From Document-Based to Model-Based System and Software Engineering

Experience report of a selective catalytic reduction system development

Morayo Adedjouma[1], Thibaud Thomas[2], Chokri Mraidha[1], Sebastien Gerard[1], and Guillaume Zeller[2]

[1] CEA, LIST, Department of System and Software Engineering
Gif-sur-Yvette, France
{firstname.lastname}@cea.fr
[2] Controlled Systems department
Plastic Omnium, Auto Inergy Division, Brussels, Belgium
{firstname.lastname}@plasticomnium.com

Abstract. Plastic Omnium (PO) provides plastic fuel and emission reduction fluid systems for car manufacturers. These products, must be customizable to respond to different manufacturers needs while being compliant to ISO26262 and A-SPICE standards. To satisfy these needs, PO employed a development approach where system functional specifications are document-based, and software architecture design conforms to AUTOSAR. However, this approach requires to manually translate between textual specifications and software models, usually rising inconsistency and maintainability issues. This paper reports on the lessons learned with the introduction of a full model-based approach to solve the issues identified in its former approach. The proposed approach is evaluated on a real case study using the open-source Papyrus tool.

Keywords: MBSE, SysML, AUTOSAR, traceability, open-source tool

1 Introduction

Plastic Omnium (PO) is the world leader of automotive blow-molded fuel systems. In 2015, PO has produced 18.8 million of plastic fuel tank systems. Furthermore, PO is one of the main suppliers of Selective Catalytic Reduction (SCR) [7] systems, such systems participate to the reduction of nitrogen oxides in exhaust gases. Since 2006, PO has designed controlled systems composed of integrated sensors, actuators and controllers including hardware and software. Such products include fluids management functions such as filling, storage, venting, gauging, feeding and control. To develop the products, PO faced challenges in terms of specification, architecture definition, integration, and qualification and certification, for which it deployed a Document Based System Engineering (DBSE) approach with several abstraction levels supported by commercial tools.

Recent market trends impose more flexibility in automotive product development. A complete system may be share between several suppliers, each supplier being responsible for a coherent subset of sensors, actuators and controllers. To respond to these new business cases, PO decided to develop its systems by decoupling the hardware controller from the application software. The application software is henceforth developped following the model-based AUTomotive Open System ARchitecture (AUTOSAR) [8] while the other development activities had been still performed using text-based specifications.

An important challenge in adopting AUTOSAR into the development approach is that there is manual modelling to be performed at lower levels from the text-based specifications. Manually translating the specifications into AUTOSAR models is extremely time consuming and error prone with regard to the large set of documents and different stakeholders involved. As a consequence, there is a breakdown in the traceability chain rising, among others, inconsistencies and maintainability issues in the development process regarding an ISO26262 [10] and Automotive SPICE (A-SPICE) [9] compliancy. To overcome these weaknesses, PO is involved since 2013 in moving to a full Model Based System Engineering (MBSE) approach supported by the open-source Papyrus tool [4].

The objective of this paper is to report on the impact of the introduction of the MBSE approach and tools at PO with regard to the challenges faced with the former approach. The MBSE approach aims at developing a coherent and rigorous product end-to-end development process which gives back to traceability a central role. The intent is also to better address the requirements in terms of architecture, qualification and certification for automotive systems as recommended by standards like ISO26262 and A-SPICE. The key components of the approach are modelling languages, a modelling methodology and a modelling tool. PO chose the System Modeling Language (SysML) [12], a largely adopted modelling language in industry, for realizing the system levels activities. The Software activities are compliant to AUTOSAR. The approach was implemented in Papyrus, an open-source tool that supports both SysML and AUTOSAR, and offers facilities to automatically transform and trace the models from one language to another.

The remainder of the paper is organized as follows. In Section 2, we highlight the challenges faced by PO with its former DBSE approach. Section 3 describes the conceptual components of the adopted MBSE approach. In Section 4, we evaluate the major features of the approach on a real case study and we discuss the outcomes observed in Section 5. Section 6 concludes the paper and sketches some future perspectives.

2 Document Based System Engineering Approach

2.1 Overview of the Document Based Approach

To specify and design controlled systems that meet automotive manufacturers' expectations, PO has been following a DBSE approach. The specification and design processes map the left side phases of a V-cycle.

