Large-Scale DS(M)L Deployment in Thales

ITSLE Workshop – Sept. 15th, 2014

Stéphane Bonnet, Daniel Exertier, Benoît Langlois
Melbourne, June 2014
Modelling workshop for a new operational project

Toronto, July 2013
Singapore, Feb. 2014
Agenda

**Who, What**
Thales, method & workbench for System, Software and Architecture definition

**How, How+**
Rationale for DSL, technological choices, Sirius, Kitalpha

**Deployment**
Key enabling factors
Who, What
Whenever critical decisions need to be made, Thales has a role to play. In all its markets — aerospace, space, ground transportation, defence and security — Thales solutions help customers to make the right decisions at the right time and act accordingly.

World-class technology, the combined expertise of 65,000 employees and operations in 56 countries have made Thales a key player in keeping the public safe and secure, guarding vital infrastructure and protecting the national security interests of countries around the globe.

**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)

**A balanced revenue structure**

<table>
<thead>
<tr>
<th>Defence</th>
<th>Civil</th>
</tr>
</thead>
<tbody>
<tr>
<td>55%</td>
<td>45%</td>
</tr>
</tbody>
</table>

**Revenues in 2012**

€ 14.2 billion euros

**Shareholders**

(at 31 May 2013)

- French State 27%
- Dassault Aviation 26%
- Float 47%
  - of which employees 3%
Dual markets
Military & Civil

AEROSPACE

SPACE

GROUND TRANSPORTATION

DEFENCE

SECURITY
Thales : A Wide Spectrum of Complex Systems

1. Payloads for telecom satellites
2. Air Traffic Management
3. Sonars
4. Security for interbank transactions

1. Rail signalling systems
2. In-flight entertainment and connectivity
3. Military tactical radiocommunications

1. Avionics
2. Civil satellites
3. Surface radars

€14 billion in revenues
Need for an Engineering Transformation

Market

- Systems are more complex
- Scope changes (from equipments to integrated systems)
- Do more... cheaper and faster

Model-Based Systems Engineering

Objectives

- Better quality of the systems: Integration, seamlessness, coherency, traceability
- Early validation
- Better productivity of engineering activities
- Collaborative engineering
- Best practice & know-how capitalization

Capella
How to improve quality, productivity, agility and flexibility of overall engineering?
How to improve quality, productivity, agility and flexibility of overall engineering?

- Eco-system wide collaboration
  - A single architecture reference
How to improve quality, productivity, agility and flexibility of overall engineering?

- Eco-system wide collaboration
  - A single architecture reference

- Complexity mastering
  - Multi-level engineering
  - Separation of concerns
How to improve quality, productivity, agility and flexibility of overall engineering?

- Eco-system wide collaboration
  - A single architecture reference

- Complexity mastering
  - Multi-level engineering
  - Separation of concerns

- Early validation
  - Integrated specialty engineering
  - Trade-off analysis
  - Short decision loop
How to improve quality, productivity, agility and flexibility of overall engineering?

- Eco-system wide collaboration
  - A single architecture reference

- Complexity mastering
  - Multi-level engineering
  - Separation of concerns

- Early validation
  - Integrated specialty engineering
  - Trade-off analysis
  - Short decision loop

- Mastering transitions
  - Information refinement
  - Coherency maintenance
  - Multi-level impact analysis
Capella: A DSL++ Supporting the Arcadia Method
Critical Information Systems

Ground Exploitation Systems
Command & Control (air, sea, railways...)
Large secured Communication Networks...
Satellite Control Networked Ground Stations

Embedded Systems

Combat Systems (Radar, Self Protection, Optronics...)
Mission Systems (Air, Sea, Ground)
Satellite Constellations
Avionics Suites
Computing Systems
Electrical Power Systems
Thermal Cooling Systems
Railways signalling Systems

Users worldwide

Operational Projects

Australia
Canada
Germany
Italy
France
The Netherland
United Kingdom
Singapore
How
EMF outside Java/ Eclipse 😊

- Core Metamodels: MD SoFa, UML, etc.
- Core Patterns
- Model navigation
- Model checking
- Wizards
- Diagram generation
- Etc.

