

**time**

*start* *later* *only if needed*
Potential Applications of Budget based design

• resource use (CPU, memory, disk, bus, network)
• timing (response, latency, start up, shutdown)
• productivity (throughput, reliability)
• Image Quality parameters (contrast, SNR, deformation, overlay, DOF)
• cost, space, time
What kind of budget is required?

<table>
<thead>
<tr>
<th>Static</th>
<th>Dynamic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Typical Case</td>
<td>Worst Case</td>
</tr>
<tr>
<td>Global</td>
<td>Detailed</td>
</tr>
<tr>
<td>Approximate</td>
<td>Accurate</td>
</tr>
</tbody>
</table>

Is the budget based on wish, empirical data, extrapolation, educated guess, or expectation?
Summary of Budgeting

A budget is a quantified instantiation of a model

A budget can prescribe or describe the contributions by parts of the solution to the system quality under consideration

A budget uses a decomposition in tens of elements

The numbers are based on historic data, user needs, first principles and measurements

Budgets are based on models and estimations

Budget visualization is critical for communication

Budgeting requires an incremental process

Many types of budgets can be made; start simple!
The Boderc project contributed to Budget Based Design. Especially the work of

Hennie Freriks, Peter van den Bosch (Océ),

Heico Sandee and Maurice Heemels (TU/e, ESI)

has been valuable.
Exercise Budget

Make a technical budget for one of the key performance parameters.

- a good budget has 20 to 30 contributing elements
- elements should be balanced (remove or combine insignificant contributions)
- use the previously defined parts and dynamic behavior
Abstract

We discuss a systems design approach where several design options are maintained concurrently. In LEAN Product Development this is called set-based design. Concentioneal systems engineering also promotes the concurrent evaluation of multiple concepts, the so-called concept selection. Finally, LEAN product development advocates to keep options open as long as feasible; the so-called late decision making.
Problem Solving Approach

1. Problem understanding by exploration and simple models

2. Analysis by
   - exploring multiple propositions (specification + design proposals)
   - exploring decision criteria (by evaluation of proposition feedback)
   - assessment of propositions against criteria

3. Decision by
   - review and agree on analysis
   - communicate and document

4. Monitor, verify, validate by
   - measurements and testing
   - assessment of other decisions

vague problem statement

conflicting other decision

insufficient data

no satisfying solution

invalidated solution
Examples of Pugh Matrix Application

**Swivel concept selection**

<table>
<thead>
<tr>
<th>Evaluation Criteria</th>
<th>CBV</th>
<th>Clamp</th>
<th>Dynamic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maturity</td>
<td>10</td>
<td>5</td>
<td>50</td>
</tr>
<tr>
<td>Development level</td>
<td></td>
<td>2</td>
<td>20</td>
</tr>
<tr>
<td>Cost</td>
<td>20</td>
<td>5</td>
<td>100</td>
</tr>
<tr>
<td>Hardware cost</td>
<td>4</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>Development cost</td>
<td>5</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Design robustness</td>
<td>25</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Design life</td>
<td>125</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Swivel cycles</td>
<td>75</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Pressure cycles</td>
<td>75</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Pressure range</td>
<td>125</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Internal</td>
<td>100</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>External</td>
<td>100</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Temperature range</td>
<td>100</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Installation</td>
<td>20</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Initial install/retrieval</td>
<td>40</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Connection/disconnection</td>
<td>4</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>Operation</td>
<td>25</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Swivel resistance</td>
<td>25</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Spool Length Short</td>
<td>75</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Spool Length Long</td>
<td>75</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Hub loads</td>
<td>50</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Weight</td>
<td>985</td>
<td>1165</td>
<td></td>
</tr>
<tr>
<td>CBV</td>
<td></td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>Clamp</td>
<td>20</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Dynamic</td>
<td>50</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Scores</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Time to connect</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Need for ROV</td>
<td>-</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>Design</td>
<td>-</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>Robustness</td>
<td>-</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>Connector design</td>
<td>+</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Number of parts</td>
<td>+</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>Handle roll-off</td>
<td>-</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>Influence other</td>
<td>+</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Redundancy</td>
<td>+</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Design</td>
<td>+</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Interchangeability</td>
<td>+</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Cost</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>HW cost</td>
<td>S</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>Manufacturing cost</td>
<td>+</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>Engineering cost</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Service cost</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>Maturity</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>SUM -</td>
<td>7</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>SUM +</td>
<td>1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>SUM S</td>
<td>5</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Pos.</td>
<td>3</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

**EDP-LRP connection**

<table>
<thead>
<tr>
<th>Evaluation Criteria</th>
<th>Score</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time to connect</td>
<td></td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Need for ROV</td>
<td></td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Design</td>
<td></td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Robustness</td>
<td></td>
<td>-</td>
<td>S</td>
<td>S</td>
<td>+</td>
</tr>
<tr>
<td>Connector design</td>
<td></td>
<td>+</td>
<td>-</td>
<td>S</td>
<td>+</td>
</tr>
<tr>
<td>Number of parts</td>
<td></td>
<td>-</td>
<td>S</td>
<td>-</td>
<td>S</td>
</tr>
<tr>
<td>Handle roll-off</td>
<td></td>
<td>+</td>
<td>-</td>
<td>S</td>
<td>+</td>
</tr>
<tr>
<td>Influence other</td>
<td></td>
<td>+</td>
<td>-</td>
<td>S</td>
<td>+</td>
</tr>
<tr>
<td>Redundancy</td>
<td></td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>S</td>
</tr>
<tr>
<td>Design</td>
<td></td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Interchangeability</td>
<td></td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Cost</td>
<td></td>
<td>S</td>
<td>S</td>
<td>-</td>
<td>S</td>
</tr>
<tr>
<td>HW cost</td>
<td></td>
<td>+</td>
<td>-</td>
<td>S</td>
<td>-</td>
</tr>
<tr>
<td>Manufacturing cost</td>
<td></td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Engineering cost</td>
<td></td>
<td>-</td>
<td>-</td>
<td>S</td>
<td>+</td>
</tr>
<tr>
<td>Service cost</td>
<td></td>
<td>+</td>
<td>-</td>
<td>S</td>
<td>+</td>
</tr>
<tr>
<td>Maturity</td>
<td></td>
<td>+</td>
<td>-</td>
<td>S</td>
<td>+</td>
</tr>
</tbody>
</table>

From master paper Halvard Bjørnsen, 2009

From master paper Dag Jostein Klever, 2009
Evolution of Design Options

Concept Selection, Set Based Design and Late Decision Making

version: 0
March 11, 2015
CSSBsetEvolution

Gerrit Muller

Embedded Systems Innovation
Conclusions

Evolving multiple concepts increases insight and understanding (LEAN product development: set-based design, SE: Pugh matrix)

Articulation of criteria sharpens evaluation

The discussion about the Pugh matrix is more valuable than final bottomline summation

Delaying decisions may help to keep options (Lean Product Development: late decision making, finance: real options)
Make a **decision matrix** for one of the concept selections.

