Conceptual Modeling: How to Connect Architecture Overview and Design Details?

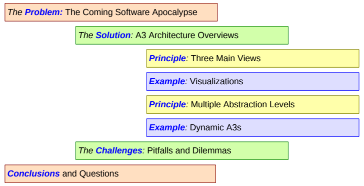
by *Gerrit Muller* USN-NISE, TNO-ESI
e-mail: `gaudisite@gmail.com`
`www.gaudisite.nl`

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.

Distribution

August 21, 2020
status: preliminary draft
version: 0



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 Coming Software Apocalypse?

“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)”

https://www.theatlantic.com/technology/archive/2017/09/saving-the-world-from-code/540393/

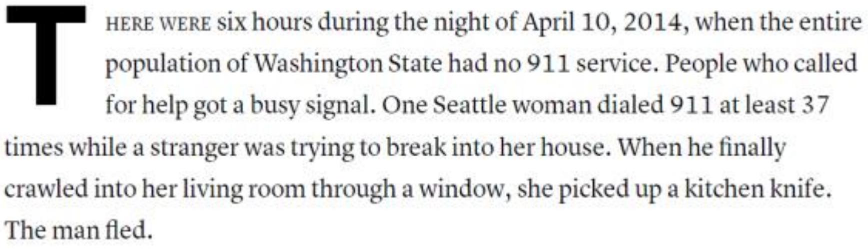
The Atlantic

The Coming Software Apocalypse

A small group of programmers wants to change how we code—before catastrophe strikes.

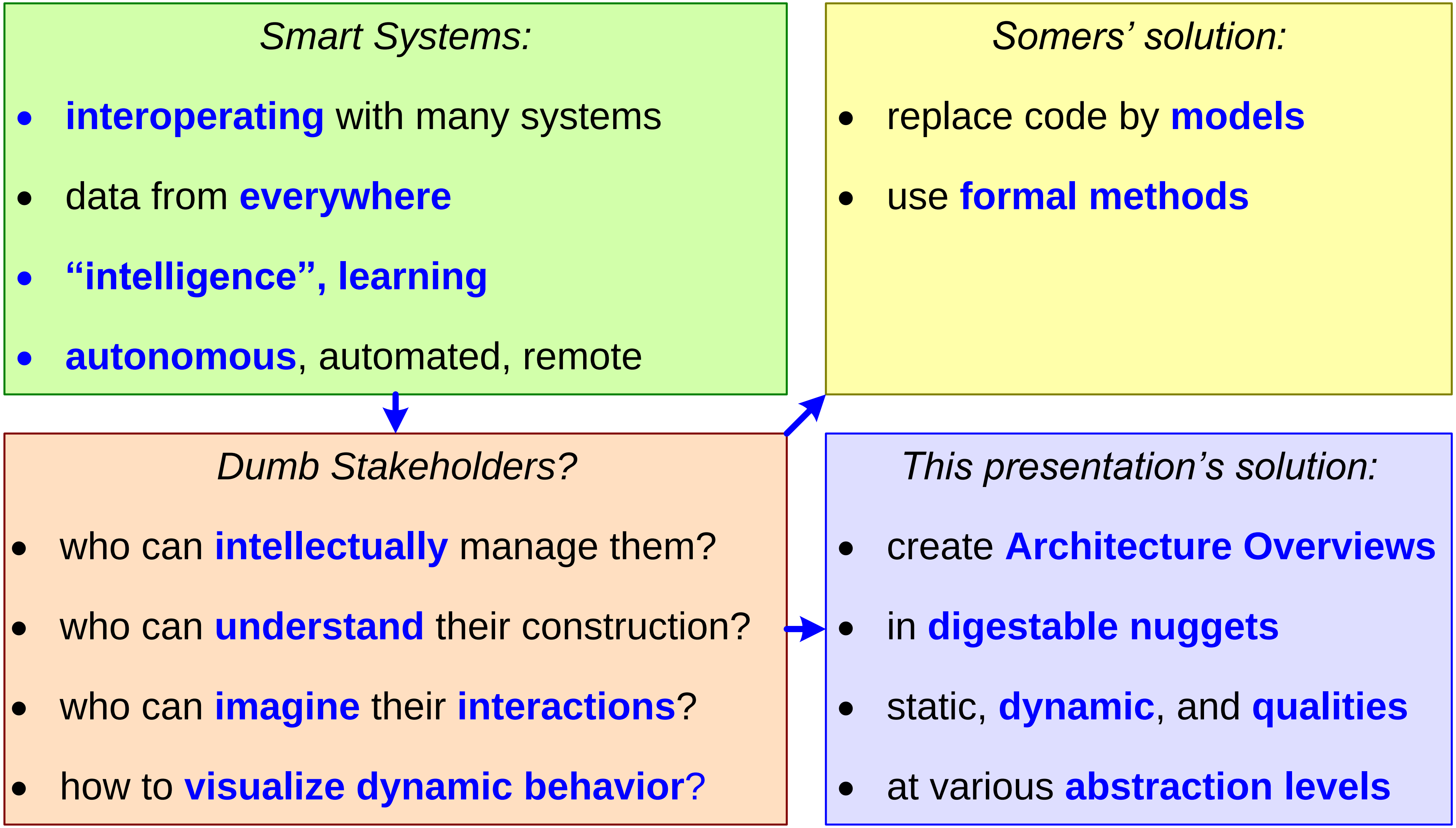
Lynn Scurfield

James Somers | Sep 26, 2017



THERE WERE six hours during the night of April 10, 2014, when the entire population of Washington State had no 911 service. People who called for help got a busy signal. One Seattle woman dialed 911 at least 37 times while a stranger was trying to break into her house. When he finally crawled into her living room through a window, she picked up a kitchen knife. The man fled.

How to Architect Smart Systems?



Smart Systems:

- **interoperating** with many systems
- data from **everywhere**
- **“intelligence”, learning**
- **autonomous**, automated, remote

Somers’ solution:

- replace code by **models**
- use **formal methods**

Dumb Stakeholders?

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

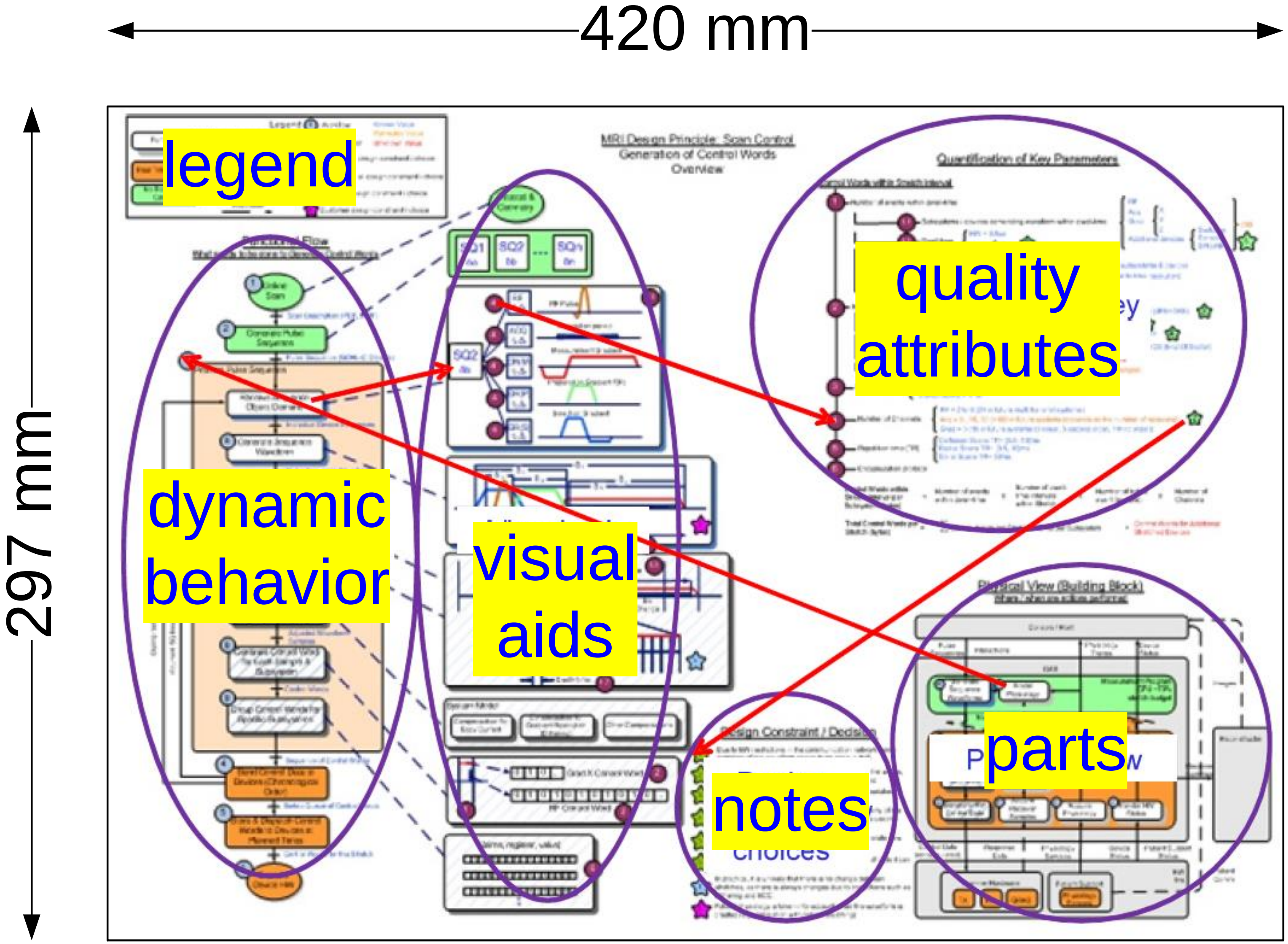
This presentation’s solution:

