Module Reasoning: Linking Business to Technology

by Gerrit Muller       Buskerud University College
                      e-mail: gaudisite@gmail.com
                      www.gaudisite.nl

Abstract

This module addresses *Threads of Reasoning* as a means to connect business and operational needs to design and technology choices.

The complete course MA 611™ is owned by TNO-ESI. To teach this course a license from Buskerud University College is required. This material is preliminary course material. The final material and course information can be found at: [www.esi.nl/cursus](http://www.esi.nl/cursus).

July 31, 2014
status:          preliminary
draft           version: 0.2
**goal of this module**

Be able to relate *Customer* and *Operational* objectives to design and technology choices.

Be able to provide rationale for design decisions.

**content of this module**

Key driver method and recommendations

Threads of reasoning approach

Example in Health Care domain

**exercise**

Key driver graph
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

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Define the scope specific.</td>
<td>in terms of <strong>stakeholder</strong> or <strong>market segments</strong></td>
</tr>
<tr>
<td>2. Acquire and analyze facts</td>
<td>extract <strong>facts</strong> from the <strong>product specification</strong> and ask <strong>why questions</strong> about the <strong>specification</strong> of existing products.</td>
</tr>
<tr>
<td>3. Build a graph of relations between drivers and requirements by means of brainstorming and discussions</td>
<td>where <strong>requirements</strong> may have <strong>multiple drivers</strong></td>
</tr>
<tr>
<td>4. Obtain feedback</td>
<td>discuss with <strong>customers</strong>, observe <strong>their reactions</strong></td>
</tr>
<tr>
<td>5. Iterate many times</td>
<td>increased understanding often triggers the <strong>move</strong> of issues from <strong>driver</strong> to <strong>requirement</strong> or vice versa and <strong>rephrasing</strong></td>
</tr>
</tbody>
</table>
Recommendation for the Definition of Key Drivers

- 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
Transformation of Key Drivers into Requirements

Customer

What

Customer objectives

Key (Customer) Drivers

Derived Application Drivers

Customer

How

Application

Product

What

Functional

What

Customer

How To

means

may be skipped or

articulated by several

intermediate steps

functions

interfaces

performance figures

goal

version: 0.2
July 31, 2014
REQfromDriverToRequirement
Abstract
A method of reasoning is described, which addresses cross-cutting issues. The basis is fast iteration in the problem and solution space.

A thread of reasoning is a set of highly relevant related issues, which are addressed by articulating the problem in terms of tension and analyzing it in the CAFCR framework.

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

slow response

C customer objectives
A Application
F Functional
C Conceptual
R Realization
Creating Insight

**Step 2: Creating Insight**

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

**Performance**

**Response time model**

**Step 2 creating insight**
Deepening Insight

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

**Story**
- specific needs
- step 3 deepening insight

**Simulations, test, measurements**
- specific facts
step 4 broadening insight
**Problem identification and articulation**

<table>
<thead>
<tr>
<th>Customer objectives</th>
<th>Application</th>
<th>Functional</th>
<th>Conceptual</th>
<th>Realization</th>
</tr>
</thead>
</table>

**need and problem selection criterions**

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

**definition in terms of tension**

- throughput
- cost
- safety

- high performance sensor
- high speed moves
Iteration during the analysis

Threads of Reasoning
©2006, Embedded Systems Institute
version: 2.4
July 31, 2014
TORanalysis/Iteration

©2006, Gerrit Muller
Thread of related issues

Customer objectives
Application
Functional
Conceptual
Realization

Threads of Reasoning
©2006, Embedded Systems Institute
16 ©2006, Gerrit Muller
Threads of reasoning illustrated by medical imaging case

by Gerrit Muller       Buskerud University College

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

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.