- define at least 3 concepts
- define 7 to 10 criteria for selection
- score the concepts against the criteria, for example using a scale from 1 to 5: 1 = very poor, 5 = very good
- recommend a concept with a rationale

<table>
<thead>
<tr>
<th>criterion 1</th>
<th>concept 1</th>
<th>concept 2</th>
<th>concept 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><img src="image" alt="1" /></td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>criterion n</td>
<td>4</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>

**best, because ...**
Budgeting

Budget: Decomposition of Contributions

Tens of (Measurable) Numbers

<table>
<thead>
<tr>
<th>memory budget in Mbytes</th>
<th>code</th>
<th>obj data</th>
<th>bulk data</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>shared code</td>
<td>11.0</td>
<td></td>
<td></td>
<td>11.0</td>
</tr>
<tr>
<td>User Interface process</td>
<td>0.3</td>
<td>3.0</td>
<td>12.0</td>
<td>15.3</td>
</tr>
<tr>
<td>database server</td>
<td>0.3</td>
<td>3.2</td>
<td>3.0</td>
<td>6.5</td>
</tr>
<tr>
<td>print server</td>
<td>0.3</td>
<td>1.2</td>
<td>9.0</td>
<td>10.5</td>
</tr>
<tr>
<td>optical storage server</td>
<td>0.3</td>
<td>2.0</td>
<td>1.0</td>
<td>3.3</td>
</tr>
<tr>
<td>communication server</td>
<td>0.3</td>
<td>2.0</td>
<td>4.0</td>
<td>6.3</td>
</tr>
<tr>
<td>UNIX commands</td>
<td>0.3</td>
<td>0.2</td>
<td>0.0</td>
<td>0.5</td>
</tr>
<tr>
<td>compute server</td>
<td>0.3</td>
<td>0.5</td>
<td>6.0</td>
<td>6.8</td>
</tr>
<tr>
<td>system monitor</td>
<td>0.3</td>
<td>0.5</td>
<td>0.0</td>
<td>0.8</td>
</tr>
<tr>
<td>application SW total</td>
<td>13.4</td>
<td>12.6</td>
<td>35.0</td>
<td>61.0</td>
</tr>
<tr>
<td>UNIX Solaris 2.x</td>
<td></td>
<td></td>
<td></td>
<td>10.0</td>
</tr>
<tr>
<td>file cache</td>
<td></td>
<td></td>
<td></td>
<td>3.0</td>
</tr>
<tr>
<td>total</td>
<td>74.0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Concept Selection and Evolution

Understand Problem, Analyze, Decide, Monitor

1. Problem understanding by exploration and simple models
2. Analysis by + exploring multiple propositions (specification + design proposals) + exploring decision criteria (by evaluation of proposition feedback) + assessment of propositions against criteria
3. Decision by + review and agree on analysis + communicate and document
4. Monitor, verify, validate by + measurements and testing + assessment of other decisions

Evolution of design Options

intentionally left blank
Abstract

This module provides methods and techniques to analyze the business and lifecycle context.
Abstract

This document explains how simple financial estimates can be made by system architects. These simplistic estimates are useful for an architect to perform sanity checks on proposals and to obtain understanding of the financial impact of proposals. Note that architects will never have full fledged financial controller know how and skills. These estimates are zero order models, but real business decisions will have to be founded on more substantial financial proposals.
Product Margin = Sales Price - Cost

Margin per product. The margin over the sales volume, must cover the fixed costs, and generate profit transportation, insurance, royalties per product, ...

Cost per product, excluding fixed costs purchase price of components may cover development cost of supplier

Material

Labour

Miscellaneous

Margin

Retailer margin and costs

Cost price

Sales price

Street price

Simplistic Financial Computations for System Architects.

Gerrit Muller
Profit as function of sales volume

- Sales volume
- Income
- Expenses
- Break even point
- Profit
- Fixed costs
- Variable costs
- Expected sales volume

Graph showing the income and expenses as a function of sales volume, with the break even point and expected sales volume indicated.
Investments, more than R&D

- financing
- marketing, sales
- training sales & service
- NRE: outsourcing, royalties
- research and development

Business dependent:
- pharmaceutics industry
- sales cost >> R&D cost

Strategic choice:
- NRE or per product

Including:
- staff, training, tools, housing
- materials, prototypes
- overhead
- certification

Often a standard staffing rate is used that covers most costs above:
- R&D investment = Effort * rate

Simplistic Financial Computations for System Architects.
version: 1.3
March 11, 2015
SFCinvestments
Income, more than product sales only

\[
\sum_{\text{services}} \text{income}_{\text{service}} \\
\sum_{\text{options}} \text{sales price}_{\text{option}} \times \text{volume}_{\text{option}} \\
\text{sales price}_{\text{product}} \times \text{volume}_{\text{product}}
\]

- License fees
- Pay per movie
- Content, portal updates
- Maintenance
- Other recurring income
- Products, options, accessories

Simplistic Financial Computations for System Architects.

version: 1.3
March 11, 2015
SFCincome
## The Time Dimension

<table>
<thead>
<tr>
<th></th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
</tr>
</thead>
<tbody>
<tr>
<td>investments</td>
<td>100k$</td>
<td>400k$</td>
<td>500k$</td>
<td>100k$</td>
<td>100k$</td>
<td>60k$</td>
<td>20k$</td>
</tr>
<tr>
<td>sales volume (units)</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>material &amp; labour costs</td>
<td>-</td>
<td>-</td>
<td>40k$</td>
<td>200k$</td>
<td>400k$</td>
<td>600k$</td>
<td>600k$</td>
</tr>
<tr>
<td>income</td>
<td>-</td>
<td>-</td>
<td>100k$</td>
<td>500k$</td>
<td>1000k$</td>
<td>1500k$</td>
<td>1500k$</td>
</tr>
<tr>
<td>quarter profit (loss)</td>
<td>(100k$)</td>
<td>(400k$)</td>
<td>(440k$)</td>
<td>200k$</td>
<td>500k$</td>
<td>840k$</td>
<td>880k$</td>
</tr>
<tr>
<td>cumulative profit</td>
<td>(100k$)</td>
<td>(500k$)</td>
<td>(940k$)</td>
<td>(740k$)</td>
<td>(240k$)</td>
<td>600k$</td>
<td>1480k$</td>
</tr>
</tbody>
</table>

*cost price / unit = 20k*

*sales price / unit = 50k*

variable cost = sales volume * cost price / unit  
income = sales volume * sales price / unit  
quarterm profit = income - (investments + variable costs)
The “Hockey” Stick

Simplistic Financial Computations for System Architects.

version: 1.3
March 11, 2015
SFHockeyStick
What if ...?