- create **Architecture Overviews**
- in **digestable nuggets**
- static, **dynamic**, and **qualities**
- at various **abstraction levels**

Figure of Contents™

- *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


420 mm
297 mm
legend
quality attributes
dynamic behavior
visual aids
parts
notes
MRI Design Principle: Scan Control
Generation of Control Words
Overview
Quantification of Key Parameters
Physical View (Building Block)
Design Constraint / Decision

source: PhD thesis Daniel Borches http://doc.utwente.nl/75284/

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

Parts + Dynamic Behavior + Quality Attributes

*The **Problem:*** The Coming Software Apocalypse

*The **Solution**:* A3 Architecture Overviews

***Principle**:* Three Main Views

***Example**:* Visualizations

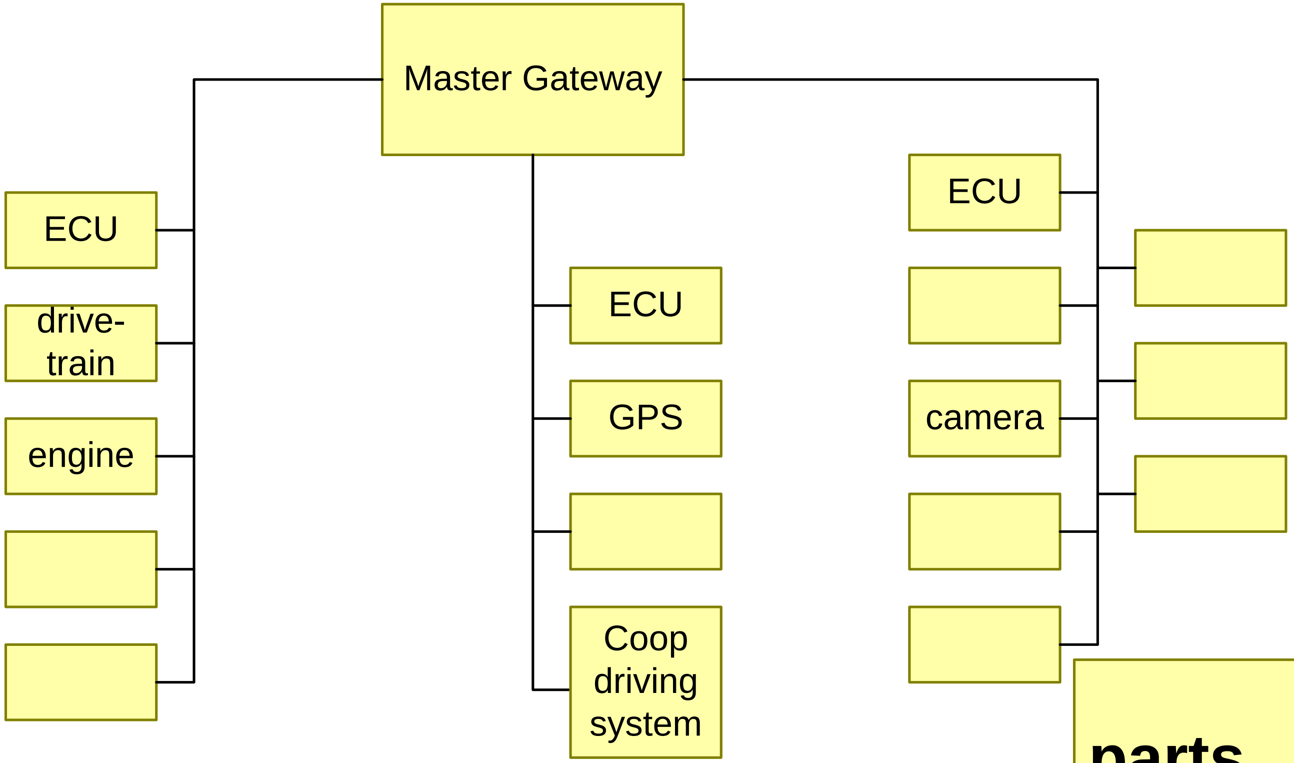
***Principle**:* Multiple Abstraction Levels

***Example**:* Dynamic A3s

*The **Challenges**:* Pitfalls and Dilemmas

Conclusions and Questions

Dominating Mindset: Parts (static)

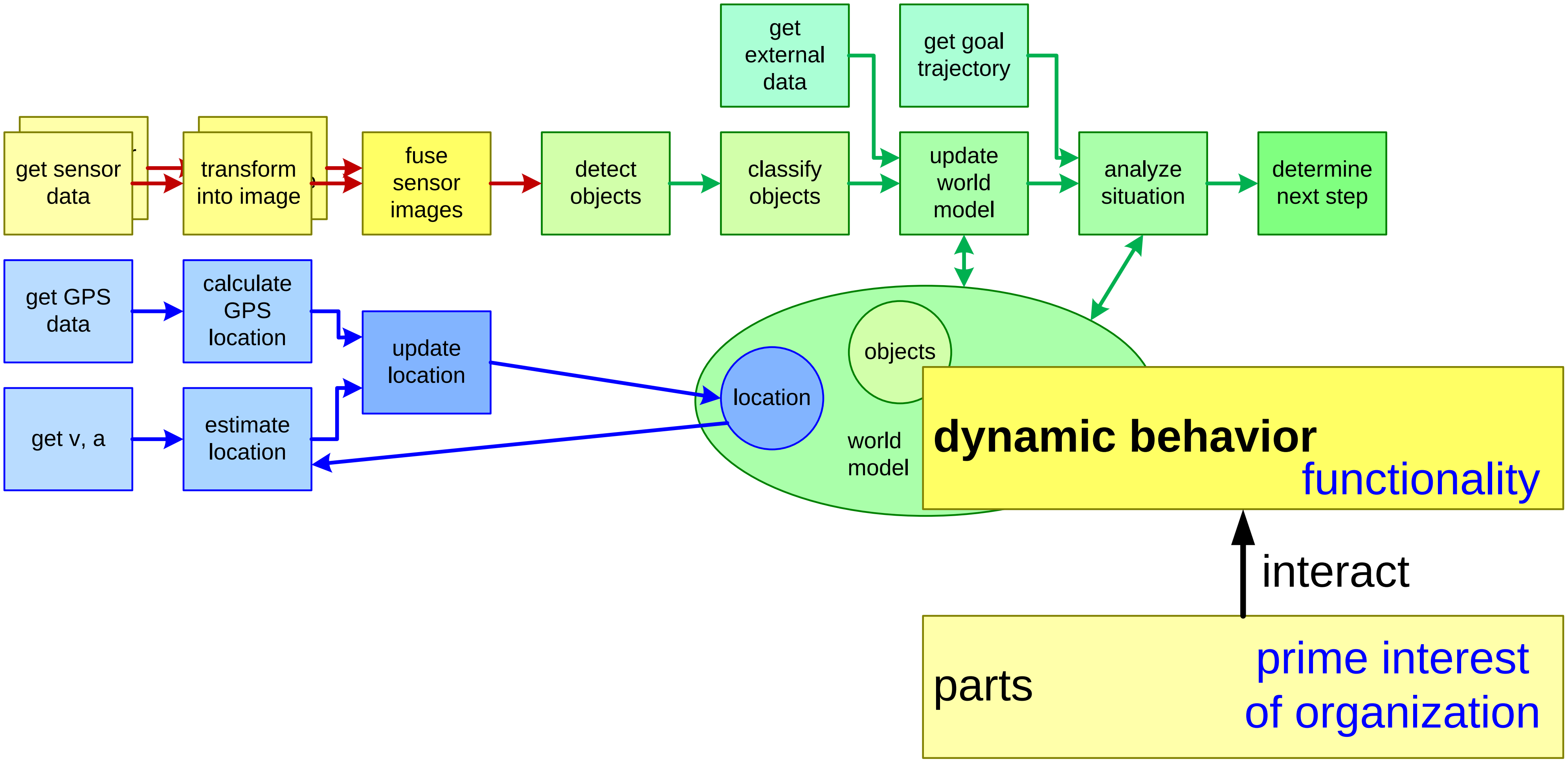


From: A reference architecture for cooperative driving by Sagar Behere, Martin Törngren, De-Jiu Chen

