Architecture-centric System & Software Engineering

ARCADIA: A technical Approach to enforce Engineering

jean-luc.voirin@fr.thalesgroup.com
Contents

- Who is Thales?
- System engineering: limits of existing Practices
- ARCADIA: Model-based as an answer for real life
  - Multi viewpoint Co-engineering & trade-off analysis
  - Multi-level Co-engineering
  - Securing and optimising IVV (Integration, verification, validation)
  - Adaptation to Real-Life Projects
- ARCADIA specific tooling support
WHOEVER SAFETY AND SECURITY ARE CRITICAL, THALES DELIVERS. TOGETHER, WE INNOVATE WITH OUR CUSTOMERS TO BUILD SMARTER SOLUTIONS. EVERYWHERE.

**Employees**

65,000 (workforce under management at 31 Dec. 2012)

**Global presence**

56 countries

**Research and development**

2.5 billion euros (approx. 20% of revenues)
Global leadership

N 1 worldwide
- Payloads for telecom satellites
- Air Traffic Management
- Sonars
- Security for interbank transactions

N 2 worldwide
- Rail signalling systems
- In-flight entertainment and connectivity
- Military tactical radiocommunications

N 3 worldwide
- Avionics
- Civil satellites
- Surface radars

€14 billion in revenues
System engineering: limits of existing Practices
The System Engineering « ecosystem » Challenge: Collaboration

Customer User

System Architects

Sub-Systems, Equipment, Parts... Architects

Software Teams

Hardware Teams

Integration, Verification, Validation, Qualification, Teams

Performance, Safety, Security, Interfaces, Support, RAMT *

And all others...

• Partners & sub-contractors
• Regulation
• Production,
• Delivery & Deployment,
• Logistics, Support,
• Training,
• Disposal

* RAMT: Reliability, Availability, Maintainability, Testability

This document and any data included are the property of Thales "Company" S.A. They cannot be reproduced, disclosed or used without the company's prior written approval.
Issues in current main Practices

* RAMT: Reliability Availability Maintainability Testability

This document and any data included are the property of Thales "Company" S.A. They cannot be reproduced, disclosed or used without the company's prior written approval.
Textual Requirements-driven Engineering limits

- late discovery of design issues during IVVQ

User Textual Requirements → Tests description

Un-formalised Design

Requirements

Product Breakdown

Sub-contractor textual Requirements

Own Design

Coherency?
Seven Deadly Sins* in System Engineering

“Full Requirements driven approach weaknesses”

Bad understanding of CONOPS/CONUSE**

Incoherent reference & decisions between engineering specialties

Poor continuity between engineering levels

Late discovery of problems in definition & architecture

Underestimated Architectural Design impact / benefit

No anticipation of IVVQ, no functional mastering

No justification, no capitalisation for reuse/product line

* Deadly Sins: péchés capitaux
** CONcepts of OPerationS
ARCADIA: Model-based as an answer for real life
Modelling & Simulation must support the Engineering ecosystem efficiency

One end to end Method adapted to each domain

One Need Definition for all

Need & Architecture driving IVVQ

One consistent and checked Architecture definition

Specialities Know-how confronted to Architecture
A tooled method devoted to systems & (SW, HW) architecture engineering.

- to understand the real **customer need**,  
- define and **share** the system **architecture** among stakeholders,  
- **early validate** system design and **justify** it,  
- ease and **master IVVQ** (Integration, Validation, Verification, Qualification).

- **Model-based** and Tool-supported  
- Supporting **Collaboration** and **co-engineering**  
- Adopting **user’s language**, not the opposite  
- Open to **domain-specific** added value  
- Adapted to several **lifecycles**, workshares...  
- Dealing with **complexity & size**
Method Steps: formalising & sharing Need & Solution

User Need
What _the users_ of the system need to accomplish

System Need
What the system has to accomplish for the Users

Notional Solution
How the system will work so as to fulfil expectations

Final Solution
How the system will be developed & built

Sub-contractors input
Self protection System Example (bird eye)

Operational Analysis

Data & Interface Analysis

System Functional Analysis

Dynamic Behaviour

Logical & Physical Architectures
Command & Control System Example (partial)

Part of Physical Architecture
Comparing Approaches for Requirement management

Textual Requirements driven

User Textual Requirements
Un-formalised Design
Product Breakdown
Sub-contractor textual Requirements
Own Design

Tests description
Coherency?

