Adoption of SysML by a Railway Signaling Manufacturer

Alessio Ferrari*, Alessandro Fantechi*, Stefania Gnesi†, Gianluca Magnani‡ and Alessandro Felleca‡
*DSI - University of Florence, Florence, Italy
Email: alessio.ferrari@unifi.it, fantechi@dsi.unifi.it
†ISTI - CNR, Pisa, Italy. Email: stefania.gnesi@isti.cnr.it
‡GE Transportation Systems, Florence, Italy
Email: gianluca.magnani@ge.com, alessandro.felleca@ge.com

***Abstract*—This paper reports the experience of a railway signaling manufacturer in introducing the SysML notation within its development process by means of the TOPCASED tool. Though the tool was later substituted by MagicDraw, the experience was useful to understand the potentials of the notation for requirements formalization and analysis, together with the advantages and drawbacks of using an open-source tool in an industrial setting.**

***Keywords*-SysML; TOPCASED; industrial case-study;**

I. Introduction

General Electric Transportation Systems (GETS) is a company that produces safety-critical railway signaling applications. Given the nature of its products, in 2001 GETS started a collaboration with the University of Florence to improve its development process with the aid of formal methods. Along this path, the company has recently introduced formal modeling and code generation by means of the Simulink/Stateflow[1] platform [2], and has defined a model-based process compliant with the CENELEC standards [1], the set of norms and methods to be used while implementing a railway product for the European market. Simulink/Stateflow are powerful languages for formalizing low-level requirements, while they are less suitable for high-level system requirements specification and analysis. These activities were normally performed by GETS using a paper-based approach, with natural language documents completed by informal diagrams. Natural language is inherently ambiguous and a more formal mean to express requirements was desirable. The OMG SysML[2] language was seen as the solution to substitute the traditional text-centric specifications with a formal notation. The open-source tool TOPCASED was chosen to perform the first experimentation with SysML in a real project.

II. The TOPCASED Experience

TOPCASED version 3.0.1[3] was introduced for the requirements specification of the Failsafe Data Transmission (FDT) system, a platform that manages the switching of the traffic direction between adjacent stations. The project was small enough to introduce a new technology, and the tool was perceived as the right candidate to practice SysML, given TOPCASED's claimed orientation to safety-critical systems development. As an example of such orientation, the tool provides a *model validation feature*, that allows the internal consistency of the produced model and its compliance to the SysML standard to be checked.

A subset of the SysML diagrams was chosen which was considered sufficient to specify the system with a proper degree of detail. This subset was composed by use case, requirement, sequence and structure diagrams (i.e., package, block definition and internal block). The approach planned for the structuring of the different diagrams was aimed at following the CENELEC V-process phases, in order to give a graphical evidence of the adherence to this standard. For this purpose, a single model structured into packages was defined: each package corresponds to a CENELEC phase and includes the diagrams to fulfill the norm prescriptions for that phase. For example, the requirements phase includes mainly use case and requirement diagrams, while the architecture phase is essentially documented with block definition and internal block diagrams. The model was built incrementally, and each artifact of each phase was traced to the elements coming from the previous one.

The SysML language appeared rather intuitive to users with a UML background, and the tool was easy to learn for people with confidence with the Eclipse platform. In general, electronic/telecommunications engineers encountered more hurdles than software engineers, since some basic principles of the model-view-controller pattern are required for a proficient usage of the technologies. These problems were increased by the absence of a proper documentation for the tool. Despite the large literature on SysML, there was no complete tutorial to guide people that were new to both the tool and the language. Furthermore, the notation of internal block diagrams supported by the tool was not compliant to the one presented by the text chosen as a reference [3], and this caused a limited use of these diagrams.

Another issue was the stability of the tool. While the model was growing in size, the tool became slower and more prone to crashes, especially with the increasing number

[1] http://www.mathworks.com/products/simulink/
[2] http://www.omg.org/spec/SysML/1.2/
[3] http://www.topcased.org/

of traceability links between different diagrams. Though this drawback could be associated to the usage of a single model to formalize the whole process, this situation was felt as really frustrating, and led the team to mistrust the tool. As a consequence, many advanced features, such as the collaborative usage, were not experimented. The initial plan was allowing the independent update of the model by different actors in different process phases, but ultimately it was the project leader that took care of the integration of the whole model, according to the input of the other participants.

