Abstract

Today’s Smart systems are highly complex, due to the dynamic interactions between systems and the context, and the dynamic behavior within the system. The architecting challenge is to have overview facilitating reasoning, communication, and decision making. At the same time, details may disrupt expected behavior. Hence, architects and designers need a connection between overview and details. We will discuss application of architecture overviews in various complex systems.
“The problem is that we are attempting to build systems that are **beyond our ability to intellectually** manage.” (Nancy Leveson)

"Software failures are failures of **understanding**, and of **imagination**" 

"Really he was interested in how people see and understand systems—as he puts it, in the **visual representation of dynamic behavior**.” (Bret Victor)

How to Architect Smart Systems?

**Smart Systems:**
- interoperate with many systems
- data from everywhere
- “intelligence”, learning
- autonomous, automated, remote

**Dumb Stakeholders?**
- who can intellectually manage them?
- who can understand their construction?
- who can imagine their interactions?
- how to visualize dynamic behavior?

**Somers’ solution:**
- replace code by models
- use formal methods

**This presentation’s solution:**
- create Architecture Overviews
- in digestable nuggets
- static, dynamic, and qualities
- at various abstraction levels
The Problem: The Coming Software Apocalypse

The Solution: A3 Architecture Overviews

Principle: Three Main Views

Example: Visualizations

Principle: Multiple Abstraction Levels

Example: Dynamic A3s

The Challenges: Pitfalls and Dilemmas

Conclusions and Questions
The Solution: A3 Architecture Overviews

source: PhD thesis Daniel Borches http://doc.utwente.nl/75284/
dynamic behavior
quality attributes
visual aids
notes
legend
parts
choices

Conceptual Modeling H2 to Connect AO to details

version: 0
June 5, 2018
CMCAODa3ao
Main Principles of A3 Architecture Overviews

A3 Space limitation → show essentials

Show multiple views:
- Parts
- Dynamic behavior
- Quality attributes (and quantify!)

Visualize
- Use visual aids

Structure of A3s:
- multiple abstraction levels
The Problem: The Coming Software Apocalypse

The Solution: A3 Architecture Overviews

Principle: Three Main Views

Example: Visualizations

Principle: Multiple Abstraction Levels

Example: Dynamic A3s

The Challenges: Pitfalls and Dilemmas

Conclusions and Questions
From: A reference architecture for cooperative driving by Sagar Behere, Martin Törngren, De-Jiu Chen
Interaction Causing Emerging Dynamic Behavior

Conceptual Modeling H2 to Connect AO to details

Gerrit Muller

version: 0
June 5, 2018
CMCAODpartsDynamics
Prime Customer Interest: Key Performance

Prime system responsibility

Quality attributes

Prime interest of customer

Results in

Dynamic behavior

Functionality

Interact

Parts

Prime interest of organization

Throughput
Response Time
Accuracy
Image Quality
Reliability
Safety
Security

Sensitivity
Specificity

The Problem: The Coming Software Apocalypse

The Solution: A3 Architecture Overviews

Principle: Three Main Views

Example: Visualizations

Principle: Multiple Abstraction Levels

Example: Dynamic A3s

The Challenges: Pitfalls and Dilemmas

Conclusions and Questions
Example A3 "Transit to Operation"

Findings A3 for Validation

- **Too little knowledge** about the function and why it existed within the project
- **Not optimal market focus**, and story the project was based on
- **Easy and fast** method to collect, document and share information from peoples head
- Developing the function through **common understanding/agreement**
- We were able to document and agree on important statements like **needs, key drivers and market**
- Bridges development with **sales and marketing**, and experienced operators (**internal + external**)
- People will have to consider the **operational view** and not only the technical perspective
- Everybody was able to work with the same tool
  - Different focus
  - Cross-fertilization
- Low effort training and implementation costs
- Broader involvement of **stakeholders**
- The stakeholders saw advantages of **model based communication**
- Easy to provide **feedback**
- **Many new and improved requirements** were collected and reported
- Gives the developers a **clearer picture** of what to make, a good **overview** of the function
- Piece of the puzzle
- May not cover all paths to validation failure
The Problem: The Coming Software Apocalypse

The Solution: A3 Architecture Overviews

Principle: Three Main Views

Example: Visualizations

Principle: Multiple Abstraction Levels

Example: Dynamic A3s

The Challenges: Pitfalls and Dilemmas

Conclusions and Questions
Level of Abstraction Single System

- **Static system definition**
- **Monodisciplinary**
- **Number of details**
- **System requirements**
- **Multidisciplinary design**
- **Static system definition**
- **Monodisciplinary**

Conceptual Modeling H2 to Connect AO to details
From system to Product Family or Portfolio

Conceptual Modeling H2 to Connect AO to details

version: 0
June 5, 2018
DRAgmentidGrowth

Gerrit Muller
Product Family in Context

- Parts, connections, lines of code
- Multidisciplinary design
- Systems
- Stakeholders
- Enterprise
- Enterprise context

Conceptual Modeling H2 to Connect AO to details

17 Gerrit Muller
Capturing all information that is required for: logistics, manufacturing, legislation, maintenance, life cycle support,

Conceptual Modeling H2 to Connect AO to details
18 Gerrit Muller

version: 0
June 5, 2018
LAWFengineering
from needs and requirements to design: decomposition, interface definition, allocation, concept selection, technology choices

Design

Conceptual Modeling H2 to Connect AO to details

Version: 0

June 5, 2018

LAWFdesign

Gerrit Muller
Architecting: realization and design choices in context

Some context details are essential.

Some technical details are essential.

Conceptual Modeling H2 to Connect AO to details

version: 0
June 5, 2018
LAWFdiabolo
Architecture Overviews at Multiple Levels

Conceptual Modeling H2 to Connect AO to details

version: 0
June 5, 2018
IASAdiaboloA3
The Problem: The Coming Software Apocalypse

The Solution: A3 Architecture Overviews

Principle: Three Main Views

Example: Visualizations

Principle: Multiple Abstraction Levels

Example: Dynamic A3s

The Challenges: Pitfalls and Dilemmas

Conclusions and Questions
Example Dynamic A3s for Gas Turbines

Early Validation with Dynamic A3

- Detect 8 missing parameters (sensors, control settings, and mechanical components)
- Gave engineers “what” and “how”, and they started to ask “why”
- Finalize design and approval in only two revisions!

The Problem: The Coming Software Apocalypse

The Solution: A3 Architecture Overviews

Principle: Three Main Views

Example: Visualizations

Principle: Multiple Abstraction Levels

Example: Dynamic A3s

The Challenges: Pitfalls and Dilemmas

Conclusions and Questions
Typical Pitfalls and Dilemmas when Making A3AOs

- Lack of **context** understanding:
  - **customer** context
  - **life cycle** context
- Staying **superficial**
- Lack of **dynamic behavior** understanding
- Lack of **quality attributes** understanding
- **Short-term** orientation
- No **sense-of-urgency**
- Drowning in **multitude** of options
- Too **few perspectives**; one-dimensional thinking
The Problem: The Coming Software Apocalypse

The Solution: A3 Architecture Overviews

Principle: Three Main Views

Example: Visualizations

Principle: Multiple Abstraction Levels

Example: Dynamic A3s

The Challenges: Pitfalls and Dilemmas

Conclusions and Questions
Conclusions and Questions?

**A3AOs are:**
- practical
- powerful means to
  - create overview
  - cope with complexity

However, A3AOs require:
- architecting competence
- visualization skills
- conceptual modeling competence
- zoom in, zoom out agility
- many viewpoints capacity

**Questions**