Model-driven engineering – solutions for software intensive systems

Véronique Normand
Software Research Group
THALES Research & Technology
Management of Complexity and Separation of problems

Every 5 years, the size of SW systems is multiplied by 5 to 10.

Functional aspects
Quality of Services
Technological platforms

3 different aspects
3 different life cycles

Every 5 years, the size of SW systems is multiplied by 5 to 10.

Business Model
BM 2

Air Traffic Management v1, Air Traffic Management v2 ...
Functional requirements quasi-stability

System lifetime
10-30 years

Figure 18: Large scale distributed systems have long lifecycles
Why Model-driven Engineering?

Strategic objectives

- Safeguard from future technological evolutions.
- Capitalise our knowledge (technical or domain).
- Improve the development and maintenance of our systems (quality and productivity).
- Facilitate subcontracting / the collaboration with our partners during full life cycle development.

MDE : technological answer

- Separation of concerns and aspects (functional, quality of services, domain, technological).
- Formalise our knowledge in precisely defined language.
- Automatise development activities (traceability, impact analysis, model check, test, doc generation, code generation, ....)
- Build reusable assets including domain, technical and technological Engineering aspects.
MDSysE: a model-driven solution for software intensive systems
Building a model-driven systems engineering methodology

Model-driven methods
- What model work products?
- Concepts & notation, modelling rules
- Traceability principles and means
- Verification rules and means
- Automation rules
- Configuration mgt
- Tooling

LEVERAGE INFLUENCE

Standard modelling languages
- UML 1.4
- UML 2.0
- QoS UML 2 Profile
- SPT UML 1.4 Profile
- Testing Profile
- UML for SE Profile

Customer Requirements
- FBL
- ABL's
- System Requirements
- Component Requirements
- Subcontracted development of system components
- Subcontracted development of enabling products

Programme Management
- Technical Management
- Customers & Supplier Agreements
- Purchasing

Business Development Bids and Proposals

Subcontracted development of enabling products

Building a model-driven systems engineering methodology

This document and any data included are the property of Thales. They cannot be reproduced, disclosed or used without Thales’ prior written approval.

©THALES 2005. Template version 1.0

Research & Technology
A model-driven solution for software intensive systems

- Efficient, precise capture of systems information in an **integrated, modular** way.

- **Continuous transition to system development processes** (software and hardware development processes), with continuous traceability.

- Incremental, automated verification is for a significant part built in the model-driven process.

- **Integration & deployment** support targeting CCM middleware.

- **Automation** of a number of production and verification tasks through the MDSysE Tools.
Modelling environment extending Modelling COTS

- **Assisted edition:**
  - high-level system edition wizards, “correct by construction”
  - automated model initialisation & synchronisation
  - automatic view generation

- **Automated verification:**
  - Traceability mechanism
  - Impact analysis
  - Consistency checking
  - Quality metrics

- **CCM assembly, configuration and deployment support:**
  - PIM-PSM transformation
  - Automatic meta-data/configuration files and code generation
MDSysE Tooling: CCM assembly, configuration and deployment support

Smart GUIs

Component assembly

MDATM Context

PIMs
Conservative transformation

PSMs
others E2E CCM EET

Model Driven Benefits

Productivity Enhancement
Automatic Files and Model Generation
Consistency checks
Higher abstraction level
Separation of Concerns
Simplicity
Technological complexity managed by the tool
A single language
Smart GUIs

Model Repository

External View
Dynamic View
Implementation View
Deployment View

Consistent Views
One shot transformation

Consistency checks
Simplicity
Technological complexity managed by the tool
A single language
Smart GUIs

Research & Technology
The bigger picture: MDE as the integrating frame

Integration, configuration & deployment

Requirements Management

Verification & validation
Functional and QoS analysis & simulations
Model-based testing

Core system definition models

Planning & Mgt

Trade-off analyses & decisions

Specific analyses: safety

Documentation & visualisation

SW component Development

HW component development

Research & Technology
Challenges & strategy
Challenges

From MDA to MDE

- A landscape of multiple heterogeneous models for a system, in distinct technological spaces (UML, DB, XML, formal languages, etc.).
- Need for semantic bridging across languages and tools.

Durability / capitalisation of our methodological & domain developments (DSLs, verification rules, transformations, patterns, domain models, etc.)