parts

prime interest of organization

Interaction Causing Emerging Dynamic Behavior



Prime Customer Interest: Key Performance

Throughput

Response Time

Accuracy

Image Quality

Reliability

Safety

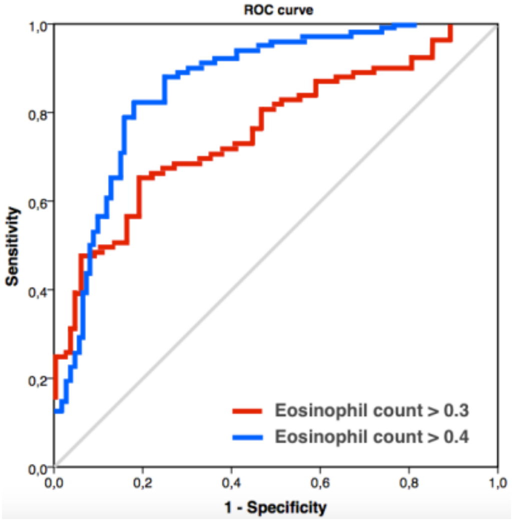
Security

Sensitivity

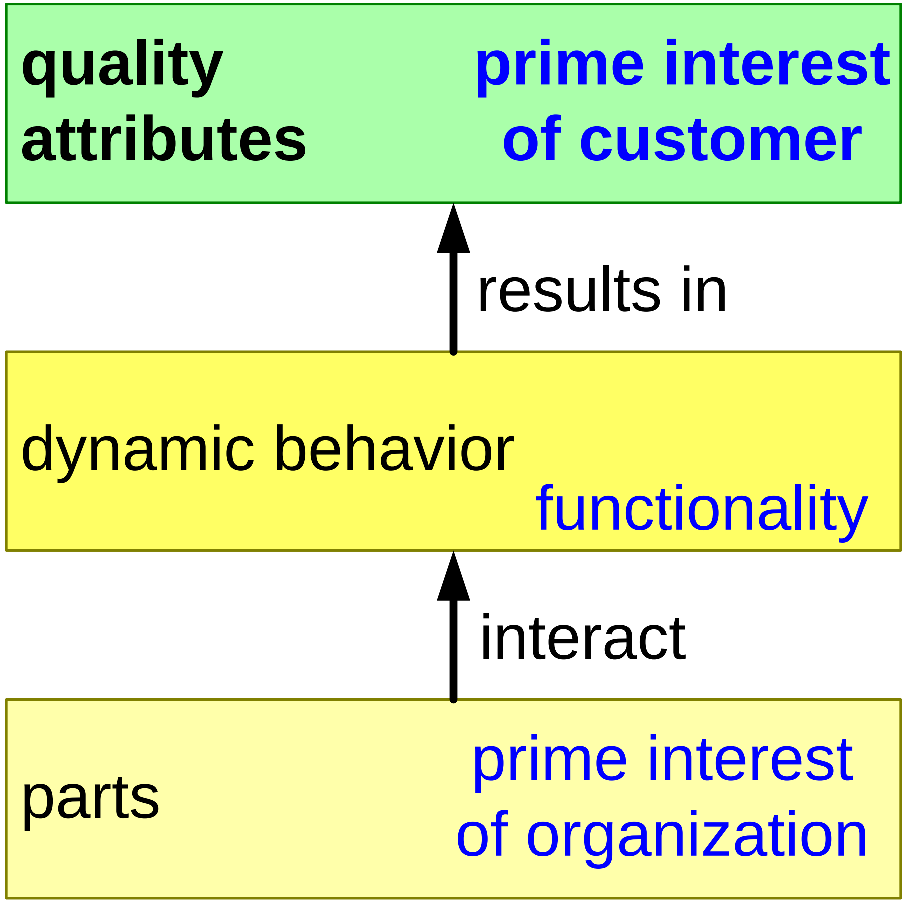
Specificity

...

prime system responsibility



eNose Sensitivity vs Specificity from:
https://www.breathcloud.org/wp-content/uploads/2017/07/Poster-ERS-2016.pdf



quality attributes **prime interest of customer**

results in

dynamic behavior functionality

interact

parts prime interest of organization

Example of Visualizations for Early Validation

The ***Problem:*** The Coming Software Apocalypse

The ***Solution****:* A3 Architecture Overviews

Principle: Three Main Views

Example*:* Visualizations

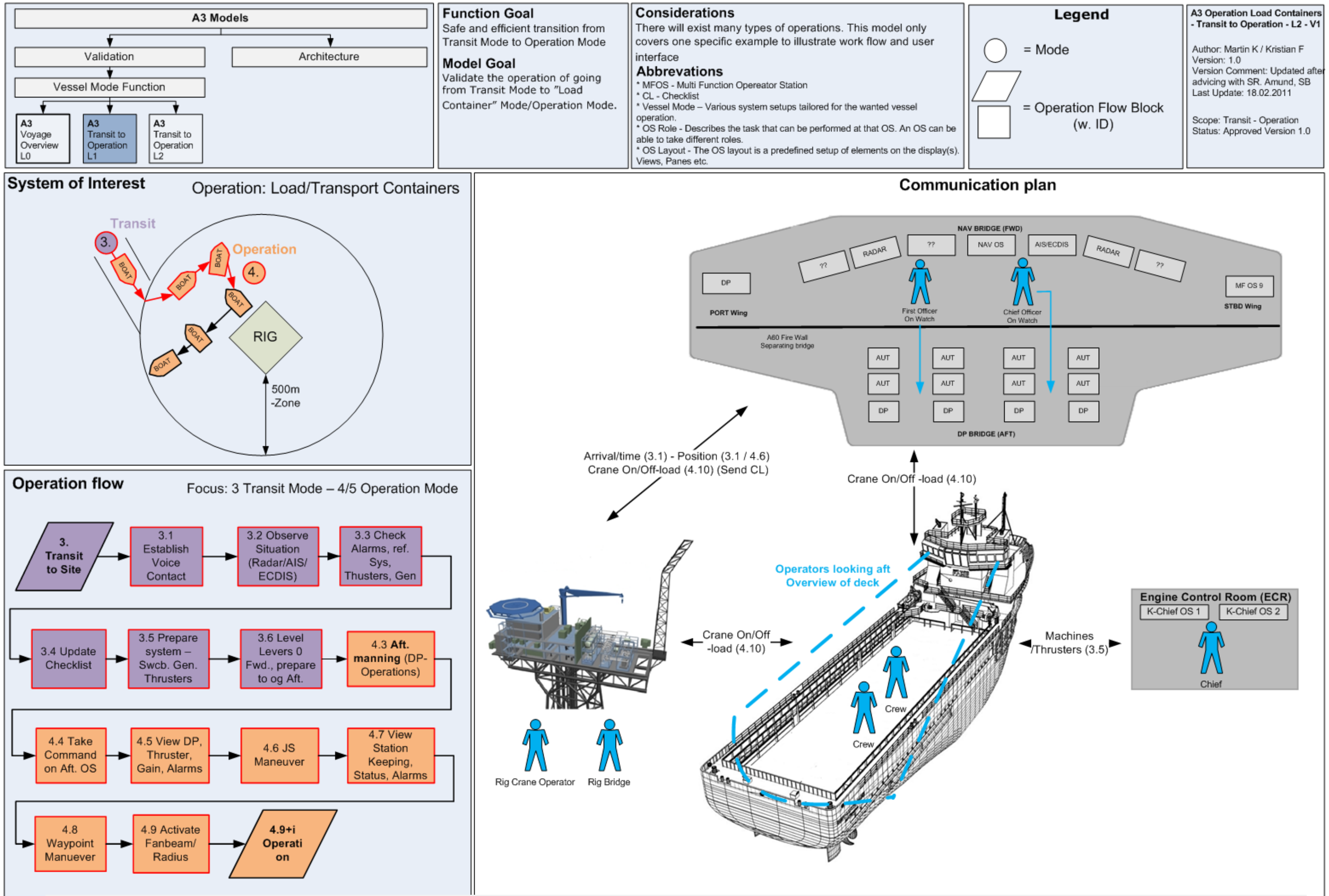
Principle*:* Multiple Abstraction Levels

Example*:* Dynamic A3s

The ***Challenges****:* Pitfalls and Dilemmas

Conclusions and Questions

Example A3 ”Transit to Operation”



A3 Models

Validation

Architecture

Vessel Mode Function

A3 Voyage Overview L0

A3 Transit to Operation L1

A3 Transit to Operation L2

Function Goal
Safe and efficient transition from Transit Mode to Operation Mode

Model Goal
Validate the operation of going from Transit Mode to ”Load Container” Mode/Operation Mode.

Considerations
There will exist many types of operations. This model only covers one specific example to illustrate work flow and user interface

Abbrevations
- * MFOS - Multi Function Opereator Station
- * CL - Checklist
- * Vessel Mode – Various system setups tailored for the wanted vessel operation.
- * OS Role - Describes the task that can be performed at that OS. An OS can be able to take different roles.
- * OS Layout - The OS layout is a predefined setup of elements on the display(s). Views, Panes etc.

Legend

○ = Mode

▱ □ = Operation Flow Block (w. ID)

A3 Operation Load Containers - Transit to Operation - L2 - V1

Author: Martin K / Kristian F
Version: 1.0
Version Comment: Updated after advicing with SR. Amund, SB
Last Update: 18.02.2011

Scope: Transit - Operation
Status: Approved Version 1.0

System of Interest Operation: Load/Transport Containers

Communication plan

NAV BRIDGE (FWD)
??
RADAR
??
NAV OS
AIS/ECDIS
RADAR
??
DP
PORT Wing
First Officer On Watch
Chief Officer On Watch
MF OS 9
STBD Wing
A60 Fire Wall Separating bridge
AUT
DP
DP BRIDGE (AFT)
Arrival/time (3.1) - Position (3.1 / 4.6)
Crane On/Off-load (4.10) (Send CL)
Crane On/Off -load (4.10)
Operators looking aft Overview of deck
Crane On/Off -load (4.10)
Machines /Thrusters (3.5)
Engine Control Room (ECR)
K-Chief OS 1
K-Chief OS 2
Chief
Crew
Crew
Rig Crane Operator
Rig Bridge

