

Thales ATM Experience in using MDE

Anne De Goeyse¹, Gilles Blanc, Nathalie Guilloreau Thales RT

¹ Thales ATM 19 rue La Fontaine 9221 Bagneux

<mailto:anne.degoeyse@thalesatm.com>

<mailto:gilles.blanc@thalesatm.com>

²Thales TRT

<mailto:nathalie.guilloreau@thalesgroup.com>

Abstract. Thales ATM develops large Air Traffic Control systems that are managed within a product line. The lifetime of such system is more than 20 years; during those 20 years new features will be added to satisfy different customers and to be up to date in the offer of ATC features. The development productivity needs to be improved from 5 to 10 % to stay on the market. Thales ATM has set up a Model Driven Engineering (MDE) process and environment on one of its major project assigning the target of a productivity increase to this first project as well as being the basement of its future product line. The Thales ATM project has now finished the system architecture and the development is close to be launched. The use of MDE environment has been proved useful from a qualitative point of view. The return on investment will be evaluated at the end of the development phase.

Keywords: MDE, UML, methodology, systems integration, Product line, Lessons Learnt.

Introduction

Thales ATM provides Air Traffic Control systems that are large software systems with a lifetime of more than 20 years. It is very well known that the cost of development and maintenance of such large systems is high, difficult to master and less and less acceptable for the customers.

Thales ATM expectations from MDE are:

- Improve the productivity by:
 - Automating the model check, the document generation, the update of the traceability links.
 - Reducing the number of problems identified during the system integration phase by keeping the system components interfaces consistent.
 - Providing support by automation to the safety analysis.
- Facilitate the product line management by keeping the product and the systems produced from the basic product consistent and with the appropriate level of quality, safety and reliability.

Thales ATM carries out one of its major project applying in a real corporative environment the Model-Driven Engineering (MDE) approach. The objective of this project is to develop the new generation of ATC system.

MDE has been used today on the system design activity phase and will be used during the development phase.

The system developed in Java and C++ will run on PC/Linux network with support of a CORBA middleware

MDE Process and Tools environment

Traditional approach

Up to now the Thales ATM system development performs:

- Classical system specification, architecture and design activities restricted to functional analysis
- A code centric SW development.
- Manual verification activities including peer reviews, safety analysis and traceability on documents as shown in the following figure:

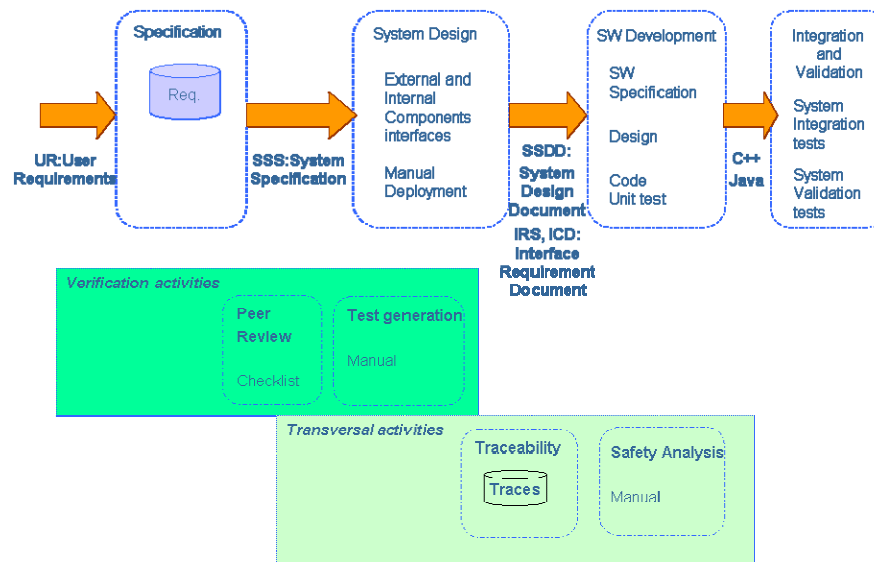


Fig. 1. Traditional development process

MDE approach

The new approach based on Model Driven Engineering (MDE) that has been set up on the project performs:

- System specification, architecture and design activities enhanced with the end user view (use cases) and the information view (data model)
- A model centric SW development: the model is the reference, the code is generated from the model
- Automated verification activities

The focus of the project today is on the system architecture and design which aim is to define the components that implement the system requirements and their deployment on the various system computers.