- Profit: $1M$
- Loss: $-0.5M$
- Delay of 3 months

- Early more expensive product + follow-on
- Delay of 3 months
- Original model
Stacking Multiple Developments

Simplistic Financial Computations for System Architects.

version: 1.3
March 11, 2015
SFCmultipleDevelopments
Fashionable financial yardsticks

Return On Investments (ROI)

Net Present Value

Return On Net Assets (RONA)  leasing reduces assets, improves RONA

turnover / fte  outsourcing reduces headcount, improves this ratio

market ranking (share, growth)  "only numbers 1, 2 and 3 will be profitable"

R&D investment / sales  in high tech segments 10% or more

cash-flow  fast growing companies combine profits with negative cash-flow, risk of bankruptcy
Make a **business plan** for the mid to long-term future.

- determine business model
- determine investments, sales volume, sales price, and costs
- estimate the cash flow and accumulated profit
- include at least 3 releases or generations of systems
Abstract

Products and enterprises evolve over time. This presentation explores the impact of these changes on the system and on the business by making (small and simple) models of life cycle aspects.
Product Related Life Cycles

- Individual systems
  - Service
    - System production
      - Upgrades and options
        - System sales
          - System creation
            - Upgrades and options
              - Model creation
System Life Cycle

- System order
- Ordering components
- Manufacturing
- Shipping
- Installation
- using
- Add option
- Maintenance
- Upgrade
- using
- Sales
- Shipping
- Refurbishing
- Shipping
- Installation
- Secondary use
- Dispose

Local changes, e.g. accounts procedures
Approach to Life Cycle Modeling

| Identify potential life cycle changes and sources |
| Characterize time aspect of changes | how often how fast |
| Determine required effort | amount type |
| Determine impact of change on system and context | performance reliability |
| Analyse risks | business |
What May Change During the Life Cycle?

- business volume
- product mix
- product portfolio
- product attributes (e.g. price)
- customers
- personnel
- suppliers
- application, business processes
- et cetera

- www.homes4sale.com
- www.apple.com/itunes/
- www.amazon.com
- www.ebay.com
- www.shell.com
- www.stevens.edu
- www.nokia.com
- stock market
- insurance company
- local Dutch cheese shop
Simple Model of Data Sources of Changes

Legend:
- Automated data inputs: Green arrows
- Interoperability: Green dashed arrows
- Human inputs: Red dashed arrows
- Error prone: Red arrows
- ~3% error rate: Red dashed arrows
- Change request: Blue arrows
- Problem report: Blue arrows

Usage context:
- Other systems

System:
- Design
- Realization

Life cycle context:
- Requirements

Other systems

Design

Realization

Requirements

Human inputs

Error prone!

~3% error rate
Data Sources of Web Server

content preparation

content provider

data quality?

content

web server

client

shop configuration

e.g., staff, roles

system configuration

e.g. resource allocation

Modeling and Analysis: Life Cycle Models

version: 0.7
March 11, 2015
MALCwebServerChanges

Embedded Systems Innovation
Example Product Portfolio Change Books

new books per year

<table>
<thead>
<tr>
<th>Country</th>
<th>Sales 2005</th>
<th>Sales 1996</th>
</tr>
</thead>
<tbody>
<tr>
<td>UK (1)</td>
<td>206k</td>
<td>107k</td>
</tr>
<tr>
<td>USA (2)</td>
<td>172k</td>
<td>68k</td>
</tr>
<tr>
<td>China (3)</td>
<td>101k</td>
<td></td>
</tr>
<tr>
<td>India (21)</td>
<td>12k</td>
<td></td>
</tr>
</tbody>
</table>

source: http://en.wikipedia.org/wiki/Books_published_per_country_per_year

Product portfolio characteristics
selection depends on business life cycle changes determined by business characteristics

source: http://en.wikipedia.org/wiki/Long_tail

WH Smith
Amazon "long tail"
Example Customer Change

internet: broadband penetration

<table>
<thead>
<tr>
<th></th>
<th>Q1 '04</th>
<th>Q2 '04</th>
<th>growth in Q2 '04</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia Pacific total</td>
<td>48M</td>
<td>54M</td>
<td>12.8%</td>
</tr>
<tr>
<td>China</td>
<td>15M</td>
<td>19M</td>
<td>26.1%</td>
</tr>
<tr>
<td>India</td>
<td>87k</td>
<td>189k</td>
<td>116.8%</td>
</tr>
</tbody>
</table>


What is the expected growth of # customers?
What is the impact on system and infrastructure?
What is the impact on CRM (Customer Relation Management)?
What is the impact on customer, sales support staff?
How much time/effort is needed for content updates?
How much staff is needed?
What is the impact of errors in content updates?
How many errors can be expected?
What is the impact of content updates on server loads?
### Web Shop Content Change Effort

<table>
<thead>
<tr>
<th>prepare</th>
<th>prepare</th>
<th>prepare</th>
</tr>
</thead>
<tbody>
<tr>
<td>change 1</td>
<td>change 2</td>
<td>change n</td>
</tr>
</tbody>
</table>

- review input
- select info
- layout & cosmetics
- check-in

<table>
<thead>
<tr>
<th>verify</th>
<th>verify</th>
</tr>
</thead>
<tbody>
<tr>
<td>change 1</td>
<td>change n</td>
</tr>
</tbody>
</table>

- inspect source
- inspect result

**Commit changes**

\[
\text{effort}_{\text{changes}} = n_{\text{changes}} \times (t_{\text{prepare}} + t_{\text{verify}}) + t_{\text{commit}}
\]

\[
\#\text{fte} = \text{effort}_{\text{changes}} / \text{hours per day}
\]

<table>
<thead>
<tr>
<th>(n_{\text{changes}}) per day</th>
<th>10</th>
<th>100</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\text{effort}_{\text{changes}})</td>
<td>1 uur</td>
<td>10 uur</td>
<td>100 uur</td>
</tr>
<tr>
<td>#\text{fte}</td>
<td>0.1</td>
<td>1</td>
<td>12</td>
</tr>
</tbody>
</table>

**Example Calculations**

\[
\begin{align*}
\text{t}_{\text{prepare}} &= 4 \text{ min} \\
\text{t}_{\text{verify}} &= 2 \text{ min} \\
\text{t}_{\text{commit}} &= 1 \text{ min}
\end{align*}
\]