The processes are refined through 5 abstraction levels as shown in Fig. 1: The *Project Requests* level to elicit the systems goals and use cases to satisfy; The *Functional Technical Specification* (FTS) level to define system functions, their performance and safety requirements, and interfaces; the *Architecture Technical Specification* (ATS) level to design system functions with basic architectural blocks; the *Component Technical Specification* (CTS) level PO to specify and assess components perimeter and technical requirements; and the *Components Implementation* level to design and develop the components.

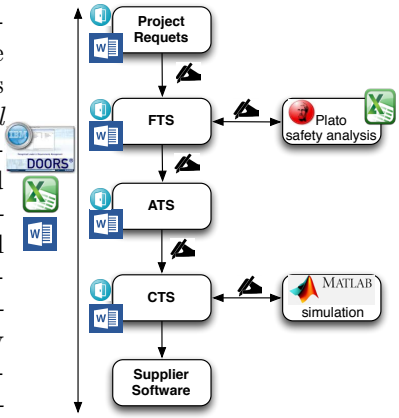

Fig. 1: PO DBSE Approach

The DBSE paradigm provided a working environment at PO based on the different phases of the development process. For each phase, there are several documents of different kinds that are defined for addressing dedicated customers, designers, suppliers, testers needs, etc. The documents structuration and organization was constructed at two dimensions: 1) *vertically* to keep a consistent abstraction level information within the same document, e.g. function versus component requirements, 2) *horizontally* to highlight specific information within one abstraction level, e.g. algorithms description and variables libraries.

In principle, the produced documents are logically related to each other according to the development process. The DBSE approach included capabilities to express traceability relationships between different documents as well as between the concepts depicted within a document. The relationships allow specifying the client and the supplier of the documents, while making use of different semantics (use, coverage, validation, etc.) to link them.

An IBM-based tool chain was advantageously deployed to support the requirement management, traceability, releases, review process, publication and to limit as much as possible manual operations to handle the methodology.

2.2 Limitations of the Document Based Approach

When PO was refining its development process to address the new business opportunity constraints, the DBSE approach raised several issues related to project planning and timelines, design information specifications, traceability, tool support, collaborative work and standards compliance.

(1) Project workflow and planning. In a DBSE approach, the document is the main work artefact and the container for requirements and design information. The project team is focused on the document process which consists in writing, reviewing, updating and publication sequential steps. The document contents are frozen in order to be released according to the project timelines. The latter are aligned with business objectives rather than requirements and design

issues. There is a huge workload to apply rigorously the sequential document process. To reduce delays and publish specifications on time, high level and low level activities are done in parallel. However, this requires an additional effort to maintain coherence and consistency in project information. Consequently, the documentation is often late defined and may become obsolete immediately after their publication. Since the unique representation of the information are the documents, the specification updates are not available before the next release.

(2) Design semantics. In the DBSE approach, the specification documents capture requirements and design information in a textual format. Some design aspects are depicted with charts, for illustration purpose of a textual statement only. The drawback is that Natural Language is not suitable to describe some design concepts, e.g. system architecture, and technical choices may wrongly be expressed as requirements. The consistency between text and drawings information is also difficult to establish: over specification may appear and then create extra work in the project, e.g. too complex design, additional validation, etc.

(3) Traceability. The DBSE approach focuses on dealing with textual statements. Traceability between high and low levels design information is very limited since the drawing elements are not traceable. Therefore, design allocation across abstraction levels is difficult, which undermines the coherence between the system level and AUTOSAR software architecture. Traceability is mainly achieved for documents releases. It is an inconvenient when a change is requested between releases, because impacts are not fully supported by traceability. It is then difficult to estimate time and costs to fulfill change requests.

(4) Communication. Team communication is supported by document publications, which are synchronized with document releases. Additional drawings, presentations are built on demand to support external communication. However, particular aspects of the system are hard to understand and communicate since the information is spread in several documents or not properly highlighted by the document format.

(5) Tooling. The DBSE approach relies on a set of commercial tools which let appear some limitations for PO needs. The requirement management tool DOORS [2] is used to handle all the documents (word, excel) produced in the DBSE approach (Fig. 1). They are stored in a tree structure, in which one module represents one document. Each module contains objects such as headings, requirements and drawings. Properties associated to each object are captured following a simple metamodel. However, these properties are managed directly at the document level and may not comply with the metamodel.