Operation flow Focus: 3 Transit Mode – 4/5 Operation Mode

3. Transit to Site → 3.1 Establish Voice Contact → 3.2 Observe Situation (Radar/AIS/ ECDIS) → 3.3 Check Alarms, ref. Sys, Thusters, Gen → 3.4 Update Checklist → 3.5 Prepare system – Swcb. Gen. Thrusters → 3.6 Level Levers 0 Fwd., prepare to og Aft. → 4.3 **Aft. manning** (DP-Operations) → 4.4 Take Command on Aft. OS → 4.5 View DP, Thruster, Gain, Alarms → 4.6 JS Maneuver → 4.7 View Station Keeping, Status, Alarms → 4.8 Waypoint Manuever → 4.9 Activate Fanbeam/ Radius → **4.9+i Operati on**

from: Kristian Frøvold, Applying A3 reports for early validation and optimization of stakeholder communication in development projects, INCOSE 2017 in Adelaide, Australia http://gaudisite.nl/INCOSE2017_Frovold_A3.pdf

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

Multiple Abstraction Levels

The ***Problem:*** The Coming Software Apocalypse

The ***Solution****:* A3 Architecture Overviews

Principle*:* Three Main Views

Example*:* Visualizations

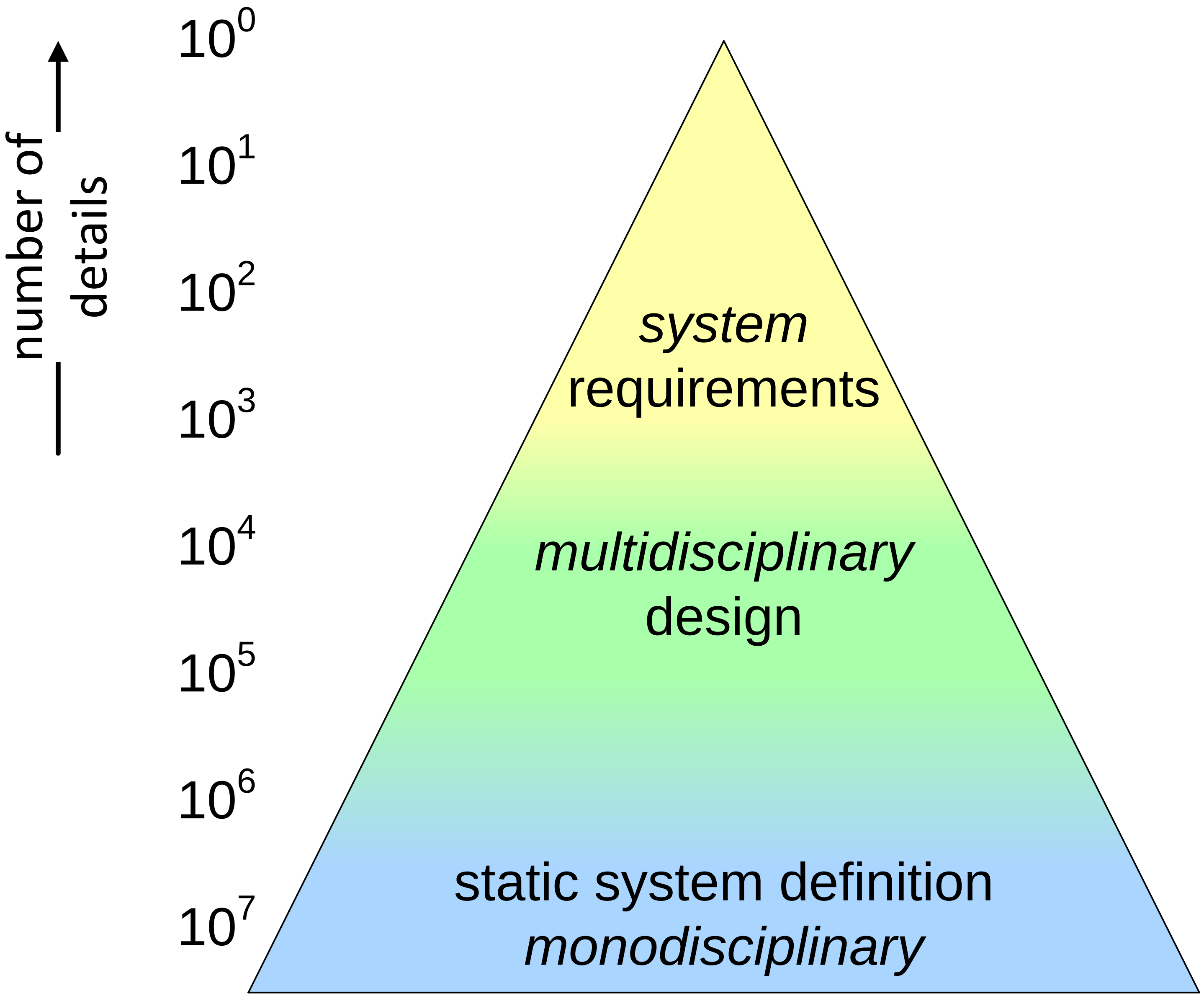
Principle: Multiple Abstraction Levels

Example*:* Dynamic A3s

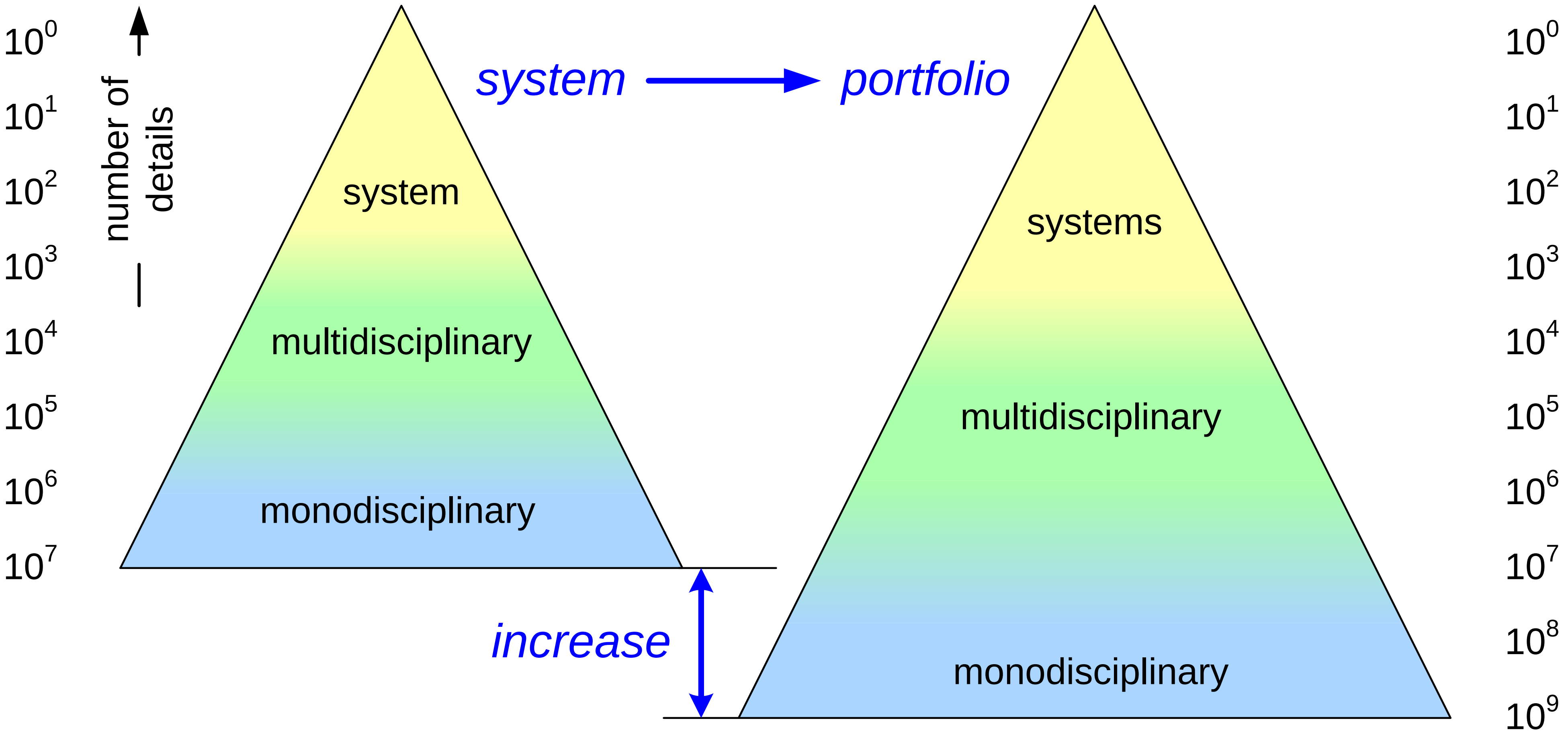
The ***Challenges****:* Pitfalls and Dilemmas