The system architecture and design uses MDSysE method and tools.

MDSysE in brief

MDSysE is used to perform the system Architecture and Design activity. It presents 4 views of the system:

- The contextual view which presents the system, its actors and the interfaces between the actors and the system
- The Logical view that breaks up the system into Logical Components (LC) to which the requirements of the system are allocated.

- The Physical view that breaks up the system into Physical Components (PC). A Physical Component implements one Logical Component or implement generic function like communication
- The End Product Breakdown Structure that enables to organize the system development into CSCIs (Computer Software Configuration Item)

The Unified Modeling Language (UML) is used in the different views. It has been enriched with MDSysE items: the Logical Components (LC) and the Physical Components (PC).

The following steps are performed to design the system:

1. The identification of the actors and their interfaces with the system is carried out in the context view.
2. The Logical Components (LC) are identified in the Logical model. A LC can be refined in more detailed LC. The refinement is automated: all information that needs to be within the lower level is automatically associated to the refined LCs.
The Logical model includes sequence diagrams that model the way the actors use the system.
3. The Physical Components (PC) are identified in the Physical view. The refinement from the logical level to the Physical level is automated as are the different level of logical refinements.
4. The CSCIs are identified in the End product Breakdown Structure (EPBS) view. The PCs are gathered into CSCIs to organize the development and to package the system.
5. The PCs are transformed in CORBA Components (CCM). The refinement generates the code needed for the CORBA platform and the CCM interface classes.
6. The CCM interface classes are imported to be used by the software development teams as reference to their development.

The different views are summed up in the figure below.

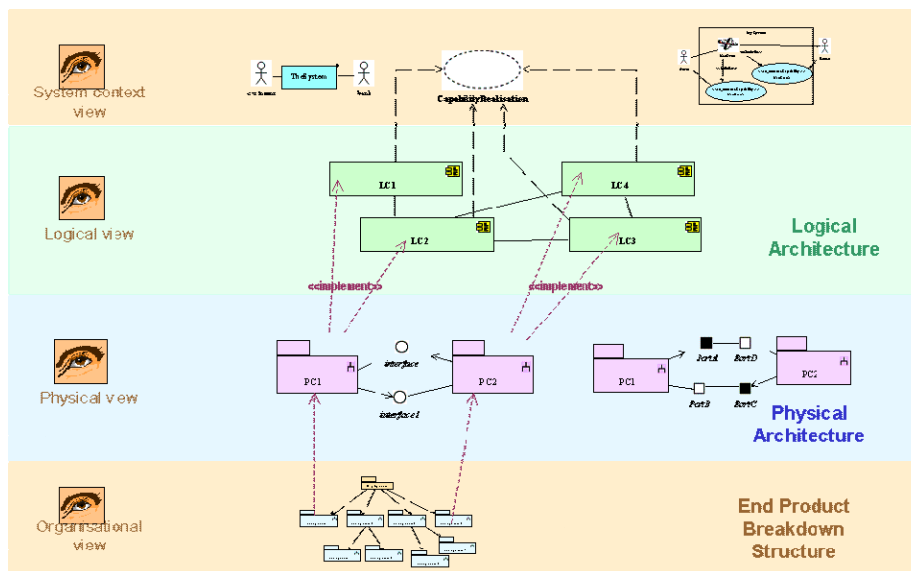


Fig. 2. MDSysE process steps

Verification activities

Beside the development activities, verification activities are performed in order to detect potential defects as soon as possible.

In order to increase the productivity and limit as much as possible the possibility of human errors, the verification activities are automated:

- The modeling rules defined on the project to homogenize the work done by different designers are checked
- Automatic generation of tests (CSCI Tests and Integration tests) from the use cases is under study
- The traceability links which are set within the model are analyzed in a traceability tool for impact and coverage analysis
- The safety analysis is facilitated and help is provided to:
 - Check that the safety requirements are implemented
 - Check that the safety requirements implementation does not jeopardize the project budget balance by giving too much constraints to the SW development
 - Check that the system after implementation of new features still complies to the safety requirements

To do this, the system model is decorated with the appropriate safety information and transformed into the safety model that enables to perform Fault Tree Analysis and Failures Mode and Effects and Criticality Analysis (FMECA).

The system design activities are summed up in the figure below.