Textual Reqs + Model driven

Requirements
Need
Solution
Engineering Specialities Assessment
Product Breakdown
Sub-contractor Requirements
Model
Own Design

(derived, reconstructed link)
Tests description

Collaborate!
Agility and flexibility for SE require

- Scalable, adaptable method (e.g. ARCADIA), supporting following features:
  - Multi viewpoint trade-off analysis
  - Early validation in short decision loop,
  - Multi-level impact analysis
  - Toolied support of ecosystem collaboration
Multi viewpoint Co-engineering & trade-off analysis
Early Validation: Specialities Know-how confronted to Architecture

Multi-viewpoint trade-off Analysis (see ISO 42010 standard)

- Safety, RAMT
- Security
- Performance
- Interfaces
- Product line
- Integration/V/V
- weight
- Cost...

Specialty engineering: safety, perf, interface, …
Confrontation rules for multi-viewpoints trade-off
Example of Modelling & Validation:
On-board Electrical Power System

Energy & Thermal system
Of a commercial Aircraft
Three interleaved, multi-physics Models

**Power Model**
- Generation
- Distribution

**Thermal Model**
- Conditioning
- Pressurisation
- Equipments Cooling

**Command & Control Model**
(coming soon)

- Electrical Generators, Loads, converters, bus bars...

- Turbines, Compressors, heat exchangers, valves...

- Electrical power

- Thermal Power

- Interfacing & computing units...

- Data
Power Systems Example: dedicated Viewpoints

**Power & Thermal performance**
depending on flight phase consumption, incl. Overloaded components detection based on power computation, linked to thermal Model

**Safety / Integrity**
incl. Failure containment, redundancy rules & analysis, failure scenarios & propagation, monitoring efficiency, shielding...

**Reliability & Availability**
incl. Reliability computing, reconfiguration issues, flight delay...

**Spatial (3D) arrangement**
Early identification of spatial arrangement constraints impacting the architecture

And also: Mass, Cost, Reliability...
Multi-level Co-engineering
Support of consistency between engineering levels

- **Automated Transition between Engineering Levels**
  - Iterative, conservative
  - Coherency control

- **Mastering complexity through multiple abstraction levels**
Early Validation: How to validate Architecture Vs Need

Need consistency & coherency
Traceability links

Impact Analysis:
Architecture Vs operational & functional Need Confrontation
Traceability & implementation links

Operational Analysis:
Requirements

System Functional & NF Need
Model

Logical Architecture
Model

Physical Architecture
Model

Product Breakdown

Architecture Vs Non-functional Need Confrontation

Viewpoint analysis
Traceability & implementation links

Solution
Components

Evaluation Rules
ViewPoints
Functions
Safety
Security
Performance
Interfaces
IVVQ, Product Line, Cost…
Securing and optimising Integration, verification, validation
Using ARCADIA Engineering Models to master IVVQ

**Operational Need, Functional Contents**

- Define IVV Strategy
- Master Development Ups and Downs
- Control Maturity of Deliveries
- Optimize IVVQ Globally (incl. Enabling Systems & Test Means)

**System Components**

**Test Means & Benches**
Definition of Test & IVV Scenarios

Deliveries definition based on operational & functional capabilities

Definition of Test benches & simulation Needs

Impact analysis
• Need evolution
• Component delivery delay
• Integration trouble...

Non-regression optimisation & mastering
Adaptation to Real-Life Projects
**ARCADIA goes far beyond Formalisms & languages:**

- **Method** defining full model design & conformance rules
  - How to define elements
  - How to link & relate them to each other
  - How to justify & check definition

- **Operational** Analysis & Capability integration

- **Modelling** Viewpoints for non-functional constraints support
  - (Safety, Performance, HF, RAMST, Cost...)

- **Semantic architecture Validation** through Engineering Rules formalisation

- Yet compatible & interoperable with most standards
Adaptation to real-Life LifeCycles: example of functional analysis

Top-down

1-

2-

Use-Cases-driven

1-

2-

3-

Bottom-up

1-

2-

Operational-driven

Functional chain-driven

1-

2-

3-
Examples of Use of ARCADIA Method in THALES

Critical Information Systems
- Ground exploitation systems
- Command & Control (air, sea, railways...)
- Large secured Communication networks...
- Satellite control networked ground stations

Embedded Systems
- Combat Systems & Equipment (radar, Self Protection, optronics...)
- Mission Systems (air, sea, ground)
- Satellite constellations
- Avionics Suites
- Computing Systems
- Electrical Power Systems
- Thermal cooling Systems
- Railways signalling Systems
- Air Traffic Management Systems
ARCADIA specific tooling support
Melody Advance solves Weaknesses of COTS* or OSS*:

- ARCADIA Method support & guidance for modelling
  - Method Steps, encyclopedia, rules, diagrams...
- User-oriented Semantics
  - Engineer concepts rather than abstract/profiled language
- Support for « modelling in the large »
  - Performance on large models, ergonomy, modelling aids...
- Support for Viewpoint extensions, modelling & analysis
  - Model extensions, diagrams extensions, viewpoint management...
- « semantic » Import/export capabilities (excel, SysML, AF, ...)