**Hours per day**

\[
\text{hours per day} = 8 \text{ hours}
\]
Example of Client Level Changes

- up-to-date information: bestsellers, what other customers are looking at right now
catalogue entries
main access through search
personalization

- other advertisements

- standard boilerplate

- styling: frequently updated, fashion!
snapshot of www.amazon.com
Example of Time Scale Model for Changes

- **3 months**: Problem response
- **1 year**: Clinical prototype, Procedural change
- **10 years**: Workstation useful life
  - **MR scanner useful life**
  - **New generation of magnets, gradients, detectors**
  - **Commodity hardware and software**
  - **Minor SW release**
  - **Major SW release**
What is the security model?
What is the impact on server loads?
What is the impact on staffing?
What is the impact of changes in staff?
What is the impact of changes on security?
new faults = average fault density * #changes

\[
#\text{errors} = \sum_{\text{faults}} f(\text{severity, hit probability, detection probability})
\]

<table>
<thead>
<tr>
<th>severity</th>
<th>hit probability</th>
<th>detection probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jansen iso</td>
<td>low</td>
<td>high</td>
</tr>
<tr>
<td>Janssen operator iso</td>
<td>high</td>
<td>high</td>
</tr>
</tbody>
</table>
Analyze the evolution during the lifecycle.

- identify sources of change in customer context, life cycle context, and technology
- determine per change the expected rate of change and the required response time to the change
Simplistic Financial Computations

Product Margin = Sales Price - Cost

Margin per product. The margin over the sales volume, must cover the fixed costs, and generate profit transportation, insurance, royalties per product, ...

Cost per product, excluding fixed costs purchase price of components may cover development cost of supplier

Profit as function of sales volume

Hockey stick and scenarios

intentionally left blank

Summary Module Architectural Reasoning Business and Life Cycle
215 Gerrit Muller
Life Cycle

Multiple Life Cycles

- Individual systems
- Service
- System production
  - Upgrades and options
  - Production
- System sales
  - Upgrades and options
  - Sales

System Life Cycle

- System creation
- System production
- System sales
- Service
- Upgrades and options
  - Production
  - Sales
- Disposal

System Life Cycle:

- System order
  - Ordering components
  - Manufacturing
  - Shipping
  - Installation
  - Upgrade
  - Using
  - Add option
  - Maintenance
  - Upgrade
  - Using
  - Local changes, e.g., accounts procedures
  - Sales
  - Shipping
  - Refurbishing
  - Installation
  - Secondary use
  - Dispose

Analyze Frequency, Response Need, and Impact

- Identify potential life cycle changes and sources
- Characterize time aspect of changes
  - How often
  - How fast
- Determine required effort
  - Amount
  - Type
- Determine impact of change on system and context
  - Performance
  - Reliability
- Analyze risks
  - Business

Logarithmic Axis of Change Frequency

- Problem response
  - 3 months
- Clinical prototype
  - 1 year
- Procedural change
- Legislation change
  - 10 years

- Workstation useful life
- MR scanner useful life

- Commodity hardware and software
- New generation of magnets, gradients, detectors
Abstract

This module provides methods and techniques to integrate insights across views. Lines and Threads of reasoning form the main framework.
Abstract

Many stakeholder concerns can be specified in terms of qualities. These qualities can be viewed from all 5 “CAFCD” viewpoints. In this way qualities can be used to relate the views to each other. The meaning of qualities for the different views is described. A checklist of qualities is provided as a means for architecting. All qualities in the checklist are described briefly.
Quality needs as generic integrating concepts

- Customer objectives
- Application
- Functional
- Conceptual
- Realization

Qualities as Integrating Needles

Version: 1.3
March 11, 2015
QNneedles
## Security as example through all views

<table>
<thead>
<tr>
<th>C (Customer) objectives</th>
<th>A (Application)</th>
<th>F (Functional)</th>
<th>C (Conceptual)</th>
<th>R (Realization)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sensitive information</td>
<td>selection</td>
<td>functions for</td>
<td>cryptography</td>
<td>specific</td>
</tr>
<tr>
<td></td>
<td>classification</td>
<td>administration</td>
<td>firewall</td>
<td>algorithms</td>
</tr>
<tr>
<td></td>
<td>people</td>
<td>authentication</td>
<td>security zones</td>
<td>interfaces</td>
</tr>
<tr>
<td></td>
<td>information</td>
<td>intrusion</td>
<td>authentication</td>
<td>libraries</td>
</tr>
<tr>
<td></td>
<td>authentication</td>
<td>detection</td>
<td>registry</td>
<td>servers</td>
</tr>
<tr>
<td></td>
<td>badges</td>
<td>logging</td>
<td>logging</td>
<td>storage</td>
</tr>
<tr>
<td></td>
<td>passwords</td>
<td></td>
<td></td>
<td>protocols</td>
</tr>
<tr>
<td></td>
<td>locks / walls</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>guards</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>administrators</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>trusted</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>not trusted</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>desired characteristics, specifications &amp; mechanisms</td>
<td>missing</td>
<td>holes between</td>
<td>bugs</td>
<td></td>
</tr>
<tr>
<td></td>
<td>functionality</td>
<td>concepts</td>
<td>buffer overflow</td>
<td></td>
</tr>
<tr>
<td></td>
<td>quantification</td>
<td></td>
<td>non encrypted</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>storage</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>poor exception</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>handling</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>social contacts</td>
<td>missing</td>
<td>holes between</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>open passwords</td>
<td>concepts</td>
<td>buffer overflow</td>
<td></td>
</tr>
<tr>
<td></td>
<td>blackmail</td>
<td></td>
<td>non encrypted</td>
<td></td>
</tr>
<tr>
<td></td>
<td>burglary</td>
<td></td>
<td>storage</td>
<td></td>
</tr>
<tr>
<td></td>
<td>fraud</td>
<td></td>
<td>poor exception</td>
<td></td>
</tr>
<tr>
<td></td>
<td>unworkable</td>
<td></td>
<td>handling</td>
<td></td>
</tr>
<tr>
<td>procedures</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Qualities as Integrating Needles

220 Gerrit Muller
Quality Checklist

**usable**
- usability
- attractiveness
- responsiveness
- image quality
- wearability
- storability
- transportability

**dependable**
- safety
- security
- reliability
- robustness
- integrity
- availability

**effective**
- throughput or productivity

**interoperable**
- connectivity
- 3rd party extendible

**liable**
- liability
- testability
- traceability
- standards compliance

**efficient**
- resource utilization
- cost of ownership

**serviceable**
- serviceability
- configurability
- installability

**future proof**
- evolvability
- portability
- upgradeability
- extendibility
- maintainability

**down to earth attributes**
- cost price
- power consumption
- consumption rate
  - (water, air, chemicals, et cetera)
- size, weight
- accuracy