Conclusions and Questions

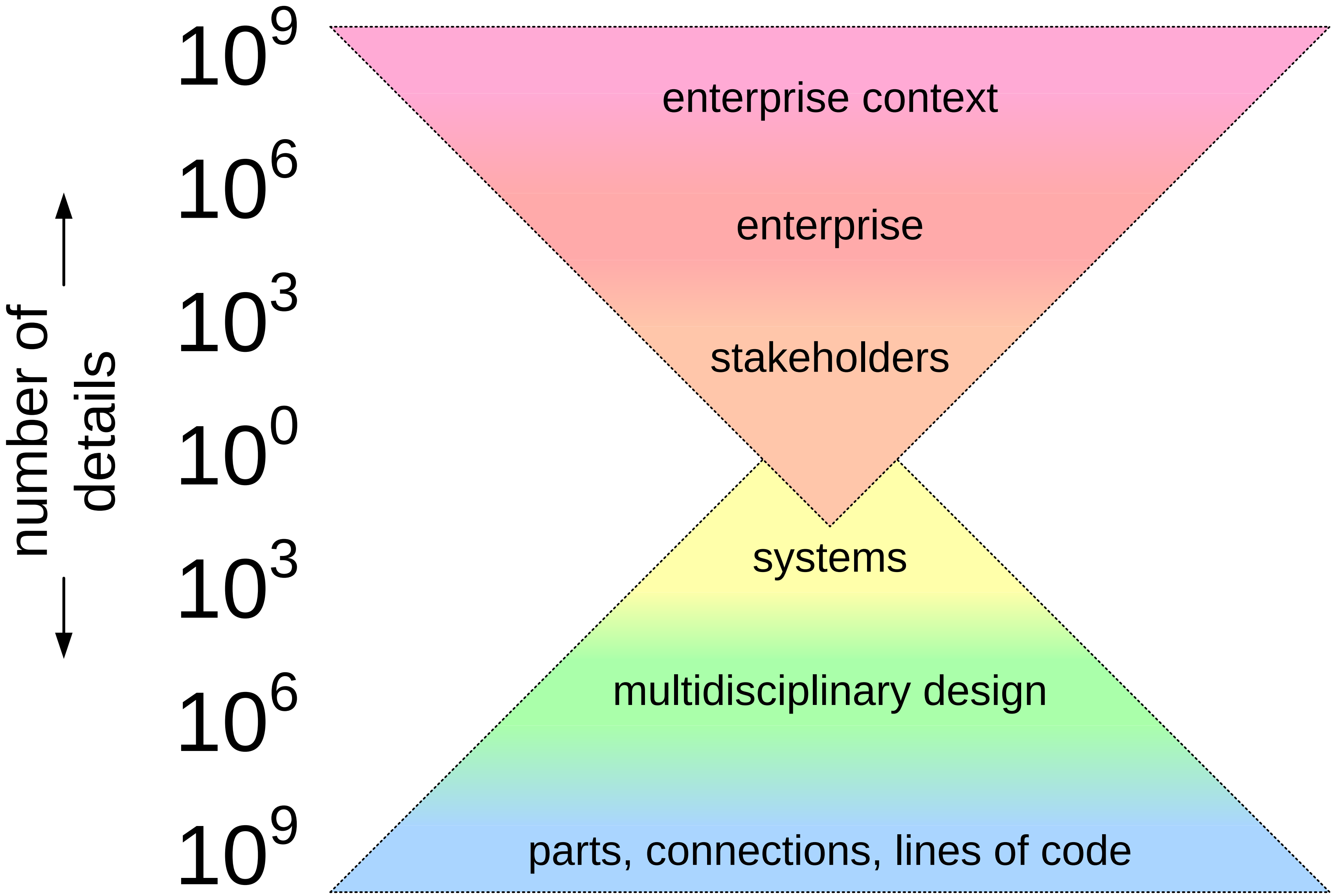
Level of Abstraction Single System



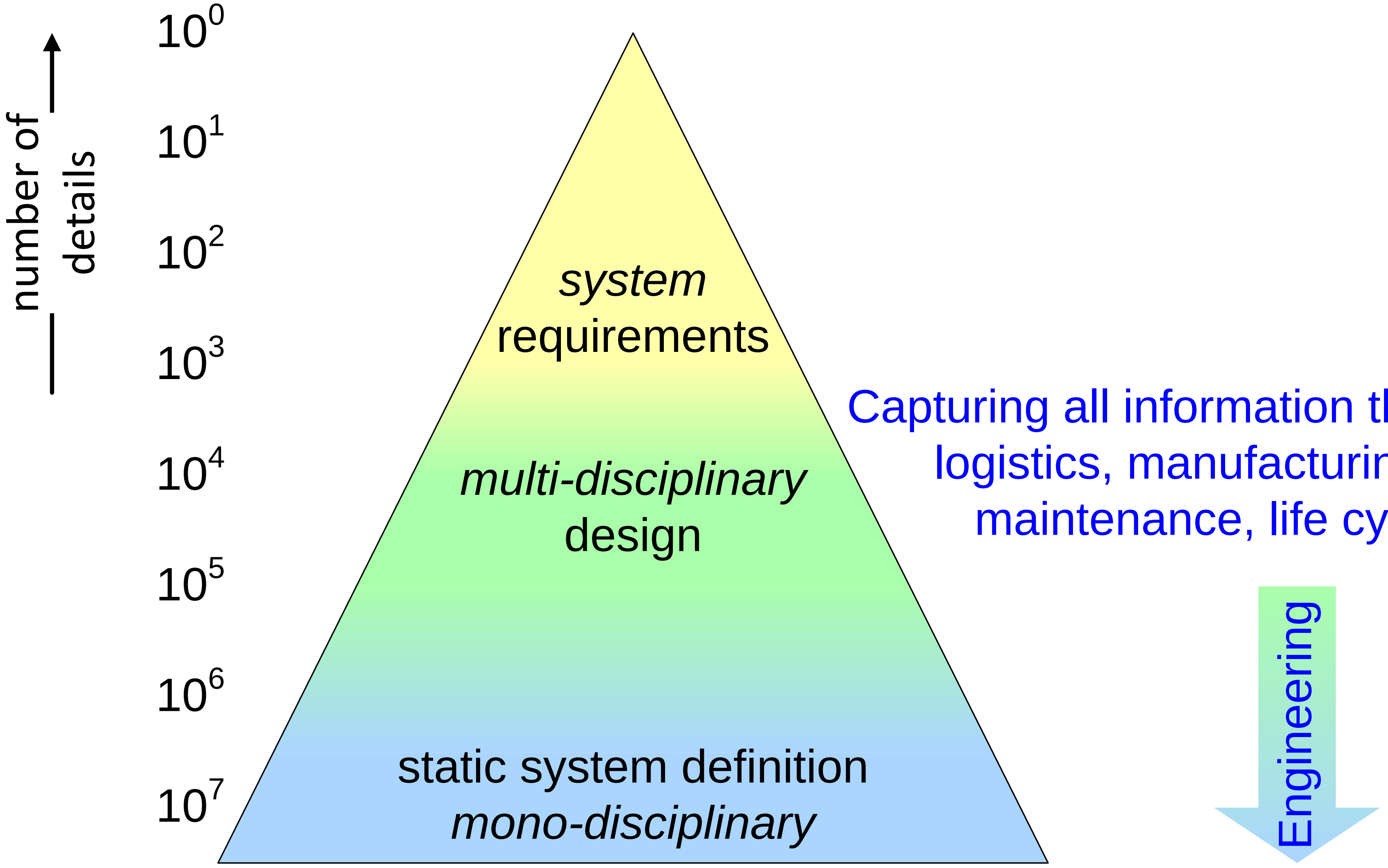
From system to Product Family or Portfolio



