Arcadia and Capella on the Field: Real-World MBSE Use Cases

MBSE Symposium, Canberra
October 27th, 2014

Stéphane Bonnet, Fabrice Lestideau, Jean-Luc Voirin
**ENGINEERING KEY QUESTIONS**

- How is the customer need received? How are its **consistence and feasibility checked**?

- Which are the **engineering phases in the solution elaboration**, how are they related?

- **How is complexity managed?**

- How are different alternatives evaluated, **how do the specialists collaborate?**

- **How is the solution justified** w.r.t. the need and the different constraints?

**FACTS**

- Needs and solutions are more complex, more stakeholders, more constraints, less time

- **The approach Doors / Word / Visio / Excel reaches limits**

- Manual processes are not compatible with agility and short loops
The Thales MBSE Approach

Model-Based Engineering Method for Architectural Design

Graphical Modelling Workbench supporting Arcadia
Arcadia: Model-Based Method for Architectural Design

NEED UNDERSTANDING

What the users of the system need to accomplish

What the system has to accomplish for the users

HOW THE SYSTEM WILL WORK TO FULFILL EXPECTATIONS

How the system will be developed and built

SOLUTION ARCHITECTURAL DESIGN

What the users of the system need to accomplish

What the system has to accomplish for the users

How the system will work to fulfill expectations

How the system will be developed and built
Arcadia: Multi-Viewpoint Trade-Off Analysis
Guidance
[Embedded methodological browser]

Complexity management
[Abstraction via computed information]

Productivity tools
[Automated transitions and diagram creation accelerators]

Model Analysis & Navigation
[Model validation, semantic browser]

Multi-criteria analysis
[Viewpoints and management framework]

First operational deployments in 2009
Used on most major engineering projects
Currently being open sourced
Use Case 1:

Managing System Design Complexity
Issues in the latest phases of operational validation

Very good design documents, but in silos

1 man month to **reverse a first level of detail** in a model, based on existing documents

**First time overall views have been available**

- Good support for discussion
- Visualization of transverse functional chains
Managing System Design Complexity

275 Functions
(230 Leaves)

578 Functional Exchanges between leaf functions

5 levels of decomposition
Contextual Diagrams: Low-level internals
Challenge: Build and maintain simplified views

How to analyze transverse topics?
How to get transverse overviews?
Managing System Design Complexity

Computed Diagrams: High-level Functions, Low-level Exchanges
Managing System Design Complexity

Children of F21 and F22 not displayed

Ports on F21 and F22 are graphically computed
Managing System Design Complexity

MODEL

F1

F2

F21

F22

VIEW

F1

F2

Children of F2 not displayed

Ports on F2 are graphically computed
Children of F1 and F2 not displayed

Ports on F1 and F2 are graphically computed
Managing System Design Complexity

Tag-based simplification mechanism

Computed graphical simplifications free engineers from tedious and error-prone maintenance of abstraction levels
Managing System Design Complexity: Global Overview

End-to-end visualization of Functional Chains
Use Case 2:

MBSE-based Change Management
Use Case 2: Change Management

Context

- Maritime Patrol Program delivered to the Customer
- New functionalities asked by the Customer

MBSE usage

- Up-to-date model of the delivered System available
- Modification of the model in order to:
  - Estimate feasibility, cost and risks
  - Drive developments and IVVQ
- Product line management
A regular reading pattern

- Multiple contributors modelling the same way
- Facilitates first access to diagrams
- Eases diagram review
- Allows quick inconsistency detection
Need Representation based on Delivered Solution

SSS: Need

PIDS: Reverse Engineering from Software Specification

Operator

HMI

DATA

PROCESS/INTERFACES

EQUIPMENT
Managing Change: Feasibility and Risks

New Customer needs

Existing Functional Chains

Impacted Functions

New Functional Chains
### COST Analysis Viewpoint

Elementary work decomposition

- Panels
- External / Internal Interfaces
- Data Memorisation
- Processing Complexity

---

#### Capella Outputs (model export)

The table below shows the computed data for estimated cost.

<table>
<thead>
<tr>
<th>Part</th>
<th>FK</th>
<th>N</th>
<th>O</th>
<th>P</th>
<th>Q</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Computed Data: Estimated Cost

The table illustrates the estimated cost computation based on various factors such as panels, external/internal interfaces, data memorisation, and processing complexity.
Use Case 3:

Multi-Level Engineering
Use Case 3: Multi-Level Engineering and Automated Transitions

Context
- Complex systems with full Thales responsibility (from Mission System to SW Component)

MBSE usage
- Setup a global, multi-level engineering approach
- Joint effort with Thales Airborne Systems / Thales Corporate to specify and develop an automated, iterative transition
- Incubation on two projects
- Now integrated in the product and used in other contexts
Multi-Level Engineering and Automated Transitions

Radar Engineering

Operational Analysis
- Pilot
- Target
- Missile
- Jammer

System Analysis
- SYS 1
- SYS 2
- SYS 3
- Environment

Logical Architecture
- Mode A
- Mode B
- Mode C
- Technical Functions

Physical Architecture
- PU computer
- STR
- CFOQ
- EAAD
- RDP

Product Breakdown Structure
- Mode A
- Mode B
- Mode C
- RDP... Modes, F Svc
- PU computer

Processing Unit Engineering

Traceability link
System Physical Architecture

Subsystem Need Analysis

Subsystem Logical Architecture

Subsystem Physical Architecture

Multi-Level Engineering and Automated Transitions

Co-Engineering

Computed system - subsystem traceability
Use Case 4:

Model-driven IVV
Functional Chains Driving Engineering Activities

| Requirements are clarified with Functional Chains |
| Test Procedures are linked to Functional Chains |

<table>
<thead>
<tr>
<th>Req</th>
<th>Chain</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>R11</td>
<td>CF-1</td>
<td>E0</td>
</tr>
<tr>
<td>R20</td>
<td>CF-3</td>
<td>E2:1</td>
</tr>
<tr>
<td></td>
<td>CF-2</td>
<td></td>
</tr>
</tbody>
</table>

IVVQ Management

Test_1 {R11,...} MDR-CF-1
Test_2 {R20, R21,...} MDR-CF-2
IVV Strategy: Requested Versions / Developed Versions

Define operational content expected for each project milestone

Deduce functional content and components to be delivered

Define components versions and content
Red: Delayed, missing
Grey: expected in further version
Release management viewpoint:
Automated visualization of versions
Developed Version 1
Available elements in BLUE
Developed Version 2
Available elements in CYAN
Developed Versions 1 & 2
Common available elements in GREY
Compare Planned vs Developed versions
Lots of Other Different Use Cases

- Capella customizations
- Multi-level MBSE
- Measured gains on IVV
- Progress Monitoring
- Code generation
- Product Line modelling
- Model-driven IVV
- Performance analysis
- Cost estimation
- Legacy Interfaces

And more to come!
Thank you for your attention!

Any Questions?