**ecological**
- ecological footprint
- contamination
- noise
- disposability

**manufacturability**
- logistics flexibility
- lead time

**maneuverability**
- logistics flexibility
- lead time

**logistics friendly**
- manufactureability
- logistics flexibility
- lead time

**consistent**
- reproducibility
- predictability

**reproducibility**
- predictability
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.
Overview of the reasoning approach

1. select starting point:
   - actual dominant need or problem

2. create insight:
   - submethod in one of CAFCR views
   - qualities checklist

3. deepen insight via facts:
   - via tests, measurements, simulations
   - story telling

4. broaden insight via questions:
   - why
   - what
   - how

5. define and extend the thread:
   - what is the most important / valuable
   - what is the most critical / sensitive
   - look for the conflicts and tension

continuously

consolidate in simple models
communicate to stakeholders
refactor documentation
From starting point to insight

step 1 starting point

- Customer objectives
- Application
- Functional
- Conceptual
- Realization

slow response
Creating Insight

Customer objectives
Application
Functional
Conceptual
Realization

performance
response
time model

step 2 creating insight
Deepening Insight

Threads of Reasoning
226    Gerrit Muller

version: 2.4
March 11, 2015
TORdeepeningInsight
Broadening Insight

Customer objectives

Application

Functional

Conceptual

Realization

why?

what?

how?

why?

what?

how?

why?

what?

how?

step 4 broadening insight
Problem identification and articulation

**need and problem selection criterions**

- **Customer objectives**
- **Application**
- **Functional**
- **Conceptual**
- **Realization**

**Threads of Reasoning**

- important
- valuable
- critical
- difficult
- sensitive
- vulnerable

*definition in terms of tension*

- throughput
- cost
- safety

- high performance sensor
- high speed moves

version: 2.4
March 11, 2015
Iteration during the analysis

- Problem
- Objective criteria
- Objective ranking
- Intuitive ranking
- Architect intuition
- Detect mismatch
- Improve solution
- Adjust intuition
- Improve criteria
- Improved problem understanding
- Improved solution understanding
- Solution
Thread of related issues
Abstract

The medical imaging workstation case is introduced. An architecting method based on the CAFCR viewpoints is explained, consisting of 4 elements:

• the CAFCR viewpoints
• qualities as integrating needles
• story telling
• threads of reasoning

A thread of reasoning is build up in steps, based on this case. The underlying reasoning is explained.
Easyvision serving three URF examination rooms

URF-systems  EasyVision: Medical Imaging Workstation

typical clinical image (intestines)
X-ray rooms from examination to reading around 1990

Threads of reasoning illustrated by medical imaging case

version: 0
March 11, 2015
XRayRoomsOld
X-ray rooms with Easyvision applied as printserver

Examination Room

Control Room

Corridor or closet

Reading Room

X-ray source

detector

printer

light box
Comparison screen copy versus optimized film

old: screen copy

new: SW formatting

20 to 50% less film needed

Threads of reasoning illustrated by medical imaging case

version: 0
March 11, 2015
MedicalImagingFilmComparison
Challenges for product creation

product policy:
standard HW
SW "only"
40 MHz CPU
64 MByte memory
10 MBit/s ethernet
1 GByte disk

ca 1 film / minute
film = 4k*5k pixels
subsecond retrieve
screen = 1k*1k

print throughput
view response time

image quality
image processing

view response time

threads of reasoning illustrated by medical imaging case

237 Gerrit Muller
Top level decomposition

![Diagram of top level decomposition with SW and HW sections. SW includes tools, applications, framework, libraries, and operating system. HW includes network, laser printer, workstation, desk, cabinet, remote control, optical disc, and legend with make and buy options.]
CAFCR viewpoints

**What** does Customer need in Product and **Why**?

- **C**ustomer objectives
- **A**pplication
- **F**unctional
- **C**onceptual
- **R**ealization

**Product How**

- drives, justifies, needs
- enables, supports

Threads of reasoning illustrated by medical imaging case

version: 0
March 11, 2015
CAFCRannotated

Gerrit Muller

239
Quality needs as generic integrating concepts

Threads of reasoning illustrated by medical imaging case

240  Gerrit Muller
From story to design

**Customer What**
- Customer objectives

**Customer How**
- Application

**Product What**
- Functional
- Conceptual

**Product How**
- Realization

- "What does Customer need in Product and Why?"

- Market vision
- A priori solution knowledge

- Story
  - Analyze design

- Case
  - Analyze design

- Design

Threads of reasoning illustrated by medical imaging case

version: 0
March 11, 2015
SHTfromStoryToDesign
Chronology of Easyvision RF R1 development

1991
- basic application
- toolboxes
- 100 kloc
- interactive viewing

1992
- performance problems
- IQ problems

1993
- Easyvision RF integrated product
- 360 kloc
- print server + communication + interactive viewing

marketing opinion:
"All the functionality is available, we only have to provide a clinical UI"
Thread of reasoning based on efficiency-quality tension

Customer objectives
- time efficient
- diagnostic quality
- safety (liability)

Specification issues
- system response
- system throughput
- image quality

Concepts
- resource management
- processor, memory
- internal logistics
- concurrency, processes

Design space
- concurrency, processes

Threads of reasoning illustrated by medical imaging case

Gerrit Muller

version: 0
March 11, 2015
MITORthread
Technology innovations

- standard UNIX based workstation
- full SW implementation, more flexible
- object oriented design and implementation (Objective-C)
- graphical User Interface, with windows, mouse etcetera
- call back scheduling, fine-grained notification
- data base engine, fast, reliable and robust
- extensive set of toolboxes
- property based configuration
- multiple coordinate spaces
Thread of reasoning; introvert phase

Introvert view: cost and impact of new technologies

Philips operational view (manufacturing, service, sales)

- **C**ustomer objectives
  - useable
  - efficient

- **A**pplication
  - response time

- **F**unctional
  - purchase price

- **C**onceptual
  - SW only
  - OO design

- **R**ealization
  - memory use

Introvert view: cost and impact of new technologies

Philips operational view (manufacturing, service, sales)
Threads of reasoning illustrated by medical imaging case

version: 0
March 11, 2015
MSmemoryZeroMeasurement
Solution of memory performance problem

Threads of reasoning illustrated by medical imaging case

Threads of reasoning illustrated by medical imaging case

Version: 0
March 11, 2015
MSmemoryUsageReduction
Visualization memory use per process