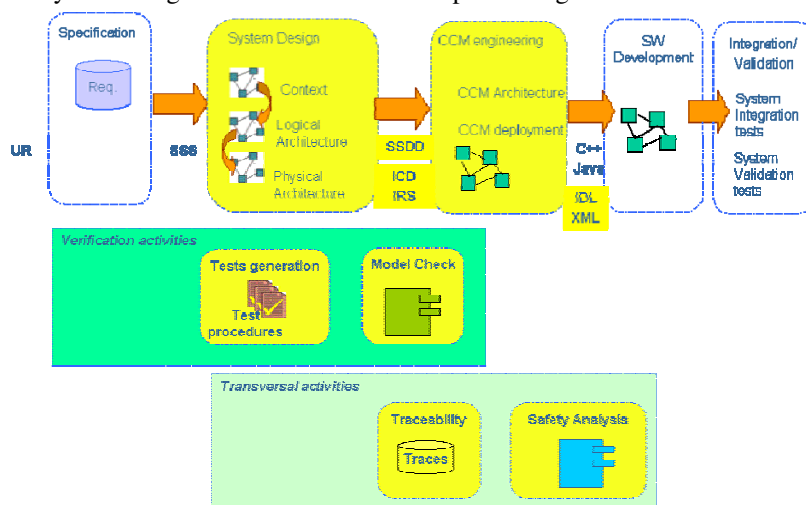


Fig. 3. Development and verification activities

Miscellaneous

Collateral activities need to be dealt with:

- The Change Management within a MDE approach
- The metrics to track the progress, to check the quality of the delivered product and to make sure the MDE approach implemented in the project environment is efficient

Change Management

The specific issue of MDE related to Change Management is to be able to manage parallel version of a system model (multi customer, incremental development) and to report the changes from one model to

another automatically. The solution chosen today to solve this problem is to work with one model including all the versions.

Metrics

The tracking metrics and the efficiency metrics used are not specific to the MDE approach:

- The tracking metric used is the use cases status metric.
- The metric to check the efficiency of the MDE approach compared to the traditional approach is the effort per additional feature.

The specific metrics related to the MDE approach are the metrics on the quality of the model. We use the metrics described in [1] that can be applied to MDSysE models

- Lack of cohesion of methods (LCOM)
- Coupling between objects (CBO)
- Depth of inheritance tree (DIT)

Tools environment

To support those feature, a set of tools are used:

- Doors to identify the requirements
- MDSysE and CCM Tools for the system design
- Reqtify to gather and use the traceability links
- UPM safety tool to help the safety analysis
- IBM tool to study the test automation
- OCL tool to check the modeling rules

The tool chain is described in the figure below:

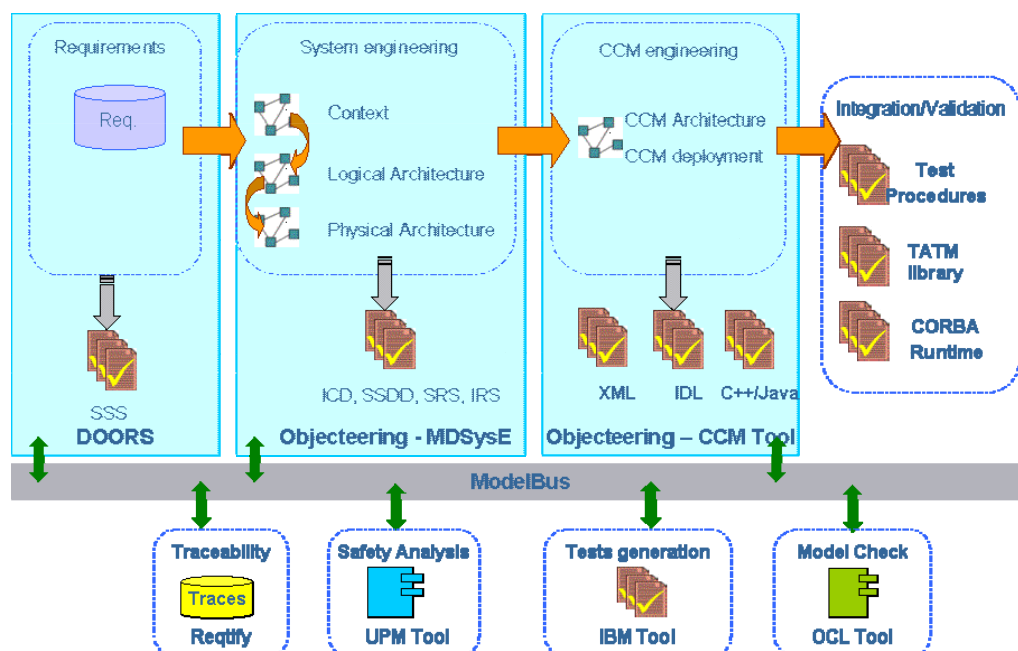


Fig. 4. Tool chain

To guarantee that the system model will have a lifetime appropriate to the product line need, the tools included in the tool chain have to be substitutable in order to be independent of the tools vendor lifetime.