Product Family in Context

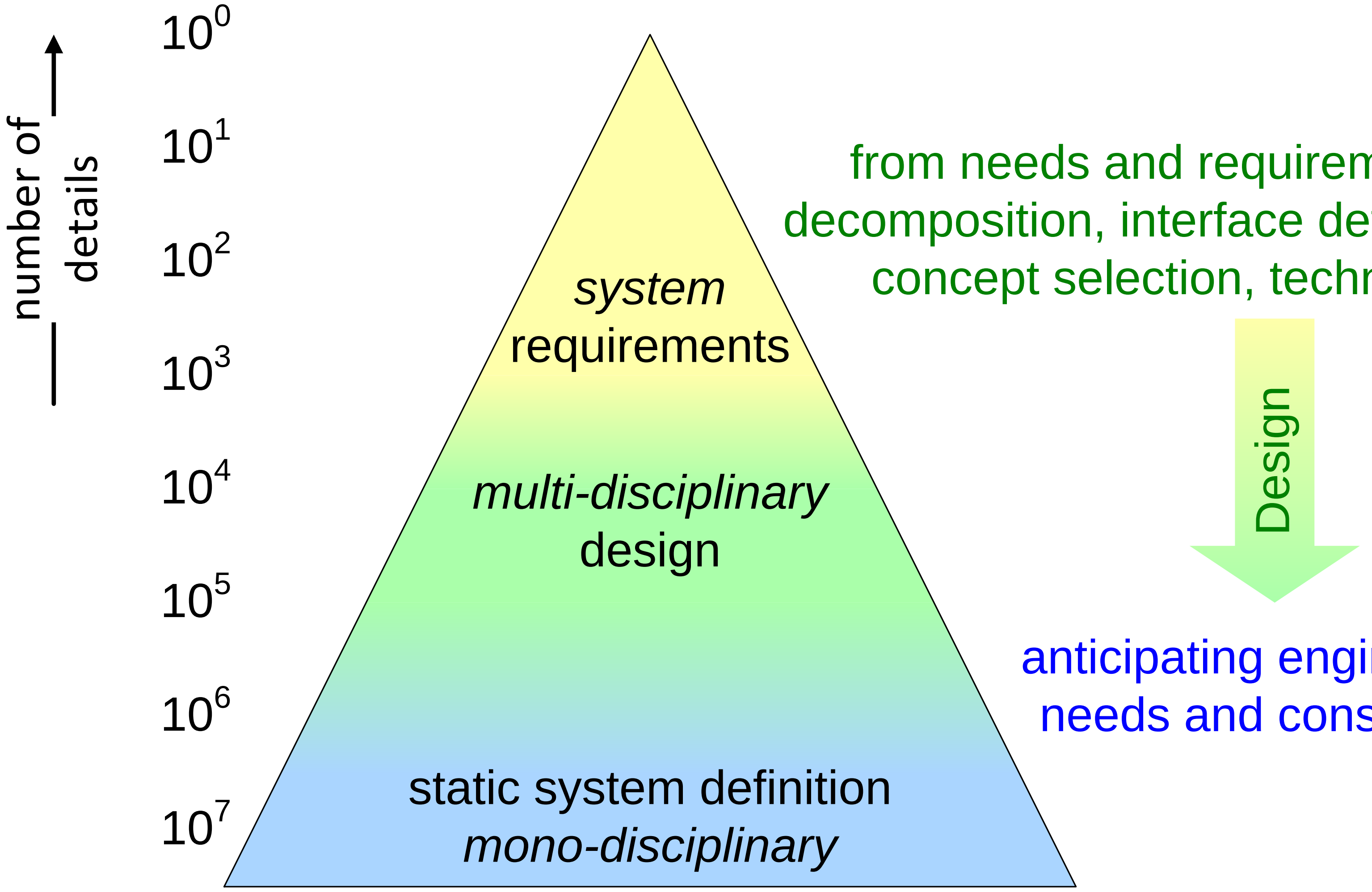


Engineering



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

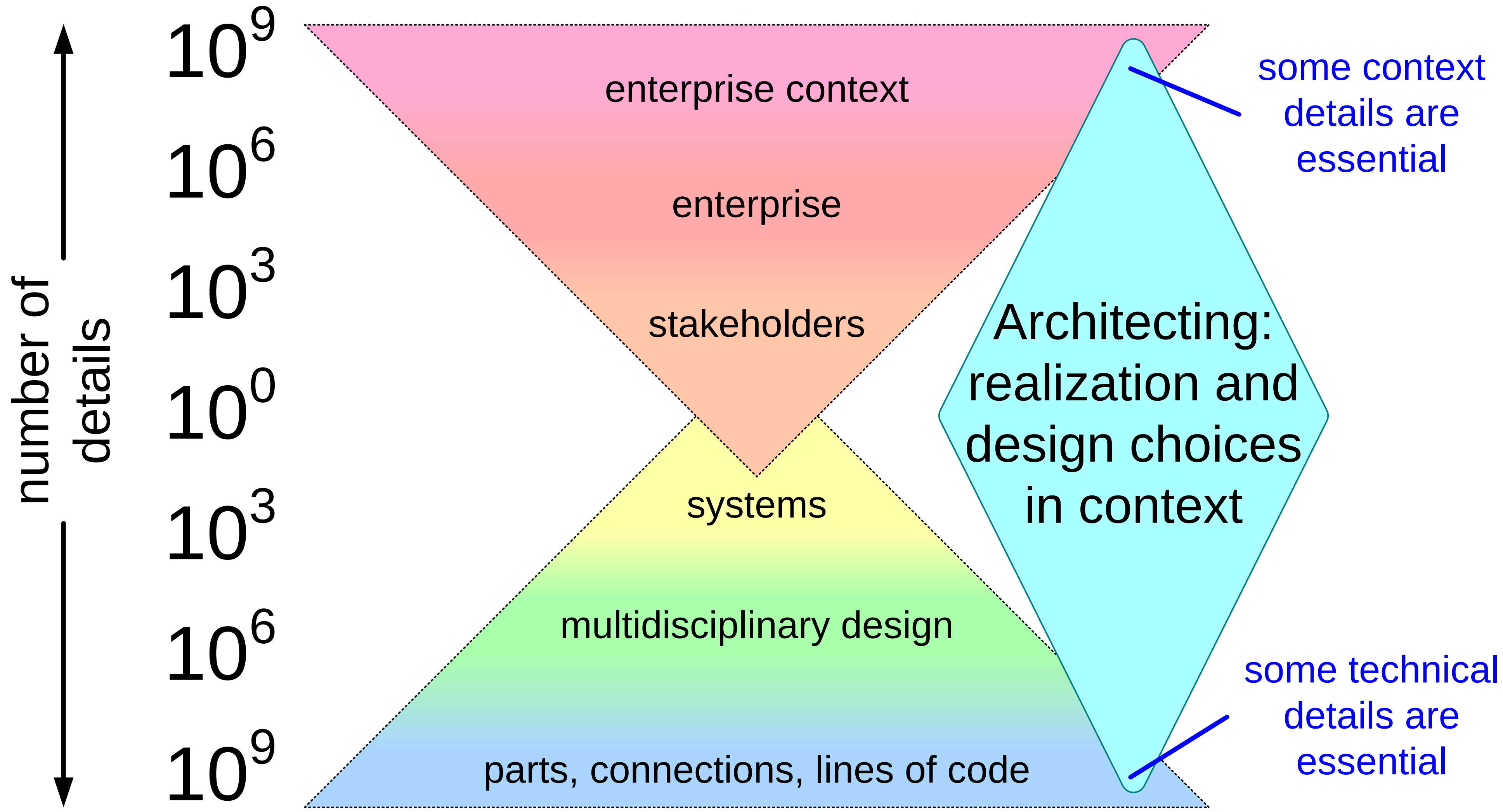
Design



from needs and requirements to design: decomposition, interface definition, allocation, concept selection, technology choices

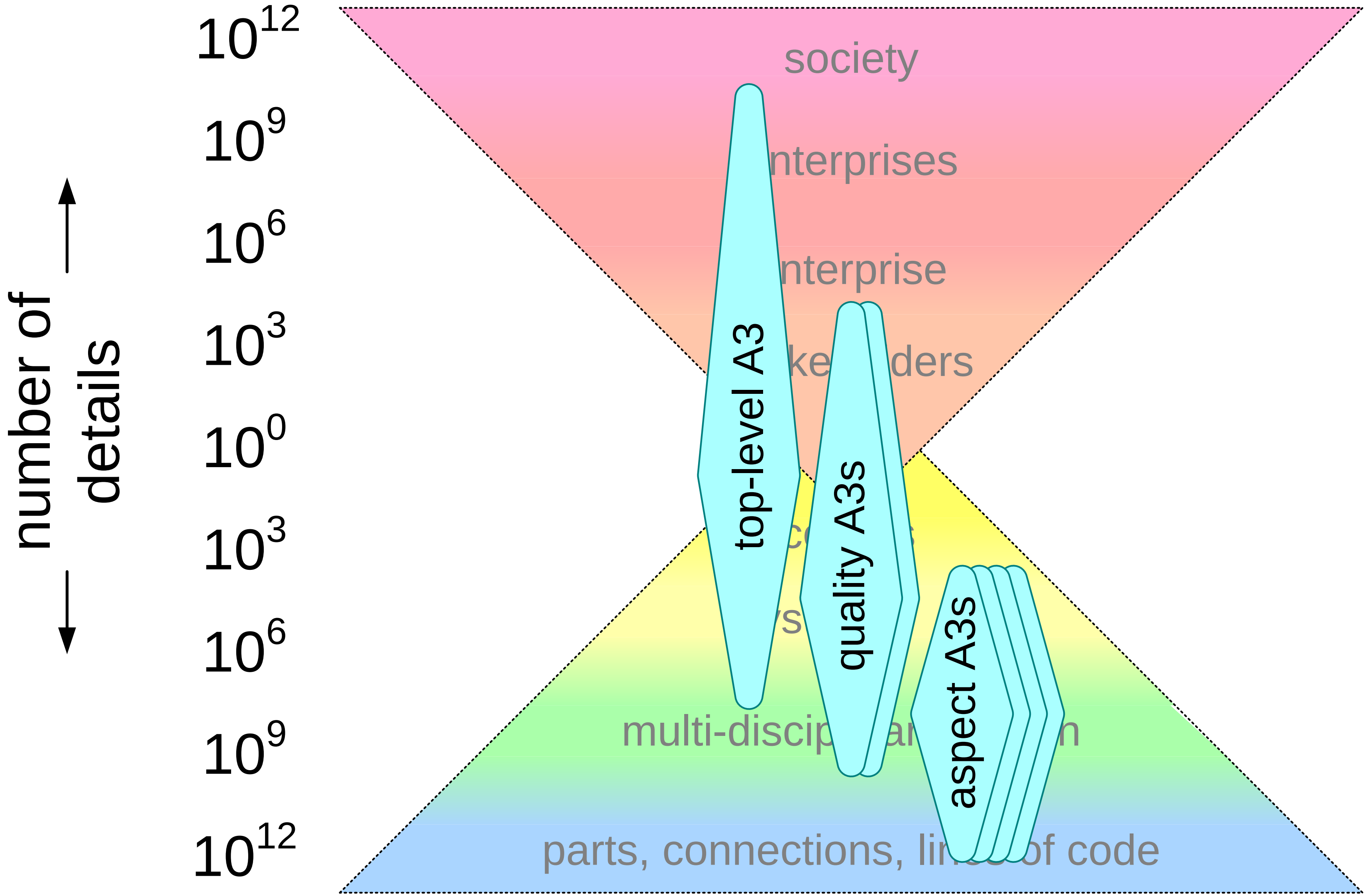
anticipating engineering needs and constraints

Architecting



number of details
10^9
10^6
10^3
10^0
10^3
10^6
10^9
enterprise context
enterprise
stakeholders
systems
multidisciplinary design
parts, connections, lines of code
Architecting: realization and design choices in context
some context details are essential
some technical details are essential