Drawings are edited in another environment and then copied in the documents. Publications are done through a publishing engine which allows generating appropriate documentation based on module content. The safety analyses are performed at FTS level using the commercial tool Plato [5] and Excel files while Matlab [3] is used for design simulation purpose (see Fig. 1). The integration between the different tools is not complete and imposes manual exchange operations. For instance, synchronization between requirement management, de-

sign and failure analysis is particularly time consuming and error prone. Tool deployment also requires huge effort to maintain an up-to-date environment.

(6) Standards compliance. As the market context evolves, PO wanted to be more align with standards. PO starts developing the application software of its controlled systems following AUTOSAR and using Matlab. AUTOSAR, as Matlab, are based on rich metamodels that are not easy adaptable to a DBSE approach. This creates a large gap between within the development process, which undermine the overall traceability. The limited traceability and tooling interoperability capabilities become an obstacle to reach the A-SPICE level 2, with regard to requirements traceability. It also a weakness to be aligned with the system development process and safety analyses as ISO26262 recommends.

Regarding all the limitations identified with the DBSE approach, PO decided to shift from its document-centric approach to a model-centric approach.

3 Shifting to A Model Based System Engineering Approach

The INCOSE defines MBSE as "the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phase [13]. Such MBSE approach necessitates one or many modeling languages, a development methodology and a framework that implements the modeling languages, preferably customized to support the development methodology. PO selects the modelling languages, methodology and framework in such a way its MBSE approach enables covering all the engineering phases from the requirement specification to the software development.

Modelling languages. PO considered the following criteria for the choice of a Modeling Language (ML): a) the ML shall provide means to capture requirements, structural and behavioral design, and implementation of architectural elements; b) the ML shall provide a generic support for traceability; c) the ML shall be a standard language, easily learnable by PO engineers to ease communication between stakeholders and to facilitate recruiting resources already trained; d) the ML shall be supported by available and robust tools; e) the ML shall be extensible and customizable to fit into PO process needs.

The most known system-level architecture description languages in the automotive domain are EAST-ADL2 [6], and AADL [14]. EAST-ADL2 is an de-facto standard for automotive system development. EAST-ADL2 provides concepts for requirements specification, system functions specification and design, and traceability support. In addition to the language, a specification and functional design methodology is provided with a link to AUTOSAR. However, its behavioral specification is limited to a subset of state machines that may not be sufficient for covering PO needs. AADL is a language mostly used in avionics and aerospace domain. In comparison to EAST-ADL2, AADL provides software and hardware concepts for lower abstraction levels. Some concepts like process and threads are provided but can be incompatible with AUTOSAR related concepts, while

requirements are not supported by the core language. AADL, as well as EAST-ADL2, have poor tooling support and very limited extension and customization capabilities.

In [15], a survey places the Unified Modeling Language (UML) [11] as the most used, tool supported and disseminated modeling language. UML provides a standard extension mechanism called profile, which allows enriching the language with domain specific concepts. SysML is such an UML profile that specializes UML concepts for system engineering, and that can be further specialized using profiles. SysML fulfils the four criteria of PO for a modeling language. Hence, SysML and AUTOSAR were chosen as basis to build PO MBSE approach.

Abstraction layer	Technical Specification purpose	SysML diagrams*	Key model element
System (SysTS)	Operational Viewpoint identifying system frontiers, system functionalities, main environment constituents, interactions with the environment constituents	UC, BDD, IBD, PKG	Use case Actor
Function (FTS)	Functional Viewpoint refining SysTS by identifying System Functions (inputs/outputs, behaviors, requirements, performance criteria, fault tolerance)	REQ, BDD, IBD, ACT, SEQ, STM, PAR, PKG	System Function (SysML Block) Requirement
Architecture (ATS)	Organic Architecture Viewpoint refining FTS by identifying Basic Blocks (sensing/actuating, processing) and how they realize System Functions	REQ, BDD, IBD, PKG	System Function Design (SysML Block) Design Block (SysML Block)
Component (CTS)	Component-oriented Viewpoint refining ATS by assembling Basic Blocks according to an abstract target platform made of HW/SW modules	REQ, BDD, IBD, ACT, SEQ, STM, PAR, PKG	HW and SW modules (SysML block) Behavior (SysML Activity)
Software (SwCTS)	Building a SW architecture aligned with AUTOSAR	BDD, IBD, ACT, SEQ, STM, PKG	AUTOSAR SW component Internal Behavior (AUTOSAR runnables flow)