- We need model repositories and tool builders to:
  - Facilitate sharing of models
  - Enable in-house methodology development, modular, reusable and adaptable according to programs context
  - Minimize tool-vendor lock-in

- Standardisation (QVT, SysML, QoS, MARTE, UPDM, etc.)
- Toward tool-independent assets based on metamodeling techniques and standards.
Challenges

Building sound and cost effective MDE methodologies

- Defining the “right” modelling frameworks for a domain
  - Adequate metamodels & syntaxes: relevant abstractions, granularity, with validation mechanisms that match the target system types and complexity, and the specific engineering concerns.
  - Example: modelling systems of systems

- Mastering production and exploitation costs of MDE
  - Scalability for design & analysis of complex systems
  - Legacy management
  - Adoption: decision model, MDD capability levels, TCM
Partners
- THALES
- CEA (French Atomic Energy Agency)
- INRIA (National Institute for Research in Computer Science and Control)

Common research programme
- Program started in March 2003
- Initial duration = 3 years

Objective
- Integration Research program on Model Driven and Middleware technologies
IST Integrated Project
2 years duration; 1500 person/months
Total Cost : 21 M€; total Funding : 11 M€
Co-ordination : THALES Research & technology
19 Partners :
  - End-users
    - THALES, IBM, France Telecom, Enabler, AS Aprote
  - Tool Vendors
    - Softeam, Adaptive, SINTEF, Imbus
  - Research labs and academia
    - INRIA, ESI, Univ Pierre et Marie Curie, UPM, Univ of York, IBM HRL, Fraunhofer
  - Dissemination
    - SIGS-DATACOM

3 Objectives :
  - **Objective A** : Develop a solution to enable a significant increase of software systems development productivity thanks to Model Driven Development (MDD);
  - **Objective B** : Lead its industrialisation;
  - **Objective C** : Ensure its successful adoption by the industry
Eclipse MDDi project

MDDi: Model-Driven Development Integration
A MODELWARE and CARROLL Research initiative

- Project goal: Extend Eclipse and develop an Open Source platform dedicated to the integration of MDD tools
- Organization around three themes:
  - Modeling tool interoperability
  - Semantics interchange via a description of modeling languages
  - Support for development processes and methodologies

- The project proposal to be official shortly (creation review today).
- Actors: Stephen Mellor (Mentor Graphics) - leader
  - Current committers: Adaptive, CEA-List, France Telecom, INRIA Nantes, LIP6, SINTEF, THALES, UPM
Answering these challenges

- A set of integrated research programmes
  - with Academic partners
  - with other industrial partners,
  - at national and international levels
- Proactive participation to standardisation bodies
- Tooling:
  - Open Source strategy
  - Close collaborations with tool vendors (IBM/Rational, Softeam)

Based on existing solutions, toward critical mass, integrated Research and Technology structures addressing the shared challenges
MDDi Platform outline

Semantic Binding Component

Tool Integration Component

- Action Semantic Editor
- Profile Design Tool
- Semantic Mappings
- Meta-model Registry
- ModelBus
- Adapter 1
- Adapter 2
- Adapter n

Eclipse Tools

- UML2
- EMF
- GEF

ANTLR
Velocity

External Tools
MDDi
Eclipse Platform & Tools

JFace

THALES
What is MODELWARE? Open Source Platform

- **COTS or Proprietary tools**
  - COTS generic tool
  - COTS generic tool
  - COTS generic tool
  - Domain specific tool

- **ModelBus**
  - Tool integration guide & toolkit
  - Other components

- **Tool infrastructure**
  - Modeling services repository

- **MDD Assets**
  - Modeling services repository
  - Other components
  - CCM
  - QoS
  - System engineering
  - MDD orchestration tool
  - QVT engine

- **Other components**
  - Tool integration guide & toolkit

**Step 1: Methodology Development**
- Methodology design tool
- Create / register assets
- Load assets

**Step 2: Software Development**
- Software Modeling Tool A
- Software Modeling Tool B
- Load assets

**Model Bus**
- Repository / Registry of assets

**Software Modeling**
- Tool A
- Tool B
Our mission: improve the Group competitiveness in Software Intensive Systems development

MIRROR Activities

- Standardisation (OMG)
- Speed-up transfert of academic results
- Applied research in collaborative frame

- MDD solutions (tools + method)
- Tool Vendors collaboration (Rational, ILogix, Softeam)

- Dissemination to BGs (support, coaching, training)
- High level experts on MDE
- Technical Change Management for BGs (TCM)