- Core patterns adaptation
- Domain-specific patterns
- Libraries
- Framework

Tools Infrastructure (80% generated)
Tool Function Variation Tool
Engineering tools (navigation, checking, wizards...)
Standard

Modelling Platform Variation Tool
UML version

Tools Extension (20% developed)
Graphical DSML

- Tight fit with exact domain and needs
- Short learning curve thanks to familiar terms and concepts
- Complete freedom in expressivity (language and representations)

Profiling UML/SysML

- Usually well-accepted in Thales by Software engineers, less by System engineers
- Language complexity
- Limited capabilities regarding diagram customizations
- Restraining the usage of a UML tool to selected scope of concepts is difficult

Graphical DSML or UML/SysML Profile?

Graphical DSML

- Cost ... until Sirius!

Profiling UML/SysML

- Easier interoperability with standards
- Widely understood (or misunderstood)
Thales previous experiences with UML Profiling

- Poor adoption by system engineers
- Meta-models constrained by UML concepts
- Representations constrained by existing UML diagrams

Need for graphical DSMLs

- Freedom both in language and representation
- Heavier and more technical (GMF) developments
- Originally 2 or 3 foreseen modeling workbenches

Separation workbench / business concerns

- Generic infrastructure for model management and representations
- Focus on business added-value
- Capitalisation
What is Sirius?

Provides the **specification tools** to define, test, and distribute **graphical modeling workbenches** (diagram, table and tree editors) on top of the Eclipse platform, **with little technical knowledge**

Provides the associated **runtime environment**
What is Sirius?

Now Open Source!

Workbench Configuration

- My
  - Planning
    - Talks and Tracks
      - Default
        - Talk Node
        - Section Tools
    - Tracks
      - Talk To track
      - Track Container
        - Track sub nodes
        - Gradient white to light gray
    - platform:/resource/org.eclipsecon.conference
  - conference
    - Conference
  - Track
  - Talk

Runtime

- Technology
  - New talk
  - New talk
  - New talk

- Egit Rocks

- Introducing Sirius

- Modeling
  - Introducing Sirius
**2007** First Obeo/Thales prototype to validate the concepts

**2008** Specification and development of Sirius foundations

**2009 - Present** Mature product

**2008** Thales Capella modelling workbench

**2009** First operational pilot projects, launch of Obeo Designer, based on Sirius

**Sirius: A Collaboration Thales / Obeo**
How+

A comprehensive development environment for enriching DSMLs
Reminder: Multi-Viewpoint Analysis is the DNA of Arcadia

Thales Corporate Engineering provides the Capella workbench and a few generic-purpose viewpoints.

Development and runtime environment for viewpoint-based modelling workbenches.

Thales BUs often need to develop specialty engineering viewpoints.
ISO/IEC WD3 42010 (2010-06-08)

“An architecture framework establishes a common practice for creating, interpreting, analyzing and using architecture descriptions within a particular domain of application or stakeholder community.”
ISO/IEC WD3 42010 (2010-06-08)

“An **architecture description** includes one or more architecture views. Each architecture view (or simply, view) addresses one or more of the system concerns held by the system’s stakeholders.

Each **architecture view** expresses the architecture of the system-of-interest in accordance with an **architecture viewpoint** (or simply, viewpoint). Each viewpoint frames one or more system concerns. Each concern can be framed by one or more viewpoints.

Each view is governed by its viewpoint: the **viewpoint** establishes the conventions for constructing, interpreting and analyzing the view to address concerns framed by that viewpoint. Viewpoint conventions can include languages, notations, model kinds, design rules, and/or modelling methods, analysis techniques and other operations on views.”
Viewpoint-Based Architecture Description of Complex Systems

Viewpoints
- Metamodels
- Rules
- Representations
- Tools
- Services

Stakeholders

System of interest

Mass

Cost

Architecture Description
Kitalpha: Development and Execution Environments

Architecture Framework Development

- Description: AF DSL
- Generation: AF Generation
- Packaging: Architecture Framework

Viewpoint Development

- Description: Viewpoint DSL
- Generation: Viewpoint Generation
- Customization: Additional Development
- Packaging: Metamodels, User Interface, Diagram, Tools