* UC: Use Case, BDD: Block Definition Diagram, IBD: Internal Block Diagram, PKG: Package Diagram, REQ: Requirement Diagram, ACT: Activity Diagram, SEQ: Sequence Diagram, STM: State Machine Diagram, PAR: Parametric Diagram

Fig. 2: Components of the MBSE Approach

Modelling methodology. PO defines its modelling methodology based on 5 abstraction levels. The specifications are realized in the System Technical Specification (SysTS) and Functional Technical Specification (FTS) models. The system design is achieved in the Architecture Technical Specification (ATS) and Component Technical Specification (CTS) models. The AUTOSAR implementation is achieved in the Software Component Technical Specification (SwCTS) model. Table 2 presents the main goals of the different models with their associated diagrams and key model elements. Traceability relationships like ≪realize≫, ≪satisfy≫, ≪refine≫, ≪assembly≫, etc. are used to trace the above mentionned models. Fig. 4 shows a traceability crosscutting view from SysTS to SwCTS level.

Modelling framework. PO chooses Papyrus [4] to deploy its MBSE approach. Papyrus is an open source tool that provides a generic SysML editor and support for AUTOSAR software component template modelling and export in ARXML standard [1]. Papyrus provides advanced customization capabilities based on profiles and specific tooling artefacts including model explorer, versioning, source control, model concurrent access, code and documentation generation, libraries, user type management and change request. The benefits of the platform consist in its flexible integration with others tools. Hence, PO creates a gateway between

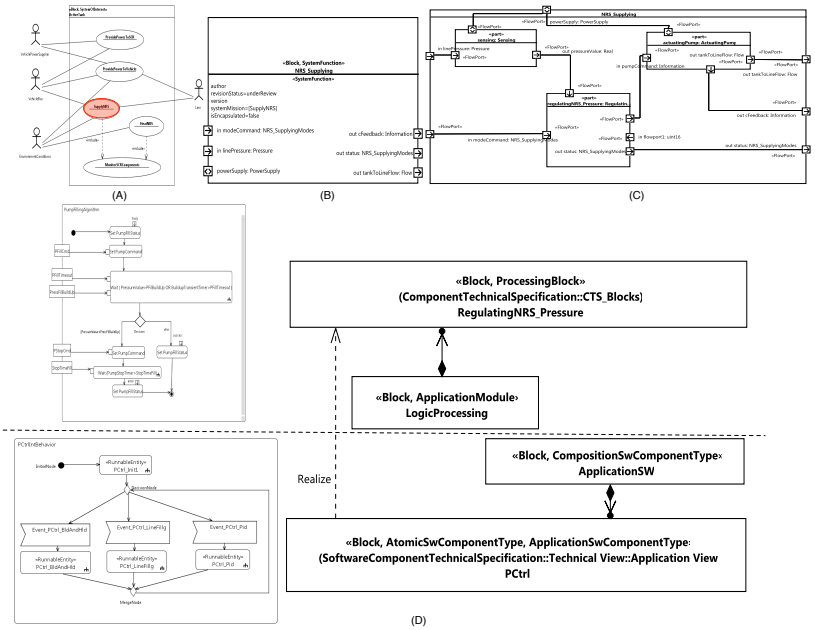

Fig. 3: Example of diagrams at SysTS, FTS, ATS, CTS, SwCTS levels

DOORS and Papyrus. It was simpler to integrate Matlab with the framework as they are both model-based environments.

4 Case Study

To initiate the transition from DBSE to MBSE, different training sessions were organized in order to convey a unified meaning of MBSE to PO team. A representative excerpt of the Selective Catalytic Reduction (SCR) [7] system of PO, previously developed with the DBSE approach, was re-engineered to evaluate the MBSE approach.