The final step has been the generation of the documentation. HTML was the preferred format, since with a plain document one would have lost the traceability among artifacts, instead preserved by the hyperlinks. Nevertheless, this choice was criticized by the validation team, on the basis that the format would not have been accepted by the assessors[4]: with a structured document one has a guided direction of reading and understanding, while with HTML one has to choose the navigation path, with the consequent problems of overall uptake.

Since the plain document generation capability of the tool was found insufficient, the team had to re-write the documentation by hand, including the SysML diagrams as figures. At this point there were two document sets to maintain, with imaginable versioning problems, and since the SysML models already had their role for the development of the platform, it was decided to keep the textual documentation as the main reference for further changes.

Despite the goal of a complete renewal of the specification and documentation approach was not achieved, the experience did not result in a total failure. The SysML requirement diagrams, used for structuring natural language requirements, have a poor semantics with few connectors (it is not even possible to define requirements with boolean logic relations), and do not give too much added value in themselves with respect to structured paper requirements.

However, the possibility to clarify these requirements with other formal diagrams, and to perform mutual tracing, gave a consistent support in requirements disambiguation and early discovery of underspecification. The participants agreed that the aid of use case and sequence diagrams as a mean for communication between the requirements manager and the developers simplified the understanding of natural language requirements and increased the level of confidence on the intended behaviour of the system during the implementation.

Furthermore, the SysML model worked as a centralized reference for the other activities (e.g., sw/hw development, tests) during the whole project, representing a useful process control tool for the project leader. For these reasons the SysML language survived within the development process of the company, while the TOPCASED tool was soon abandoned in favor of the commercial tool MagicDraw[5].

III. Lesson Learned

At the end of the project the general opinion was that using an open-source tool to perform core activities in a company with time-to-market pressure and certification constraints was not a good option for two main reasons: (1) companies prefer products with a limited but stable number of functionalities, while lively maintained open-source tools such as TOPCASED tend to have several experimental features that are progressively tuned by the community according to the users feedback; (2) companies require a direct interface with the tool providers that takes the responsibility if a problem occurs with the tool usage.

The choice of Magic Draw was driven by these considerations, and the tool actually confirmed the expectations of a more stable, documented and customer-supported platform. Nevertheless, the initial goal of passing from text-centric specifications, with diagrams clarifying the text, to diagram-centric ones, with notes accompanying the models, was missed again.

Today the company is proficiently employing the tool on large projects, intensively exploiting collaborative usage features and with a generally good opinion of the tool maturity level, but still all the *official* specifications required by the CENELEC norms are manually edited natural language documents. Assessors normally enter at the end of the development process to validate compliance with the standards, and require paper-like documents in order to have a complete picture of the activities performed by the company. While this implies a major effort in terms of production and maintenance of the documentation, it turns out that the investment on SysML pays off in terms of increased confidence on the quality of the specifications.

IV. Conclusion

SysML involves several stages and actors of the development process, and its introduction in a company shall be performed through controlled phases, ensuring the acceptance of any internal and external role player. The experience of GETS has shown that SysML is an appropriate specification mean in a safety-critical context, but *informal* natural language is still fundamental to support the evidence that a *formal* process has been followed.

References

[1] CENELEC. *EN 50128, Railway Applications - Software for Railway Control and Protection Systems*, 1997.

[2] A. Ferrari, et al. *The Metrô Rio ATP Case Study*. FMICS 2010, LNCS 6371, 1-16. Berlin, Germany: Springer, 2010.

[3] S. Friedenthal, *et al. A Practical Guide to SysML: The Systems Modeling Language*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2008.

[4] Third-part companies that certify compliance to the CENELEC standard

[5] http://www.magicdraw.com/