Threads of reasoning illustrated by medical imaging case

version: 0
March 11, 2015
MSmemoryBudget
Typical case URF examination

3 examination rooms connected to 1 medical imaging workstation + printer

examination room: average 4 interleaved examinations / hour

image production: 20 $1024^2$ 8 bit images per examination

film production: 3 films of 4k*5k pixels each

high quality output (bi-cubic interpolation)
Thread of reasoning; phase 2

Philips operational view
(manufacturing, service, sales)

How to measure memory, how much is needed?
from introvert to extrovert
Radiologist workspots and activities

- Examination Room
- dictation room
- light-box
- auto-loader
- dictation room

supervision of the examination
view and diagnose, dictate report
verify and authorise report

activities of the radiologist
Diagnosis in tens of seconds

Films loaded by clinical personnel during the day are automatically loaded into a light-box. During the day, new films are loaded by clinical personnel. During the day, a user looks at images, moves the head forward/backward, moves the head or eyes left/right/up/down, and zooms in. The user mumbles a few Latin words or clinical codes in the recorder and then presses the next button. The process takes tens of seconds. The user selects images, pans images, and generates a report.
Rendered images at different destinations

Screen:
- low resolution
- fast response

Film:
- high resolution
- high throughput

Network:
- medium resolution
- high throughput
SW Process structure 1991

- Remote systems and users
- Communication
- User interface
- User
- Data base
- UI devices
- Optical storage
- Print
- Optical disk drive
- Printer
- Network
- Disk drive
- Export
- Import

Legend:
- Software process
- User control
- Control and data flow
- Associated hardware
- System monitor
- Unix daemons
Threads of reasoning illustrated by medical imaging case

255  Gerrit Muller
Print server is based on banding

Threads of reasoning illustrated by medical imaging case
Gerrit Muller
version: 0
March 11, 2015
MICVbanding
Server CPU load

- Remote systems and users
- Communication
- Database
- Print
- Printer
- Serving one examination room
  - CPU time available for interactive viewing
    - 50 s/exam
    - 210 s/exam
- Serving 3 examination rooms
  - Import
    - 2.5 min/exam
  - Print
    - 10.5 min/exam
- Margin
  - 2 min

Threads of reasoning illustrated by medical imaging case

Version: 0
March 11, 2015
MiCVServerCPUload
Radiologists diagnose from film, throughput is important

Extrovert view shows conceptual and realization gaps!
Image quality and safety problem

Threads of reasoning illustrated by medical imaging case

version: 0
March 11, 2015
MITOR\text{false\Contouring}
Presentation pipeline for X-ray images

- Image from database
- Spatial enhancement
  - Bi-linear
  - Bi-cubic
- Interpolate
- Look up table
  - Invert
  - Contrast / brightness
- Graphics merge
- Colour LUT
- Monitor

Legend:
- SW
- HW
Threads of reasoning illustrated by medical imaging case
261   Gerrit Muller

version: 0
March 11, 2015
MICVwysiwyg
Safety problem

for user readability the font-size was determined "intelligently"; causing a dangerous mismatch between text and image

URF monitor output: fixed size letters at fixed grid

EV output: scaleable fonts in graphics overlay
Threads of reasoning illustrated by medical imaging case

263  Gerrit Muller

Version: 0  March 11, 2015  MITORthread40

Philips operational view (manufacturing, service, sales)

from extrovert diagnostic quality, via image quality, algorithms and load, to extrovert throughput
cost revisited in context of clinical needs and realization constraints; note: original threads are significantly simplified
Overview of architecting method

**method outline**

**framework**
- **Customer objectives**
- **Application**
- **Functional**
- **Conceptual**
- **Realization**

**submethods**
- + key drivers
- + value chain
- + business models
- + supplier map
- + stakeholders
- + context diagram
- + entity relationship models
- + dynamic models
- + use case
- + commercial, logistics, decompositions
- + mapping technical functions and several more
- + construction decomposition
- + functional decomposition
- + information model and many more

**integration via qualities**

**explore specific details**

**reasoning**

Threads of reasoning illustrated by medical imaging case

265  Gerrit Muller
Exercise Threads of Reasoning

1. Select 3..5 most important needs and concerns
2. Select 3..5 most important specification issues
3. Select 3..5 most critical design aspects
4. Select 3..5 most critical life cycle issues
5. Show relations: positive, negative
6. Transform into elevator pitch

Customer objectives
Application Functional Conceptual Realization
Life cycle
Integration via Qualities

Qualities Connect all Views

Look Positive and Negative

Many, Many Qualities

usable
usability
attractiveness
responsiveness
image quality
wearability
storability
transportability
dependable
safety
security
reliability
robustness
integrity
availability
effective
throughput or productivity
interoperable
connectivity
3rd party extendible
liable
liability
liveliness
traceability
standards compliance
efficient
resource utilization
cost of ownership
consistent
reproducibility
predictability
serviceable
serviceability
configurability
installability
future proof
evolvability
upgradeability
extendibility
maintainability
logistics friendly
manufacturability
logistics flexibility
lead time
ecological
ecological footprint
contamination
noise
disposability
down to earth attributes
cost price
power consumption
consumption rate
(water, air, chemicals, etc cetera)
size, weight accuracy

Summary Module Architectural Reasoning Threads and Integration
267 Gerrit Muller
Threads of Reasoning

Diverge, Converge, Zoom-in, Zoom-out

1. select starting point:
   - actual dominant need or problem
2. create insight:
   - submethod in one of CAFCR views
   - qualities checklist
3. deepen insight via facts:
   - via tests, measurements, simulations
   - story telling
4. broaden insight via questions:
   - why
   - what
   - how
5. define and extend the thread:
   - what is the most important / valuable
   - what is the most critical / sensitive
   - look for the conflicts and tension

identify most relevant issues

C customer objectives
A application
F functional
C conceptual
R realization

important
valuable
critical
difficult
valuable
sensitive
vulnerable

need and problem selection criteria

throughput
high performance sensor

cost
high speed moves

safety
definition in terms of tension

All Issues are Interrelated

Reconstruct the “Big Picture”

summary module architectural reasoning threads and integration

Gerrit Muller

Version: 0
March 11, 2015
Abstract

This module discusses modeling, especially aspects such as credibility, working range, and accuracy.
Abstract

We make models to facilitate decision making. These decisions range from business decisions, such as Service Level Agreements, to requirements, and to detailed design decisions. The space of decisions is huge and heterogeneous. The proposed modeling approach is to use multiple small and simple models. In this paper we discuss how to reason by means of multiple models.
How to use multiple models to facilitate decisions? 
How to get from many fragments to integral insight? 
How many models do we need? 
At what quality and complexity levels?
Example Graph for Web Shop