Viewpoint-based modelling workbench
Focus on Kitalpha Viewpoint DSL

Development

- ecore model
- Text file generation (manual)
- Text file generation (automatic, manual)
- Textual viewpoint description [viewpoint]*.vptext
- User interface
- Diagram
- Tools

packaging

Deployable viewpoint

installation

Use

- Data
- Representation
- Services
Kitalpha Viewpoint DSL: Example

Architecture Description

ComponentSampleAF

ComponentSampleFramework

ComponentSample

uses

uses

uses

ComponentSampleQualityAssessment

ComponentSamplePerformance

ComponentSampleSafety

uses

uses

uses

ComponentSampleSafetyPattern

Extensions
Kitalpha Viewpoint DSL: Example

Viewpoint QualityAssessment {
    name: "QualityAssessment"
    Data QualityAssessment.data
    UI QualityAssessment.ui
    Diagrams QualityAssessment.diagram
    Services QualityAssessment.services
    Build QualityAssessment.build
    Configuration QualityAssessment.conf
}
import external "http://www.polarsys.org/kitalpha/ComponentSample"

Data ComponentSampleQualityAssessment.data {
    Class QualityAssessment {
        description: "Quality Assessment"
        icon: "QualityAssessment.png"
        extends ComponentSample.AbstractComponent
        superClass external ComponentSample.ComponentElement
        Attributes:
            maturityLevel type ecore.EString
            confidenceLevel type ecore.EEnumerator
                values ( "Not Assessed", Low, Medium, High )
            assessed type ecore.EBoolean
        Associations:
            basedOn refers [0,*] QualityAssessment
            context refers [0,*] external ComponentSample.ComponentElement
            measures contains [0,*] QualityMeasure
    }
    Class QualityMeasure {
        icon: "QualityMeasure.png"
        superClass external ComponentSample.ComponentElement
        Attributes:
            criterion type ecore.EString
            measureValue type ecore.EInt
    }
}
```java
QualityAssessment.ui {
    UI QualityAssessment_QualityAssessment {
        label: "Quality Assessment"
        Container QualityAssessment_QualityAssessment_Section {
            Container QualityAssessment_QualityAssessment_AttributeGroup {
                label: "Quality Assessment Attributes"
                Field maturityLevelField label: "Maturity Level" type text, mapped-to QualityAssessment.data.QualityAssessment.
                Field confidenceLevelField label: "Confidence Level" type radiobox, mapped-to QualityAssessment.data.QualityAssess
                Field assessedField label: Assessed type checkbox, mapped-to QualityAssessment.data.QualityAssessment.assessed
            }
            Container QualityAssessment_QualityAssessment_AssociationGroup {
                label: "Quality Assessment Associations"
                Field basedOnAssociation label: "Based On" type multipleChoiceList, mapped-to QualityAssessment.data.QualityAss
                Field contextAssociation label: Context type multipleChoiceList, mapped-to QualityAssessment.data.QualityAssess
                Field measuresAssociation label: Measures type multipleChoiceList, mapped-to QualityAssessment.data.QualityIsse
            }
        }
    }
    UI QualityAssessment_QualityMeasure {
        label: "Quality Assessment"
        Container QualityAssessment_QualityMeasure_Section {
            Container QualityAssessment_QualityMeasure_AttributeGroup {
                label: "Quality Measure Attributes"
                Field criterionField label: Criterion type text, mapped-to QualityAssessment.data.QualityMeasure.criterion
                Field measureValueField label: "Measure Value" type text, mapped-to QualityAssessment.data.QualityMeasure.measure
            }
        }
    }
}
```
Class FieldDevice {
    description: "Field Device"
    icon: "FieldDevice.gif"
    extends pa.PhysicalComponent, cs.PhysicalLink, cs.PhysicalPort
    Annotation "http://www.thalesgroup.com/mde/documentation" {
        key: summary value: "Non-functional requirement"
    }
    Attributes:
        bandwidth type.ecore.EInt
        Annotation "http://www.thalesgroup.com/mde/documentation" {
            key: note value: "Value in bps"
        }
        power type.ecore.EInt
        Annotation "http://www.thalesgroup.com/mde/documentation" {
            key: note value: "Value in Watt"
        }
        weight type.ecore.EDouble
        Annotation "http://www.thalesgroup.com/mde/documentation" {
            key: note value: "Value in Kg"
        }
        isFieldDevice type.ecore.EBoolean
}