Architecture Overviews at Multiple Levels



Example of Dynamic A3s

*The **Problem:*** The Coming Software Apocalypse

*The **Solution**:* A3 Architecture Overviews

***Principle**:* Three Main Views

***Example**:* Visualizations

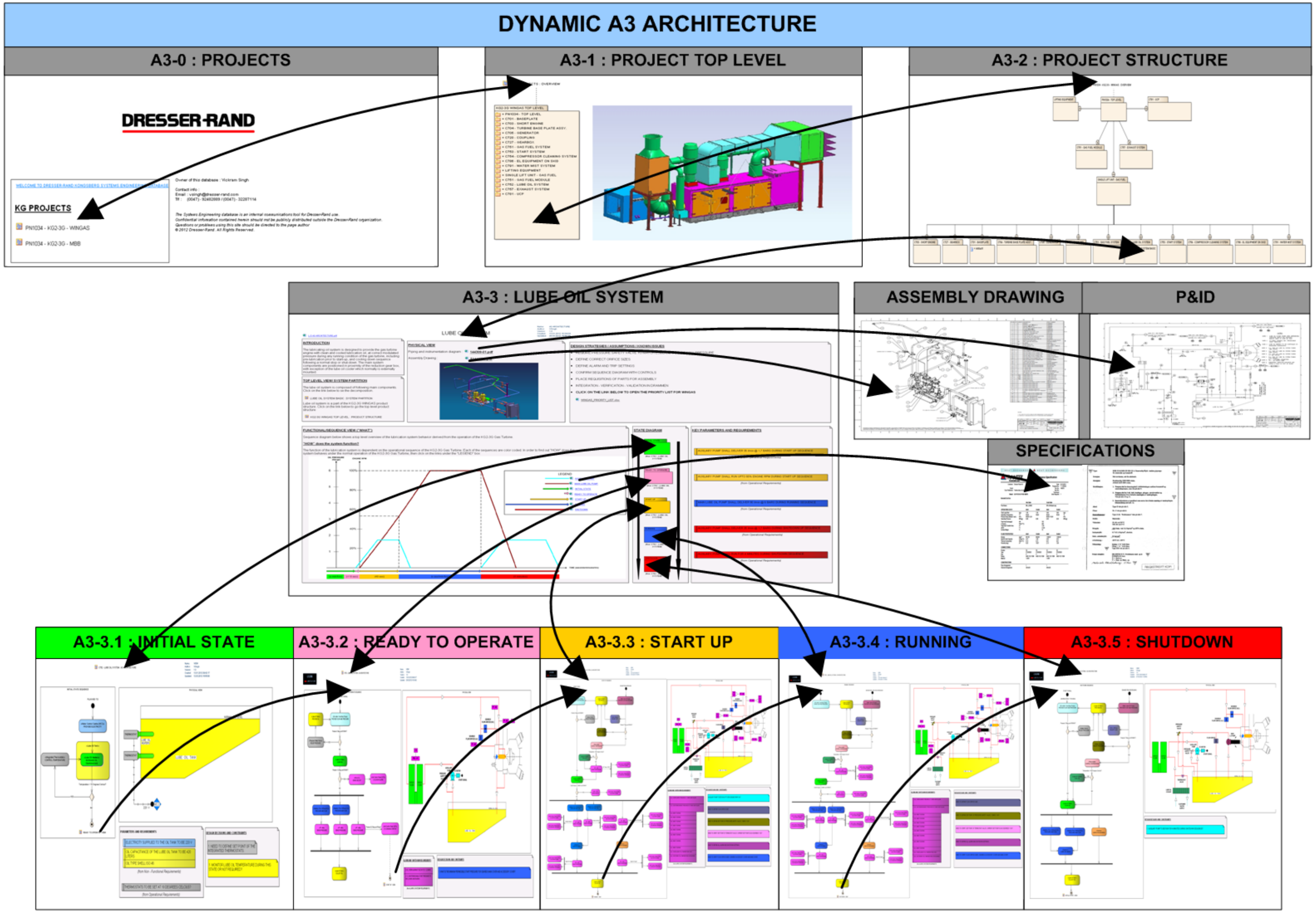
Principle: Multiple Abstraction Levels

***Example**:* Dynamic A3s

*The **Challenges**:* Pitfalls and Dilemmas

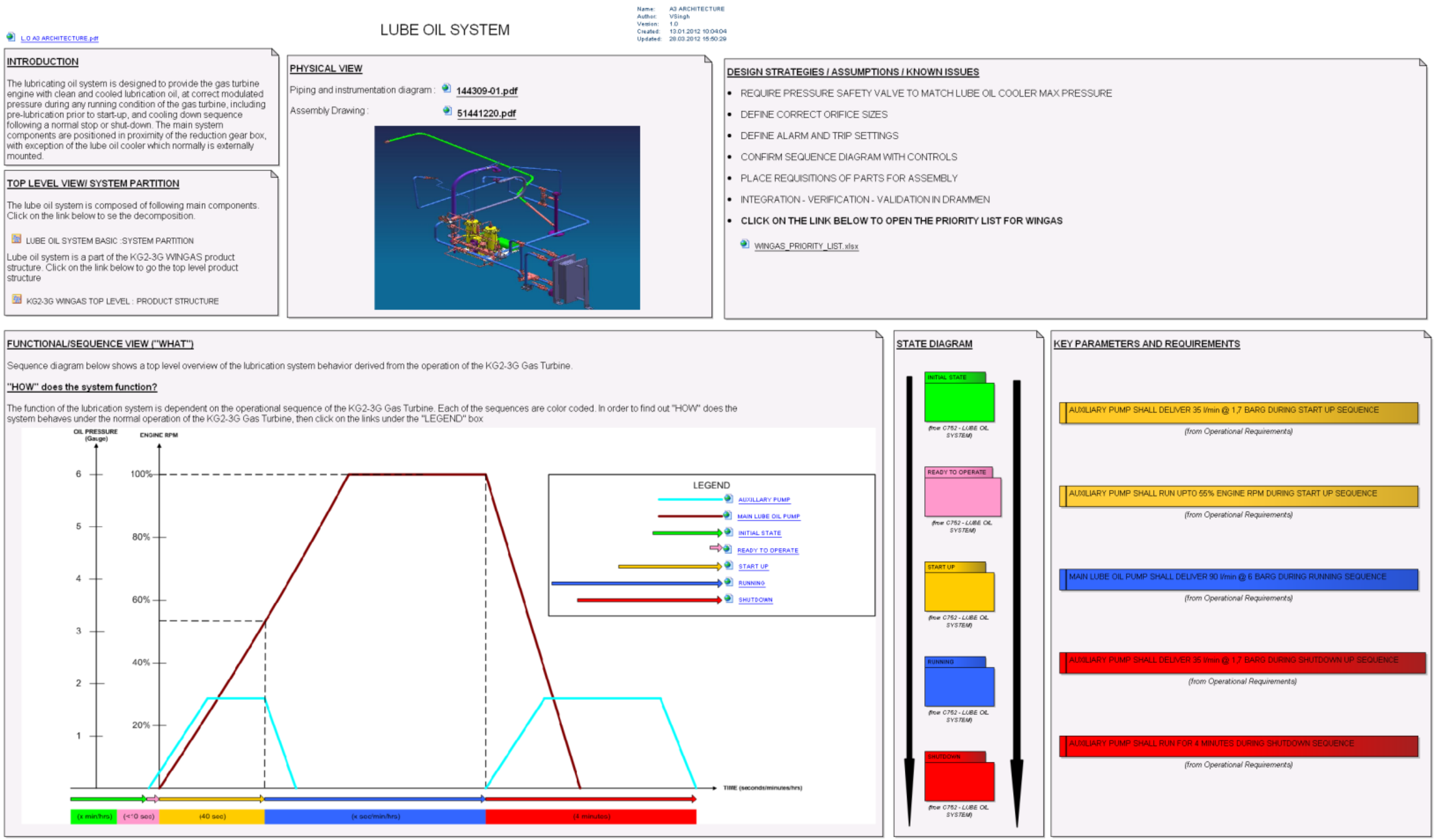
Conclusions and Questions

Example Dynamic A3s for Gas Turbines



from Vickram Singh, Knowledge Capture, Cross Boundary Communication and Early Validation with Dynamic A3 Architectures, INCOSE 2013 in Philadelphia, http://gaudisite.nl/INCOSE2013_Singh_Muller_DynamicA3.pdf

A3 Luboil System Gas Turbine



L.0 A3 ARCHITECTURE.pdf

LUBE OIL SYSTEM

Name: A3 ARCHITECTURE
Author: VSingh
Version: 1.0
Created: 13.01.2012 10:04:04
Updated: 28.03.2012 15:50:29

INTRODUCTION

The lubricating oil system is designed to provide the gas turbine engine with clean and cooled lubrication oil, at correct modulated pressure during any running condition of the gas turbine, including pre-lubrication prior to start-up, and cooling down sequence following a normal stop or shut-down. The main system components are positioned in proximity of the reduction gear box, with exception of the lube oil cooler which normally is externally mounted.

TOP LEVEL VIEW/ SYSTEM PARTITION

The lube oil system is composed of following main components. Click on the link below to se the decomposition.

LUBE OIL SYSTEM BASIC :SYSTEM PARTITION

Lube oil system is a part of the KG2-3G WINGAS product structure. Click on the link below to go the top level product structure

KG2-3G WINGAS TOP LEVEL : PRODUCT STRUCTURE

PHYSICAL VIEW

Piping and instrumentation diagram : 144309-01.pdf

Assembly Drawing : 51441220.pdf

DESIGN STRATEGIES / ASSUMPTIONS / KNOWN ISSUES

- REQUIRE PRESSURE SAFETY VALVE TO MATCH LUBE OIL COOLER MAX PRESSURE
- DEFINE CORRECT ORIFICE SIZES
- DEFINE ALARM AND TRIP SETTINGS
- CONFIRM SEQUENCE DIAGRAM WITH CONTROLS
- PLACE REQUISITIONS OF PARTS FOR ASSEMBLY
- INTEGRATION - VERIFICATION - VALIDATION IN DRAMMEN
- **CLICK ON THE LINK BELOW TO OPEN THE PRIORITY LIST FOR WINGAS**

WINGAS_PRIORITY_LIST.xlsx

FUNCTIONAL/SEQUENCE VIEW ("WHAT")

Sequence diagram below shows a top level overview of the lubrication system behavior derived from the operation of the KG2-3G Gas Turbine.