Usage Context:
- Enterprise & Users
  - Customer Interest
  - Customer Behavior
  - Salary
  - Financial

Life Cycle Context:
- Initial Cost
- Maintenance Effort
- Changes

Resource Dimensioning:
- Running Cost
- Initial Cost

System Design:
- Transactions
- CPU Load
- Memory Load
- Storage Capacity

Black Box View:
- Load
- Response Time
- Throughput
- Information

Legend:
- Assumption
- Input e.g. Measurement
- Decision
- Model
Relations: Decisions, Models, Inputs and Assumptions

Modeling and Analysis: Reasoning Approach
Gerrit Muller

version: 1.0
March 11, 2015
MARErelations
Reasoning Approach

1. Explore usage context, life cycle context and system

   **top-down**

   t2. Determine main Threads-of-Reasoning
   t3. Make main Threads-of-Reasoning SMART
   t4. Identify "hottest" issues
   t5. Model hottest, non-obvious, issues

   **bottom-up**

   b2a. "Play" with models
   b2b. Investigate facts
   b2c. Identify assumptions
   b3. Model significant, non-obvious, issues

   **learn**

6. Capture overview, results and decisions

7. Iterate and validate

*all steps time-boxed between 1 hour and a few days*

early in project

later in project
Frequency of Assumptions, Decisions and Modeling

![Diagram showing frequency of assumptions, decisions, and modeling]

- **Implicit (trivial?)**
- **Explicit**
- **Try-outs**
- **Very simple**
- **Small**
- **Key**
- **Substantial**

Legend:
- **A**: Assumption
- **i**: Input e.g. measurement
- **d**: Decision
- **m**: Model

**Modeling and Analysis: Reasoning Approach**

Gerrit Muller

version: 1.0
March 11, 2015
MAREfrequency
Life Cycle of Models

most try out models never leave the desk or computer of the architect!
many small and simple models are used only once; some are re-used in next projects

substantial models capture core domain know how; they evolve often from project to project.
creation and evolution of intellectual property assets
Examples of Life Cycle of Models

understanding  exploration  optimization  verification

try out models

load/cost
function mix
load/cost peak impact
load/stress test suite

simple and small models

customer global distribution
integral load model
web server performance

substantial models (IP assets)

global customer demographics
webshop benchmark suite
Identify a chain of models needed to support architecture development.

- models are related horizontally in the CAFCR model (across views), as well as vertically within a view

- models have various levels of detail; detailed models tend to feed/support less detailed models

- per model
  - formulate its purpose
  - indicate the main quantities that play a role
Abstract

Models only get value when they are actively used. We will focus in this presentation on analysis aspects: accuracy, credibility, sensitivity, efficiency, robustness, reliability and scalability.
What Comes out of a Model

- Varying inputs
- Varying circumstances
- Varying design options
- Varying realizations
- Specification changes
  - And ripple through
  - Model
    - Working range
    - Accuracy
    - Credibility
    - Sensitivity
    - Robustness
    - Efficiency
    - Model applicability
    - Design quality
    - Specification feasibility

Life cycle:
- Change cases
- Specification changes and ripple through

Design:
- Understanding
- Exploration
- Optimization
- Verification

Model(s):
- Use cases
- Worst case exceptions
- Varying inputs
- Varying circumstances
- Varying design options
- Varying realizations
- Specification changes
  - And ripple through

Modeling and Analysis: Model Analysis
281 Gerrit Muller
Applicability of the Model

++ε₁

- ε₂

input

accuracy

credibility

 consultants

assumptions

facts

measurements

abstraction

model(s)

accuracy

credibility

working range

model realization

credibility

propagation

abstraction

usage context

specifications

designs

realizations

version: 1.0

March 11, 2015

MAANmodelApplicability
How to Determine Applicability

<table>
<thead>
<tr>
<th><strong>try out models</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>be aware of accuracy, credibility and working range</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>simple and small models</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. Estimate accuracy of results</strong></td>
</tr>
<tr>
<td>based on most significant inaccuracies of inputs and assumed model propagation behavior</td>
</tr>
<tr>
<td><strong>2. Identify top 3 credibility risks</strong></td>
</tr>
<tr>
<td>identify biggest uncertainties in inputs, abstractions and realization</td>
</tr>
<tr>
<td><strong>3. Identify relevant working range risks</strong></td>
</tr>
<tr>
<td>identify required (critical) working ranges and compare with model working range</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>substantial models</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>systematic analysis and documentation of accuracy, credibility and working range</td>
</tr>
</tbody>
</table>
Common Pitfalls

- Discrete events in continuous world
  - Discretization artefacts
    - E.g. Stepwise simulations

- (Too) Systematic input data
  - Random data show different behavior
    - E.g. Memory fragmentation

- Fragile model
  - Small model change results in large shift in results

- Self fulfilling prophecy
  - Price erosions + Cost increase (inflation) -> Bankruptcy
Worst Case Questions

Which design assumptions have a big impact on system performance?

What are the worst cases for these assumptions?

How does the system behave in the worst case?

a. poor performance within spec

b. poor performance not within spec

c. failure -> reliability issue
# FMEA-like Analysis Techniques

## (systematic) brainstorm

<table>
<thead>
<tr>
<th>safety hazard analysis</th>
<th>potential hazards</th>
<th>damage</th>
<th>measures</th>
</tr>
</thead>
<tbody>
<tr>
<td>reliability FMEA</td>
<td>failure modes</td>
<td>effects</td>
<td>measures</td>
</tr>
<tr>
<td>security</td>
<td>vulnerability risks</td>
<td>consequences</td>
<td>measures</td>
</tr>
<tr>
<td>maintainability</td>
<td>change cases</td>
<td>impact, effort, time</td>
<td>decisions</td>
</tr>
<tr>
<td>performance</td>
<td>worst cases</td>
<td>system behavior</td>
<td>decisions</td>
</tr>
</tbody>
</table>

## Analysis and Assessment

- probability
- severity
- propagation

## Improve

- spec, design, process, procedure, ...