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.

version: 0
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

<table>
<thead>
<tr>
<th>Examination Room</th>
<th>Control Room</th>
<th>Corridor or closet</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Examination Room" /></td>
<td><img src="image2.png" alt="Control Room" /></td>
<td><img src="image3.png" alt="Corridor or closet" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Examination Room</th>
<th>Control Room</th>
<th>Reading Room</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Examination Room" /></td>
<td><img src="image2.png" alt="Control Room" /></td>
<td><img src="image4.png" alt="Reading Room" /></td>
</tr>
</tbody>
</table>

---

Threads of reasoning illustrated by medical imaging case
©2006, Embedded Systems Institute
©2006, Gerrit Muller

version: 0
July 31, 2014
XRaysRoomsOld
X-ray rooms with Easyvision applied as printserver
Comparison screen copy versus optimized film

old: screen copy

new: SW formatting

20 to 50% less film needed
Challenges for product creation

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

- **image quality**
- **image processing**

- **print throughput**

- **view response time**

- **tension**

- **ca 1 film / minute**
  - film = 4k*5k pixels

- **subsecond retrieve**
  - screen = 1k*1k

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

- **Threads of reasoning illustrated by medical imaging case**

  ©2006, Embedded Systems Institute

  Version: 0
  July 31, 2014
  IMIchallenge
## Top level decomposition

<table>
<thead>
<tr>
<th>tools</th>
<th>application</th>
<th>application</th>
<th>framework, libraries</th>
<th>operating system</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### SW

- optical disc
- workstation
- desk, cabinet
- network
- laser printer
- remote control

### HW

- make
- buy

**Legend**

 Threads of reasoning illustrated by medical imaging case
©2006, Embedded Systems Institute
©2006, Gerrit Muller

version: 0
July 31, 2014
IMIdecomposition
CAFCR viewpoints

What does Customer need in Product and Why?

Customer What

Customer How

Product What

Product How

C - Customer objectives
A - Application
F - Functional
C - Conceptual
R - Realization

drives, justifies, needs
enables, supports
Quality needles as generic integrating concepts

Threads of reasoning illustrated by medical imaging case

©2006, Embedded Systems Institute
©2006, Gerrit Muller
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

case

analyze design

design
Chronology of Easyvision RF R1 development

- **1991**: basic application toolboxes 100 kloc
- **1992**: performance problems
- **1993**: Easyvision RF integrated product 360 kloc
- **Interactive viewing**
- IQ problems
- 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

- Concepts
  - resource management
    - processor, memory
  - internal logistics
    - concurrency, processes
  - image quality
    - reinforcing
    - conflicting
  - image processing
    - algorithms
  - concurrency, processes
    - reinforcing
    - conflicting
  - design
  - space
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

Threads of reasoning illustrated by medical imaging case
©2006, Embedded Systems Institute
©2006, Gerrit Muller

version: 0
July 31, 2014
PMITORtechnologyInnovations
Thread of reasoning; introvert phase

Introvert view: cost and impact of new technologies

Philips operational view (manufacturing, service, sales)
Memory usage half way R1

![Diagram of memory usage](image)

- Total measured memory usage
  - OS
  - Code
  - Data
  - Bulk data
  - Fragmentation

- Performance
- Physical memory
- Paging to disk

---

Threads of reasoning illustrated by medical imaging case
©2006, Gerrit Muller
Solution of memory performance problem

Threads of reasoning illustrated by medical imaging case
©2006, Embedded Systems Institute
©2006, Gerrit Muller

version: 0
July 31, 2014
MSmemoryUsageReduction
Visualization memory use per process

- Shared libraries
- Communication
- Storage server
- Print server
- Other
- UNIX

Budget per process measured (left column)

<table>
<thead>
<tr>
<th>Component</th>
<th>Measured Budget</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>20 MByte</td>
</tr>
<tr>
<td>Code</td>
<td>20 MByte</td>
</tr>
</tbody>
</table>

Threads of reasoning illustrated by medical imaging case
©2006, Embedded Systems Institute
©2006, Gerrit Muller

version: 0
July 31, 2014
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

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
- Auto-loader
- Light-box
- Looks at images
- Moves head forward/backward
- Moves head or eyes left/right/up/down
- Zoom in
- Overview
- Mumbles a few Latin words or clinical codes in recorder
- Presses next button
- Image selection, panning
- Tens of seconds
- New films
- Old films
- Report
Rendered images at different destinations

**Screen:**
- low resolution
- fast response

**Film:**
- high resolution
- high throughput

**Network:**
- medium resolution
- high throughput
Threads of reasoning illustrated by medical imaging case
©2006, Embedded Systems Institute
40 ©2006, Gerrit Muller

version: 0
July 31, 2014
MITORsoftwareProcess1991
SW layers 1991

Legend:
- User Interface
- Application Functions
- Toolbox
- Operating System
- Hardware
- SW Infrastructure
- Connected System

