Arcadia and Capella: Augmenting requirements with models to improve the articulation between systems engineering levels and optimize V&V practices

INCOSE SYMPOSIUM 2019
MBSE BEST PAPER AWARD

Stéphane Bonnet, Jean-Luc Voirin, Juan Navas
THALES
1. Preamble: Arcadia method and Capella workbench

2. Model elements ARE requirements

3. Contracts between engineering levels: workflow

4. (Happy) consequences on V&V and incremental development strategy

5. Instantiated workflow
1. Arcadia and Capella
Methodology and high level concepts and viewpoints

Purpose-built to provide the notation and diagrams fitting the Arcadia approach
Arcadia: A "Simple" Engineering Language

**NEED**

- Customer Operational Need Analysis
  - What the users of the system need to accomplish
  - Define operational capabilities
  - Perform an operational need analysis

- System/SW/HW Need Analysis
  - What the system has to accomplish for the Users
  - Perform a capability trade-off analysis
  - Perform a functional and non-functional analysis
  - Formalise and consolidate requirements

**SOLUTION**

- Logical Architecture Design
  - How the system will work so as to fulfill expectations
  - Define architecture drivers and viewpoints
  - Build candidate architectural breakdowns in components
  - Select best compromise architecture

- Physical Architecture Design
  - How the system will be developed & built
  - Define architectural patterns
  - Consider reuse of existing assets design a physical reference architecture
  - Design a physical reference architecture
  - Validate and check it

- Development Contracts
  - What is expected from each designer/sub-contractor
  - Define a components IV&V strategy
  - Define & enforce a PBS and component integration contract

- SAME CONCEPTS, PLUS:
  - Components
  - Component ports and interfaces
  - Exchanges between components
  - Function allocation to components
  - Component Interface justification by functional exchanges allocation

- SAME CONCEPTS, PLUS:
  - Physical elements: interfaces, components, resources
  - Phisical links between components
  - Logical ones and implementing functional elements
  - Logical assets and components
  - Physical assets for behaviour components

- Breakdown of functions & components
- Mode & states

- Same concepts, Plus:
  - Components
  - Component ports and interfaces
  - Exchanges between components
  - Function allocation to components
  - Component Interface justification by functional exchanges allocation

- Data model: dataflow & scenario contents, definition & justification of interfaces

- Configuration Items tree
  - Parts numbers, quantities
  - Development contract (expected behaviour, interfaces, scenarios, resource consumption, non-functional properties...)

- Component wiring: all kinds of components
  - Allocation of operations & activities to actors, of functions to components, of behavior components to impl. components, of dataflows to interfaces, of elements to configuration items
A practitioner-driven journey started in Thales...
... and now continuing beyond
2. Model elements ARE requirements
Textual requirements are at the heart of the current engineering practices.

Need model helps formalize and consolidate customer and system requirements.

Solution model helps validate feasibility, elicit/justify new requirements for the system/subsystems.
Is described by

Functional Chain

Scenario

Function

Capability

Component exchange

Interface

Mode State

Functional exchange

Function implements

involves

involves

involves

involves

involves

Actors, Components
Visualize data in live during flight

- Display acquired HD video in live
- Display multi-spectral image in live
- Display thermal image in live
- Visualize all collected mission data
- Visualize substance level in live
Models add rigor to need expression / solution description

Models enable automated processing
Requirements

Model elements
A model requirement can formalize a textual requirement and explicit its effects and ramifications.
Some expectations (environmental, regulations, etc.) are easier to express with textual descriptions.
Some expectations on a model element at a given engineering level do not require a formal modeling (which is left to subsystem design)
Tooling support
Coupling models and textual requirements
RAT (The Reuse Company) authoring in Capella
Capella in Siemens TeamCenter via Obeo SMW
Certain model elements can be considered as requirements.
3. Contracts between engineering levels: workflow
Solution

Need

Level N

Direct allocation

Textual Requirements
1. Elicitation of model and textual requirements on the system
2. Architecture description specifies with the adequate level of detail how the system works and what is expected from each constituent.

The goal here is to prepare the contracts for all subsystems and guarantee their proper integration.
3. The context of a given system constituent is entirely computed (anything contributing to the definition of this constituent including allocated Functions, interfacing Components, etc.)

4. Textual requirements are created when needed, in addition to the model requirements: legal, non-functional, additional specification of internal expected behaviour.
Tablet is a constituent of a drone-based system

Tablet is the (sub)system of interest
Model-based workflow favors co-engineering over the traditional differentiation between “customer” requirements and “system” requirements.
4. (Happy) consequences on V&V and incremental development strategy
Textual Requirements

Model Requirements

Verification and validation

(Derived, reconstructed link)
System-level V&V procedures

Definition of increments with expected functional chains (user stories)

Vertical slices of architectural design across need and solution models

System architectural design

Subsystems, software, etc.
5. Instanciated workflow

Two years, 30 persons
Classic scheme, rolled out by increments

System Engineering data package
- SYSTEM BLACK BOX
- SYSTEM GLASS BOX

Sub-System Engineering data package
- SUB-SYSTEM BLACK BOX
- SUB-SYSTEM GLASS BOX

System delivery
- SYSTEM VERIFIED
- SYSTEM INTEGRATED

Sub-System delivery
- SUB-SYSTEM VERIFIED
- SUB-SYSTEM INTEGRATED

x8

Slide adapted from an original content created by the project team
Identical pattern at all levels
Automated transitions

System Logical Architecture
LA functional chains
System Internal Data

SUB-SYSTEM BLACK BOX
READ ONLY input contract

SYSTEM GLASS BOX

Sub-system Context
SA functional chains
External Data

Requirements
System verification: enhanced progress monitoring

Content removed
Test results: models facilitate analyses

Content removed
Tooling support
Managing increments with Capella
Associate functional chains to increments

Compute, visualize and compare the footprints in terms of expected components and functions
Focus on the dedicated tooling
Preview of functional increments footprints

Release management viewpoint:
Automated visualization of versions
Preview of functional increments footprints

Developed Version 1
Available elements in BLUE
Preview of functional increments footprints

Developed Version 2
Available elements in CYAN
Preview of functional increments footprints

Developed Versions 1 & 2
Common available elements in GREY
Resources
Public forums, webinars and Youtube channel

https://polarsys.org/forums/index.php/f/10/

https://www.youtube.com/playlist?list=PLfrEYVpSGVLxEFRODSWUTP8N5i3NTG4o-
Capella website

https://polarsys.org/capella/index.html
Capella Day
MUNICH 2019
bit.ly/CapellaDay2019
Sept. 16th 2019
Thank You! Questions?

Capella website:
http://www.polarsys.org/capella/
LinkedIn  
https://www.linkedin.com/groups/8605600
Twitter  
https://twitter.com/capella_arcadia
Arcadia forum:
https://polarsys.org/forums/index.php/f/12/
Capella forum:
https://polarsys.org/forums/index.php/f/13/
IFE model & doc.:
http://www.polarsys.org/capella/start.html
www.thalesgroup.com