## Measures

- effort
- time

## Systematic brainstorming and assessment

- probability
- severity
- propagation

## Brainstorm and Assessment

- analysis
- probability
- severity

## Brainstorm and Assessment

- analysis
- probability
- severity

## Brainstorming and Assessment

- analysis
- probability
- severity

## Brainstorming and Assessment

- analysis
- probability
- severity

## Brainstorming and Assessment

- analysis
- probability
- severity

## Brainstorming and Assessment

- analysis
- probability
- severity

## Brainstorming and Assessment

- analysis
- probability
- severity

## Brainstorming and Assessment

- analysis
- probability
- severity
Brainstorming Phases

wave 1: the obvious

wave 2: more of the same

wave 3: the exotic, but potentially important

don't stop too early with brainstorming!
Different Viewpoints for Analysis

usage context
- new product
  - e.g. WoW extension
- merger
- automated access

system
- new functions
- new interfaces
- new media
- new standards
- cache/memory trashing
- garbage collection
- critical sections
- local peak loads
- intermittent HW failure

life cycle context
- power failure
- network failure
- new SW release
- roll back to old SW release
Exercise Analysis of Models

Determine for a few models their **credibility**, **accuracy**, and **working range**.

- Identify top 3 credibility risks
  - identify biggest uncertainties in inputs, abstractions and realization
- Estimate accuracy of results; quantitative, e.g. order 1% or 50%
  - based on most significant inaccuracies of inputs and assumed model propagation behavior
- Identify relevant working range risks
  - identify required (critical) working ranges and compare with model working range
Many Light Models, few Substantial Models

Accuracy, Credibility, Working Range

+ $\varepsilon_1$ - $\varepsilon_2$

input accuracy credibility

abstraction

model(s)

facts

accuracy credibility working range

model realization credibility propagation

usage context specifications designs realizations
Abstract

This module provides various means to consolidate architectures.
Abstract

This presentation provides guidelines and means to capture architecture overviews. Main challenge is to maintain the overview across multiple views. Architecture Overview A3s One support multi-view. Another challenge is to make an overview accessible for a wide range of stakeholders. The architecture description should therefore be visualized such that it fits the mental model of the audience.
Maturing an Architecture Description

- sketch
- flip charts
- yellow notes
- photo
- fast
- low threshold
- changing rapidly
- volatile
- informal

- capture electronic
- Visio
- PowerPoint
- more effort
- evolving slowly
- non-volatile
- slightly more formal

- structure and order
- document A3
- major effort
- controlled change
- non-volatile
- formal

- engineer repository
Architecture Overview A3

simplified from http://www.gaudisite.nl/BorchesCookbookA3architectureOverview.pdf
A3s to Capture Architecture Overviews

- multiple related views
- digestable (size limitation)
- quantifications
- practical
- close to stakeholder experience

one topic per A3

capture "hot" topics

source: PhD thesis Daniel Borches http://doc.utwente.nl/75284/
**Example of SubSea A3 Architecture Overview**

**Workover operation; architecture overview**

* This A3 based on the work of SEMA participants: Martin Moberg, Tormod Strand, Vazgen Karlsen, and Damien Wee, and the master project paper by Dag Joostein Klever.

**Consolidating Architecture Overviews**

**March 11, 2015**

**SSMEoverviewA3**
Multiple Levels of A3s

Consolidating Architecture Overviews

Consolidating Architecture Overviews

Gerrit Muller

version: 0.2
March 11, 2015
IASAdiaboloA3
1.1 One of several prerequisites for architecture creative synthesis is the definition of **5-7 specific key drivers** that are critical for success, along with the rationale behind the selection of these items.

2.1. The essence of a system can be captured in about **10 models/views**.

2.2. A **diversity** of architecture descriptions and models is needed: languages, schemata and the degree of formalism.

2.3. The level of **formality** increases as we move closer to the implementation level.

from http://www.architectingforum.org/bestpractices.shtml
Capture your work done during the course, e.g. make photos of the flip charts.

Make a list of questions, assumptions, biggest uncertainties and unknowns

Make a list of lessons learned

Make a plan for the homework
Consolidating Architectures

Maturing, from Light to Heavy

- sketch
- capture electronic
- structure and order
- engineer repository

- fast
- low threshold
- changing rapidly
- volatile
- informal

- more effort
- evolving slowly
- non-volatile
- slightly more formal

- major effort
- controlled change
- non-volatile
- formal

Subsea A3

A3 Architecture Overview

- header
- key performance parameters

- dynamic behavior (functional model)
- visual aids
- decisions and considerations

- physical view

simplified from http://www.gaudiste.nl/BorchesCookbookA3architectureOverview.pdf

Multiple Levels of A3s

Summary Module Architectural Wrap-up
302     Gerrit Muller

version: 0.2
March 11, 2015
## Recommendations as Red Thread

### principles
- use feedback
- work incremental
- work evolutionary
- be explicit
- make issues tangible

### objectives
- support communication
- facilitate reasoning
- support decision making
- create
- maintain
- understanding
- insight
- overview

### recommendations
- Time-box
- Iterate
- Quantify early
- Measure and validate
- Multiple levels of abstraction
- (Simple) mathematical models
- Analysis of accuracy and credibility
- Multi-view
- System and its context
- Visualize

---

**Summary Module Architectural Wrap-up**

version: 0.2  
March 11, 2015  
MAO_recommendations
Abstract

This document described the homework assignment for the SEMA course. The homework is made and delivered incrementally, so that the teacher can provide feedback during the assignment.
Group Assignment

Submit each step to the teacher, and process feedback in the next step

Step 1. weeks 1..3
- Consolidate work of course in 20 slide presentation as baseline.
- Search for answers to the main questions, biggest uncertainties and unknowns, validate main assumptions.
- Elaborate most relevant models.
- Discuss your work with other stakeholders to collect more facts and figures and for early validation

Step 2. weeks 4..6
- Transform the presentation into a T-shape presentation
- Identify gaps in the “T”
- Make simple models to eliminate the gaps

Step 3 weeks 7..9
- Identify required changes in models made so far, due to increased insight;
- Change one of the models accordingly.
- Evolve the T-shape presentation (max 20 slides); the target audience of this presentation is your management.
- Present to company management
- Identify next models to be made, measurements to be done, or fact finding to take place.
- Update the presentation with feedback from management and a list of future work.
Write an individual reflection report after finishing the group assignment, answering the following questions:

What is the credibility and accuracy level (quantified, e.g. 1% or 50%) of the models so far and why?

In retrospect, formulate a problem statement that requires such modeling effort?

What would you do differently if you would have to do this again?

How are you going to apply this in practice?

preferred size 1 A4, max 2 A4.
Submission instructions

use for all deliverables the following conventions:
filename: SEMA <your name or team> <subject>).<version>.<extension>
  e.g. SEMA WOSteam presentation.2.doc
  or SEMA John Student individual report.1.docx

email to: <gerrit • muller@ gmail • com>
cc: <gunnarkb @ gmail • com>
subject: SEMA <subject>

"standard" file types preferred, e.g. pdf, jpg, doc, docx, xls, xlsx, ppt, pptx