Diagram:
- Medical Imaging R/F
- Print
- Store
- View
- Cluster
- Spool
- HCU
- Store
- Image
- Gfx
- UI
- DB
- PMS-net in
- PMS-net out
- SunOS, SunView
- Standard Sun workstation
- Desk, cabinets, cables, etc.
- RC interf
- HC interf
- DOR
- RC
- HC
- DOR driver
- 3M
- DSI

Threads of reasoning illustrated by medical imaging case
©2006, Embedded Systems Institute
©2006, Gerrit Muller

Version: 0
July 31, 2014
MITORswLayers1991
Print server is based on banding

Threads of reasoning illustrated by medical imaging case
©2006, Embedded Systems Institute
42  ©2006, Gerrit Muller

version: 0
July 31, 2014
MICVbanding
Server CPU load

- Remote systems and users
- Communication
- Database
- Print
- Printer
- Disk
- Import
- Print

3.5 CPU seconds per Mpixel output
2.5 CPU seconds per Mbyte input

Serving one examination room:
- CPU time available for interactive viewing
- 50 s/exam
- 210 s/exam
- 30%

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
©2006, Embedded Systems Institute
©2006, Gerrit Muller
version: 0
July 31, 2014
MICV server CPU load
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
©2006, Embedded Systems Institute
45 ©2006, Gerrit Muller
version: 0
July 31, 2014
MITORfalseContouring
Presentation pipeline for X-ray images

- Image from database
- Spatial enhancement
- Interpolate
- Look up table: invert, contrast / brightness
- Graphics merge
- Colour LUT
- Monitor

Threads of reasoning illustrated by medical imaging case

© 2006, Embedded Systems Institute

© 2006, Gerrit Muller
Threads of reasoning illustrated by medical imaging case

Image Quality expectation WYSIWYG

what you see at one work-spot is what you get at another work-spot

X-ray system

- Image generation
- Presentation

3rd party workstation

- Application processing
- Presentation

Easyvision

- Monitor
- Film
- Network, storage

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

**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

<table>
<thead>
<tr>
<th>Customer objectives</th>
<th>Application</th>
<th>Functional</th>
<th>Conceptual</th>
<th>Realization</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ key drivers</td>
<td>+ stakeholders and concerns</td>
<td>+ use case</td>
<td>+ construction decomposition</td>
<td></td>
</tr>
<tr>
<td></td>
<td>+ context diagram</td>
<td>+ commercial, logistics decompositions</td>
<td>+ functional decomposition + information model and many more</td>
<td></td>
</tr>
<tr>
<td></td>
<td>+ entity relationship models</td>
<td>+ mapping technical functions and several more</td>
<td>+ budget + benchmarking + performance analysis + safety analysis and many more</td>
<td></td>
</tr>
<tr>
<td></td>
<td>+ dynamic models</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Submethods

- + value chain
- + business models
- + supplier map
- + market vision
- + performance
- + safety
- + profit margin
- + standard workstation
- + CPU budget
- + memory budget
- + Moore's law
- + benchmarking
- + purchase price
- + CoO
- + render engine
- + purchase price
- + CoO

### Integration via Qualities

- Throughput
- Processing
- Library
- Diagnostic
- Quality
- Image
- Spec
- IQ
- GPU
- Flit
- Memory

### Explore Specific Details

- Market vision
- A priori solution know-how
- Use case
- Analyse design
- Detailed design

### Reasoning

Threads of reasoning illustrated by medical imaging case

©2006, Embedded Systems Institute

©2006, Gerrit Muller
Exercise

- Make a key driver graph
  - Use the key driver approach
  - Take the recommendations into account
Reflection on Exercise

+ Key drivers put requirements in broader perspective
+ Discussion creates shared understanding

~ The graph needs external feedback

- Are the key drivers really from the customer?
- Are the key drivers sharp enough?
## Conclusions

Key Driver graph connects customer objectives to system requirements

Threads of Reasoning connects Customer and Operational Objectives to design and technology choices

The overview is maintained by focusing on valuable, important, critical or sensitive aspect; Look for tensions!

## Techniques, Models, Heuristics of this module

Key driver graph

Thread of reasoning

Why, What and How

Tensions
The Boderc project contributed to Key drivers and Threads of Reasoning. Especially the work of

Lou Somers, Peter van den Bosch, Zhaouri Yuan (Océ),

Berry van der Wijst (Philips),

Adriaan van den Brand (Centric TSolve),

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

has been valuable.