- Yet ARCADIA is also deployed using other tools
  - Excel/Access, Rhapsody, System Architect/DoDAF...
  - with reduced capabilities, however

---

* COTS: Commercial Off The Shelf
* OSS: Open Source Software
Overview of the Modeling Workbench Main Features

Edition Tools
Layered diagrams, Tables, Editors
Overview of the Modeling Workbench Main Features

Edition Tools
Layered diagrams, Tables, Editors

Embedded Methodological Guide
Overview of the Modeling Workbench Main Features

**Edition Tools**
Layered diagrams, Tables, Editors

**Semantic browser, Model check, Etc.**

**Consistency**
- [ ] Components (2 items)
- [ ] Datalows (16 items)
  - [ ] Acquire meteo data (Function) shall be realized by Capture temperature (Function): both contain. 
  - [ ] Both bounds of Functional Exchange should realize bounds of the realized FunctionalExchange. 
  - [ ] Both bounds of Functional Exchange should realize bounds of the realized FunctionalExchange. 
  - [ ] Elaborate current situation (Function): shall be realized by Transmit data (Function): both contain.

**Semantic browser**
- [ ] Serve as a guide for modelers.
- [ ] Assistance in creating models.

**Model Analysis**
- [ ] Embedded Methodological Guide
Overview of the Modeling Workbench Main Features

**Logical Architecture**

- Semantic browser
- Model check, Etc.

**Edition Tools**
- Layered diagrams, Tables, Editors

**Physical Architecture**

- Logical Architecture

**Embedded Methodological Guide**

- Iterative Transition Tools
- Traceability, Generation

**Model Analysis**
- Semantic browser, Model check, Etc.
Overview of the Modeling Workbench Main Features

Edition Tools
Layered diagrams, Tables, Editors

Embedded Methodological Guide

Edition Tools
Layered diagrams, Tables, Editors

Model Analysis
Semantic browser, Model check, Etc.

Iterative Transition Tools
Traceability, Generation

Model Analysis
Semantic browser, Model check, Etc.

Embedded Methodological Guide

Edition Tools
Layered diagrams, Tables, Editors

Iterative Transition Tools
Traceability, Generation

Modularity & Reuse
Librairies, Patterns, Etc.

Modularity & Reuse
Librairies, Patterns, Etc.

Overview of the Modeling Workbench Main Features

Edition Tools
Layered diagrams, Tables, Editors

Embedded Methodological Guide

Iterative Transition Tools
Traceability, Generation

Modularity & Reuse
Librairies, Patterns, Etc.
Overview of the Modeling Workbench Main Features

Edition Tools
- Layered diagrams, Tables, Editors

Model Analysis
- Semantic browser, Model check, Etc.

Iterative Transition Tools
- Traceability, Generation

Modularity & Reuse
- Libraries, Patterns, Etc.

Model Monitoring
- Progress, metrics

Embedded Methodological Guide

Main Features
- Model Analysis
  - Semantic browser, Etc.
- Embedded Methodological Guide
- Iterative Transition Tools
  - Traceability, Generation
- Edition Tools
  - Layered diagrams, Tables, Editors
- Modularity & Reuse
  - Libraries, Patterns, Etc.
- Model Monitoring
  - Progress, metrics

This document and any data included are the property of Thales “Company” S.A. They cannot be reproduced, disclosed or used without the company’s prior written approval.
Overview of the Modeling Workbench Main Features

**Extensibility**
New diagrams, new layers, M2 extensions, Etc.

**Edition Tools**
Layered diagrams, Tables, Editors

**Embedded Methodological Guide**

**Model Monitoring**
Progress, metrics

**Modularity & Reuse**
Librairies, Patterns, Etc.

**Iterative Transition Tools**
Traceability, Generation

**Model Analysis**
Semantic browser, Model check, Etc.
Consortium & initiative
- Industrial end-users (large companies) to deploy and adapt
- SMEs for consulting & training, extensions development
- Tool vendors to connect to their own tools
- Academics to increase capabilities

ARCADIA published and freely available
- Books, standardisation...

CAPELLA modelling tool as Open Source
- New name for Open Source Melody Advance
- ECLIPSE PolarSys context
A strong lever for Engineering Transformation

Field-proven in real industrial situations

Leading to a better mastering of products, costs & cycles

Improving Architecture Quality & sharing, and IVV (Integration Verification Validation) mastering

Deployed or under adoption in various divisions,

including industrial partnerships