Case study description. The SCR main function is to provide a flow of NOx Reduction Solution (NRS) fluid to the exhaust line and the injector from the tank. It relies on environment features (temperature, atmospheric pressure, etc.) of the tank as well as car information (modes, wake up) to provide heating and power energy to them. It also monitors NRS tank temperature, quantity, quality and availability of its actuators. SCR uses communication bus to receive car information and send feedback to the vehicle.

Case study modeling. The SCR missions were modeled at the SysTS level with a Use Case diagram (Fig. 3(A)). Each use case is represented by a SF in the system description at FTS level (Fig. 3(B)), further refined with Sequence Diagram. The SF include a State Machine Diagram that describes its behavior. At ATS level, each SF is decomposed into sensor, actuator and processing blocks in an IBD (Fig. 3(C)). At the CTS level, the blocks from ATS level

are combined into mechanical modules (interface with the physical world/laws), hardware modules (sensors/actuators), application modules (pure SW modules) or driver modules (SW modules that need specific interface with the hardware).

These modules are used to build the abstract HW/SW platform. A Parametric Diagram refined them to implement the physical laws triggered by actuator commands. At SwCTS level, application and driver modules from the CTS level are realized with AUTOSAR composite or atomic SW components concepts (Fig. 3(D)).

Traceability is achieved between the abstraction levels (Fig. 4). At the FTS level, System Functions *refines* Use Cases from SysTs level. At ATS level, the System Function Design *realizes* the System Functions. At CTS level, Design Blocks from ATS level are *assembled* into modules to build the HW/SW platform. Here, the algorithms may be embedded as Activity Diagram into processing block. At SwCTS level, AUTOSAR SW Components *realizes* the CTS SW modules and the algorithms are mapped to the runnable entities of these components.

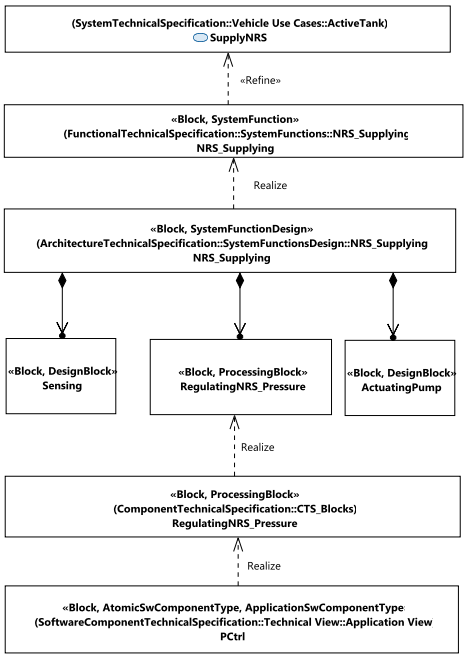

Fig. 4: Traceability crosscutting view from SysTS level to SwCTS level

5 Discussion

In this section, we highlight the benefits and limitations brought by the MBSE approach with regard to the DBSE approach ones (see Section 2).

(1) Project workflow and planning. Documents have been advantageously replaced by a global model as a specification and design environment. Project planning is notably less constrained as by a sequential workflow of documents writing. As a result, it is possible for engineers to work simultaneously on distinct levels. Design is henceforth elaborated earlier and evaluated iteratively. Traceability ensures consistency in this process between higher model elements and their lower level realization. One unanticipated finding was publication effort to convert models into documents according to quality process.

(2) Design semantics. SysML models and associated diagrams have replaced structural and behavioral design descriptions in natural language and illustrations. The results of this study have highlighted how over specification may have existed in previous project due to inappropriate use of textual requirement to

depict design aspects. Balance between requirements and design activities is recovered with formal, consistent, traceable and correlated model elements. SysML also tackles complexity and foster reuse via the reusing blocks as parts in IBD.

(3) Traceability. The MBSE approach helps to automate traceability which facilitates justifications and global consistency with regard to certification issues. The SysML traceability relationship types support all the former DBSE traceability with important improvements because they are integrated in the model itself. The trace model enables having continuity through the different abstraction levels, from system representation to AUTOSAR runnables. The traceability can be exploited for automated checks, e.g. validation rules, change impact analysis or requirements coverage.