To guarantee that we will be able to increase the productivity by automating our activities, the tools need to be interoperable. The model can be transferred from one tool to another because XML is used. It can be transferred through the Modelbus.

Results of the project

Today the project allowed Thales ATM to solve some important aspects frequently raised as obstacles to adopt a MDE approach:

- An effective and practical instantiation of the MDE process, the needed support and training to this process.
- Selection of a tools chain with necessary adaptations needed to provide an answer to the issues specific to the MDE approach.
- And last but not least the team composition, which allow a smooth and efficient experience acquisition.

Team

The first return of experience from this project in term of team skills needed is:

- End user knowledge (in this case ATC controller knowledge or even practice),
- Experience in equivalent system development (here ATC)
- At least to start the project 10% of people having a real industrial experience in the use of UML
- System architecture experience
- Safety experience
- All the team must be able to use the model including the safety and quality engineers in order to be able to perform their job

In a company that deploys MDE, there is usually no experience in MDE or even UML. There are many ways to get this experience:

- Acquire experts from outside. There may be problems if the different experts do not have the same idea of how to do the job.
- Learn by doing on a small project.

Process and tools

The positive aspects of the use of MDE at this stage of the project are:

- Have a consistent view of the system thanks to the model check
- Benefit from the model transformations by avoiding tedious and error prone copy paste (e.g. from one level of decomposition to another or from the logical model to the physical model)
- Have different views of the system
- Make all the project actors work on the same input and have their own view thanks to the model transformation (e.g. from system engineering functional, data and use case views to the safety engineering Fault Tree Analysis view)

We assume that generating the interface classes of the project components will ensure the consistency of those interfaces. This will be known in 2 years time.

The recommendations that can be made after one year of MDE use are:

- The tool needs to be in an industrial state, which means that:
 - The tool has to be tested in an operational environment within a small team before it is used at a larger scale. This is to avoid what we call operational defects (e.g. ergonomic issues) to be identified by too many people at the same time.
 - The use of the tool has to be clearly defined in a very detailed manner. This use has to be known by the team.
- The tool support has to be estimated according to the tool industrial status. It can be up to 10% of the effort.
- The tool chain has to include tools that are interoperable and substitutable

Thales ATM project production result as today:

The project achievements are today:

- The system architecture and design have been produced,
- The documents are generated automatically,
- The components interfaces are consistent,
- The components interface classes code are generated automatically,
- The safety analysis has started to use the automated help

There is room for improvement in the following parts:

- Test generation from use cases: The CSCIs tests or system qualification tests check that the system behaves as in the model use cases. The tests scenarios could be generated from the use cases. Today, there is still a lot of work to initialize the values of the data and to check the system state in a test.
- Model rule checking: A lot of model checker exist on the market, they are not yet interoperable and do not enable to add easily its own rules.
- Metrics: The metrics need to be graded for the project environment.
- Communication of the design: No solution has been found yet to be able to communicate the model to people who do not know UML (customer, manager, quality manager...).
- Reduction of the system complexity: No solution has been found yet to break up the system complexity into simple problems.

The Return Of Investment expected by Thales ATM when the MDE approach has been retained for this project will be obtained at the end of the system integration only. First results from the project today reinforce Thales ATM in the confidence that this goal will be achieved. The major savings are expecting from the automation of the development tasks and the maintenance costs reduction allowed by the existence of the model.

Thales ATM is expecting also a significant ROI from the reuse of the project production through the product line approach facilitated by MDE.

The choice of the MDE approach is confirmed now for the next steps of the project. During the next activities (Software design, software development, system integration) the implementation of the process will be refined, the development platform put in place and tools used.

It has to be noticed that a validation platform of the whole development chain is already deployed and intensively used for this purpose.

References

- [1] Applying and Interpreting Object Oriented Metrics SATC Dr Linda Rosenberg