Guide to Arcadia and Capella
Successful Adoption
Introduction: Pascal Roques

- Senior consultant, 25+ years of modeling experience
  - SADT, OMT, UML, SysML, ARCADIA/Capella

- UML2 and SysML Certified by the OMG

- Trainer for Thales on Arcadia/Capella
  - 130+ sessions, 1500+ trainees

- Member of Capella

- Author of UML/SysML best-sellers in France
  - … and of the first Capella book!
MBSE?

- From Document-Centric to Model-Centric!

- Models are not only dedicated to documentation
- Models become major engineering artifacts!
MBSE pillars

Language

Tool
MBSE pillars with SysML

- Language
- Tool
- Method (TBD!)
MBSE pillars with Arcadia/Capella

Language

Tool

Method
### Method: Arcadia

- Many concepts, diagrams and even levels are optional!

<table>
<thead>
<tr>
<th>Method Steps</th>
<th>Tasks</th>
<th>Sample Model</th>
<th>Concepts</th>
<th>Description Means</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer</td>
<td>Define operational capabilities</td>
<td>- Operational capabilities</td>
<td>- Operational processes chaining activities</td>
<td>Dataflow: functions, ops. activities, interactions &amp; exchanges</td>
</tr>
<tr>
<td>Need Analysis</td>
<td>Perform an operational need analysis</td>
<td>- Actors, operational entities</td>
<td>- Actors and system capabilities</td>
<td>Scenarios: actors, system, components interactions &amp; exchanges</td>
</tr>
<tr>
<td>System/</td>
<td>Perform a capability trade-off analysis</td>
<td>- Interactions between activities &amp; actors</td>
<td>- Functional processes through functions &amp; ops. activities</td>
<td>Functional chains, operational processes through functions &amp; ops. activities</td>
</tr>
<tr>
<td>SW/HW Need Analysis</td>
<td>Perform a functional and non-functional analysis</td>
<td>- Information used in activities &amp; interactions</td>
<td>- Scenario for dynamic behaviour</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Formalise and consolidate requirements</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logical</td>
<td>Define architecture drivers and viewpoints</td>
<td>SAME CONCEPTS PLUS:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Architecture</td>
<td>Build candidate architectural breakdowns in components</td>
<td>- Components</td>
<td>Component wiring: all kinds of components</td>
<td></td>
</tr>
<tr>
<td>Design</td>
<td>Select best compromise architecture</td>
<td>- Component ports and interfaces</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Solutions</td>
<td></td>
<td>- Exchanges between components</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Function allocation to components</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Physical</td>
<td>Define architectural patterns</td>
<td>SAME CONCEPTS PLUS:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Architecture</td>
<td>Consider reuse of existing assets design, physical</td>
<td>- Behavioural components refining logical ones, and implementing functional</td>
<td>Component wiring: all kinds of components</td>
<td></td>
</tr>
<tr>
<td>Design</td>
<td>Design a physical reference architecture</td>
<td>behaviour</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Validate and check it</td>
<td>- Implementation components supplying resources for behavioural components</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Development</td>
<td>Define a components IPWQ strategy</td>
<td>- Physical links between implementation components</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Contracts</td>
<td>Define &amp; enforce a PBS and component integration contract</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>What is expected</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>from each designer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>or sub-contractor</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Customize Arcadia!

- Define the right modeling approach for your project/product
- Arcadia must be tailored to your context:
  - Type of system (size, complexity, etc.)
  - Nature of project (small, distributed, etc.)
  - Background of people
Consider Training and Mentoring

- **Modeling is a skill, Modeling is difficult**
  - A lot of concepts, How to choose the right diagram?
  - When to stop?

- **Begin with Training**
  - Ask experts
  - Books are not sufficient
  - Different levels of training for different levels of modeling

- **Continue with Coaching**
  - Keeps you on rails
  - Model Review
Think Big, Start Small! (1/2)

- **Start with a pilot project**
  - Not too big, nor critical,
  - Not too simple ("Toy" problem)
  - Imply a small set of core modelers (volunteers), and a modeling "champion"
  - Keep the focus on project deliverables, and model only as far as you need to answer the questions

- **Provide simple guidelines**
  - Focus on simple products with obvious near term value that SEs would have to create anyway
  - Select Arcadia diagrams in a preferred sequence
  - Model validation rules help ease the fear to newcomers that they are going to "mess up" the model
  - Avoid too complex diagrams
Correct and enrich the process from feedback

- Applying best practices often requires failing a few times, first
- Do not apply techniques blindly
- Deployment has to be wise and incremental
- Projects will need to schedule time and resources to deploy infrastructure and train workforce

Model as a team!

- Collaboration in a multi-center modeling effort
- Identify and imply all important Stakeholders
- Make the model become the reference of the SE work!
- Don’t forget to create the right diagrams for the right readers!
Stay Driven by ROI! (1/2)

- **Model with a purpose!**
  - Identify your modeling objectives
  - Don’t model too much
  - Don’t model too early (upfront)
  - Integrate Metrics into your process

- **Don’t forget Configuration and Version Management**
  - Monitor the evolution of the model
  - Provide appropriate process and tooling for:
    - Baselines, branches, diff/merge, model partitioning, user rights…
  - Coordinate interactions with other models
Stay Driven by ROI! (2/2)

- Don’t duplicate models with documents
  - Documentation should be inside the model
  - Navigate through the model
  - Generate mandatory documentation

- MBSE must be integrated into existing SE processes
  - Model development becomes infused within the product development schedule
  - Management must be willing to pay the startup costs and give time for the effort to pay dividends
Capitalize Best Practices! (1/2)

- Do not create “Monster” Models
  - Model recursively
  - Don’t mix responsibilities
  - System – Subsystem add-on

- Build Libraries of Reusable Assets:
  - Units, Basic Types, Domain Types
  - Physical Components… (REC/RPL)
System/Subsystem Transition
Replicable Elements and Libraries

RPLs

Compliance

Compliance

Synchronization

New element

PROJECT MODEL P2

READ ONLY

PROJECT MODEL P3

READ WRITE

PROJECT MODEL P4

READ ONLY

LIBRARY L1

MA Eng. Phases

PROJECT MODEL P1

READ WRITE

MA Eng. Phases
Provide additional tooling:

- Naming conventions
- List of mandatory and optional validation rules
- Documentation generation templates
- Specialty Viewpoints
Architecture Evaluation

- Specialty Viewpoints
Then you have a chance to achieve MBSE!
Food for Thought...

- Each Model (and even each diagram) must have a clear objective and a defined audience.

- A is a good model of B is A can provide satisfactory answers to predefined questions about B (Douglas. T. Ross)

“All models are wrong but some are useful.”

- George Box
To Learn More...

Web Sites:

- www.polarsys.org/capella/index.html
- www.obeo.fr/en/capella-professional-offer
- www.prfc.fr/en

Books