"HOW" does the system function?

The function of the lubrication system is dependent on the operational sequence of the KG2-3G Gas Turbine. Each of the sequences are color coded. In order to find out "HOW" does the system behaves under the normal operation of the KG2-3G Gas Turbine, then click on the links under the "LEGEND" box

STATE DIAGRAM

INITIAL STATE
(from C752 - LUBE OIL SYSTEM)
READY TO OPERATE
(from C752 - LUBE OIL SYSTEM)
START UP
(from C752 - LUBE OIL SYSTEM)
RUNNING
(from C752 - LUBE OIL SYSTEM)
SHUTDOWN
(from C752 - LUBE OIL SYSTEM)

KEY PARAMETERS AND REQUIREMENTS

AUXILIARY PUMP SHALL DELIVER 35 l/min @ 1,7 BARG DURING START UP SEQUENCE
(from Operational Requirements)

AUXILIARY PUMP SHALL RUN UPTO 55% ENGINE RPM DURING START UP SEQUENCE
(from Operational Requirements)

MAIN LUBE OIL PUMP SHALL DELIVER 90 l/min @ 6 BARG DURING RUNNING SEQUENCE
(from Operational Requirements)

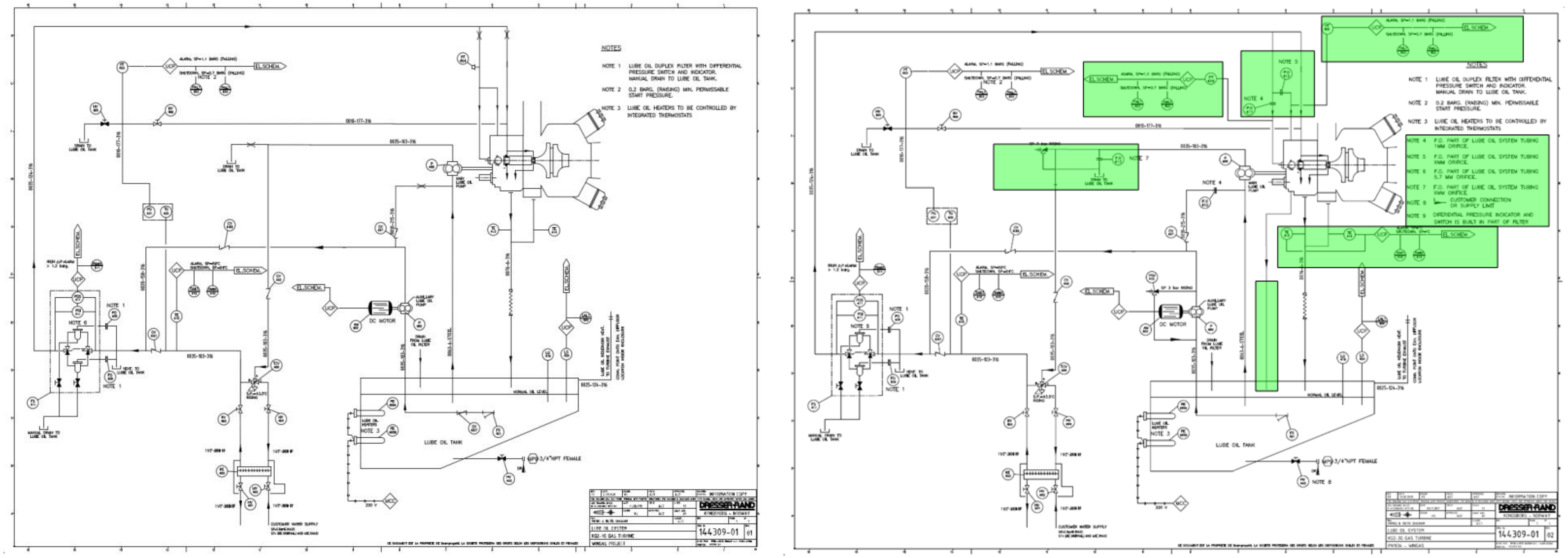
AUXILIARY PUMP SHALL DELIVER 35 l/min @ 1,7 BARG DURING SHUTDOWN UP SEQUENCE
(from Operational Requirements)

AUXILIARY PUMP SHALL RUN FOR 4 MINUTES DURING SHUTDOWN SEQUENCE
(from Operational Requirements)

from Vickram Singh, Knowledge Capture, Cross Boundary Communication and Early Validation with Dynamic A3 Architectures, INCOSE 2013 in Philadelphia, http://gaudisite.nl/INCOSE2013_Singh_Muller_DynamicA3.pdf

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!



from Vickram Singh, Knowledge Capture, Cross Boundary Communication and Early Validation with Dynamic A3 Architectures, INCOSE 2013 in Philadelphia, http://gaudisite.nl/INCOSE2013_Singh_Muller_DynamicA3.pdf

Pitfalls, Dilemmas, Conclusions, and Questions

*The **Problem:*** The Coming Software Apocalypse

*The **Solution**:* A3 Architecture Overviews

***Principle**:* Three Main Views

***Example**:* Visualizations

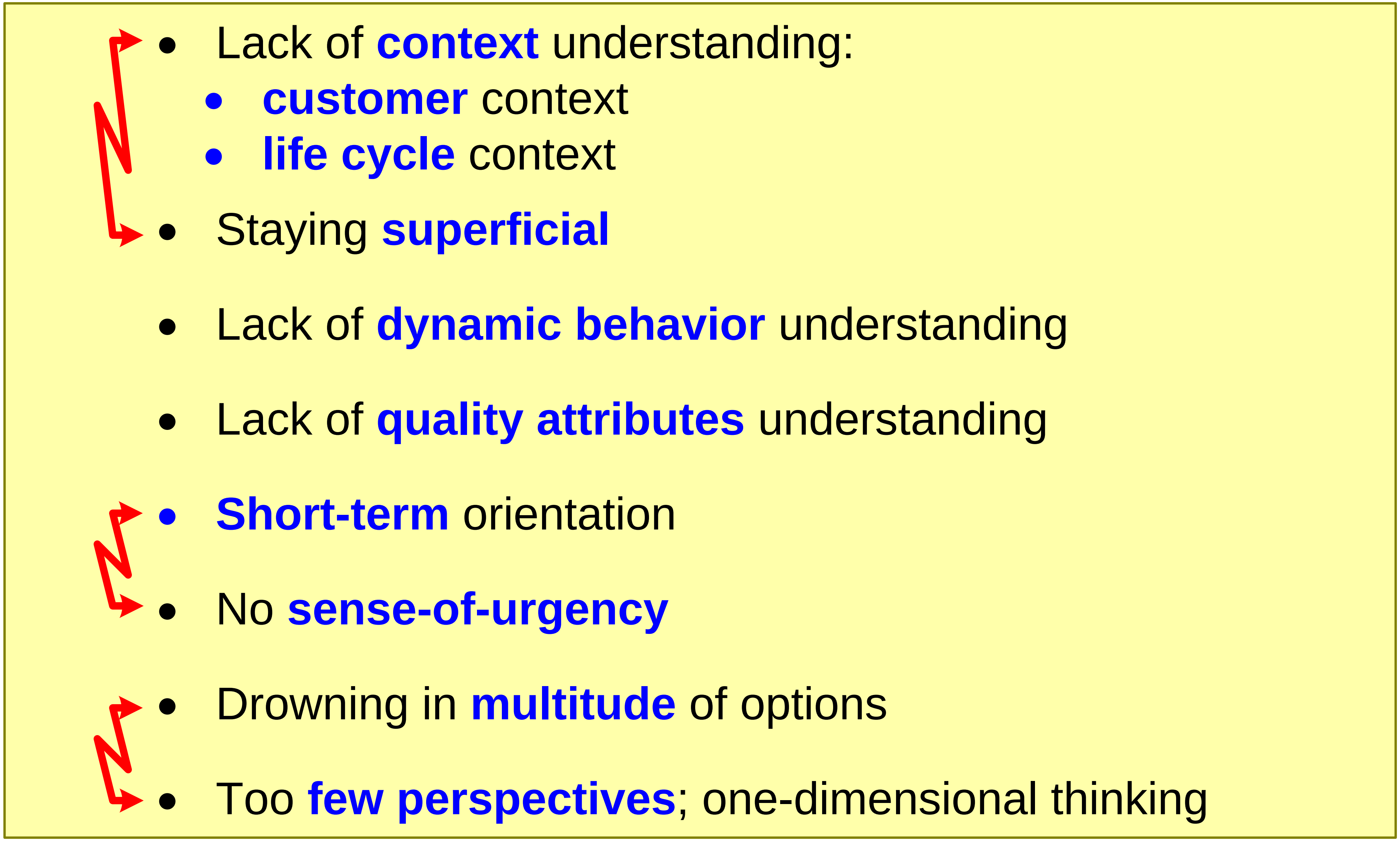
Principle: Multiple Abstraction Levels

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

Conclusions, and Questions

*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

?