[Physical Component] PC 2

Field Device
- is a field device
- Description
- NF Requirements
  - Non Functional Requirements Attributes
    - Bandwidth in bps: 12440
    - Power in Watt: 1200
    - Weight in Kg: 

Physical System
- PL 1
- PL 2
- PC 1
- PC 2
- PC 3
- [Physical Component] PC 2
- [System CI] System
- [Architecture] EPRS
- Capabilities
- Extensions
- Expert
Viewpoint DSL is only one of multiple Kitalpha components. More information to come on the Kitalpha project page

https://www.polarsys.org/projects/polarsys.kitalpha

benoit.langlois@thalesgroup.com

Sirius is now part of Eclipse’s Modelling Distribution and is the foundation of Ecore Tools 2.0. It is receiving a warm welcome from the Community

http://www.eclipse.org/sirius
Deployment
Strong support from management (*)

Important and renewed funding

Group recommendation

(*) Double-edged sword
Strong involvement of end-users

Initial definition of the DSL (Do it right!)

Clear and democratic process for the definition of evolutions and roadmaps

Business-driven maturation of low-TRL solutions
Tool maturity and added-value

Stability + Ergonomics + Performance are MANDATORY

Focus on what is missing in COTS

User experience with diagram interactions is essential

Engineers spend hours in front of the tool every day!
Deployment Enabling Factors: Ergonomics

Functions = Green

Components = Blue

Interfaces = Pink
<table>
<thead>
<tr>
<th>Deployment Enabling Factors: Capella Key Features</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Brush diagram layouts</strong></td>
</tr>
<tr>
<td><strong>Transition System Subsystem</strong></td>
</tr>
<tr>
<td><strong>Live collaboration</strong></td>
</tr>
<tr>
<td><strong>Replicable elements</strong></td>
</tr>
<tr>
<td><strong>Progress monitoring &amp; model review</strong></td>
</tr>
<tr>
<td><strong>IVV and Product Line viewpoints</strong></td>
</tr>
<tr>
<td><strong>Automated contextual diagrams</strong></td>
</tr>
<tr>
<td><strong>Key added-value features</strong></td>
</tr>
<tr>
<td><strong>Unsynchronized diagrams</strong></td>
</tr>
<tr>
<td><strong>HTML output</strong></td>
</tr>
<tr>
<td><strong>Fast Linker</strong></td>
</tr>
<tr>
<td><strong>Validation profiles</strong></td>
</tr>
<tr>
<td><strong>Batch quickfixes</strong></td>
</tr>
<tr>
<td><strong>Model Patterns</strong></td>
</tr>
<tr>
<td><strong>Semantic delete with preview</strong></td>
</tr>
<tr>
<td><strong>IncQuery &amp; Acceleo requests</strong></td>
</tr>
</tbody>
</table>

- Brush diagram layouts
- Transition System Subsystem
- Live collaboration
- Replicable elements
- Progress monitoring & model review
- IVV and Product Line viewpoints
- Automated contextual diagrams
- Key added-value features
- Ununsynchronized diagrams
- HTML output
- Fast Linker
- Validation profiles
- Batch quickfixes
- Model Patterns
- Semantic delete with preview
- IncQuery & Acceleo requests
Methodological support

The tool is just a mean to implement the engineering approach

The availability of the embedded methodological guidance has had a direct impact on operational deployment

Coaching

Initial training is not enough, ugly architectures with models are still possible!

Coaching: Definition of modeling objectives and strategies, stopping criteria, identification of achievable benefits, etc.
Development of a MBSE community

Regular thematic workshops (method experts, tool/modelling experts and operational practitioners)

Identification of MBSE “champions” in Business Units

Favor the exchange of experiences

Favor the sharing of add-ons / viewpoints
Next steps
Thank you for your attention!

Any Questions?