(4) Communication. The MBSE approach facilitates communication between the stakeholders concerned by the system under consideration. First, it allows specifying requirements, architectural design, and behavioral aspects in a unique and semi-formal formalism: SysML. Second, dedicated viewpoints or additional diagrams can be constructed from the same model if needed either by the methodology, or for design and communication purpose. Documentation outcomes are still generated according to stakeholders needs (customer, designer, suppliers, testers, etc.), but the publication is now realized on demand by extracting model elements and it is not only linked to project publication milestones. Additional communication challenges have nevertheless risen between engineers dealing with technical aspects in SysML and documents reviewers or approvers. A special effort has to be done to simplify SysML in a viewer tool and to deploy PO Domain Specific Language (DSL). To reduce this skill issue and associated business risks, training are available for employees and SysML is required for new hiring.

(5) Tooling. Papyrus tool has been deployed to support the MBSE approach. PO exploits the customization features of the tool to build its own environment. To ease the transition from DBSE to MBSE, PO implemented a gateway between DOORS and SysML. It has allowed importing and replacing the existing textual requirements with new model elements. PO uses the native papyrus SysML and AUTOSAR editors to support the definition of the different models required by its modelling methodology. To address further concerns, PO uses several profiles on top of their system modelling. A dedicated profile enables to automatically perform dysfunctional analyses compliant to ISO26262 recommended practices. Another plug-in enables to import and export modules from SysML to Matlab while keeping them synchronized. Papyrus also allows document generation in order to support document releases. Furthermore, with Papyrus, it is easy to deploy a dedicated configuration for a small skilled team. A tool chain responsible role and associated processes have nevertheless to be created to support the deployment in a large scale international team.

(6) Standards Compliance. The MBSE approach allows a smooth integration between specification, design and implementation levels. With this approach, there is no longer a gap between high-level system specification and its AUTOSAR implementation. The automated traceability mechanisms, done contin-

uously as part of the definition and decomposition process, helps meet A-SPICE expectations and respect the top-down approach fostered by ISO26262.

6 Conclusion and Future Work

This paper presents the MBSE approach for the specification and design of controlled systems in deployment at PO. The approach is built on 3 main components, namely the SysML and AUTOSAR modelling languages, a methodology defined over 5 abstraction levels and the Papyrus framework that supports the latter. We report on lessons learned from the application of the approach on an automotive system. The experience highlights how the MBSE approach responds to identified issues in the former PO DBSE approach thanks to an enriched design capability and to an end-to-end traceability in development process. Thanks to Papyrus, it leverages interest for integration of further concerns, e.g. dysfunctional analysis, design simulation, code generation. The MBSE approach also presents promising assets to facilitate process and product certification.

References

1. Autosar Specifications. `http://www.autosar.org/specifications/release-42/`.
2. IBM Rational DOORS. `http://www-03.ibm.com/software/products/`.
3. Matlab. `http://fr.mathworks.com/products/matlab/`.
4. Papyrus Modeling Environment. `http://www.eclipse.org/papyrus/`.
5. Plato AG. `/http://www.plato.de/`.
6. EAST-ADL Association. EAST-ADL Domain Model Specification V2.1.11. Technical report, 2013.
7. J. de Beeck, J. Thompson, and N. Booth. Upcoming emission regulations for passenger cars: Impact on SCR system requirements and developments. Technical report, 2013.
8. B. Dorneanu and J. Zimmermann. Enabling software business with AUTOSAR. *ATZextra worldwide*, pages 44–47, 2013.
9. International Organization for Standardization. Information technology – Process assessment. Technical report, ISO, Geneva, CH, 2012.
10. International Organization for Standardization. Road vehicles – Functional safety. Technical report, ISO, Geneva, CH, 2012.
11. Object Management Group. Unified Modeling Language, v2.4.1, formal/2011-08-06. Technical report, OMG, 2011.
12. Object Management Group. Systems Modeling Language (SysML), V1.3, formal/2012-06-01. Technical report, OMG, 2012.
13. Incose. *Incose Systems Engineering Handbook V4*. John Wiley and Sons, 2015.
14. SAE Int. Architecture Analysis & Design Language. Technical report, SAE, 2004.
15. I. Malavolta, P. Lago, H. Muccini, P. Pelliccione, and A. Tang. What industry needs from architectural languages: A survey. *TSE Journal*, pages 869–891, 2013.