Automatic Transformation of SysML Model to Event-B Model for Railway CCS Application

Shubhangi Salunkhe | Abdul Rasheeq | Georg Holtmann

Introduction

- In **EULYNX V & V**[1] approach to prove the **safety-critical requirements, formal methods** are used. Where, the **semi-formal** model was transformed to **UML-B**[3] model manually. Later, UML-B model transformed to **Event-B**[4](Formal Model) model. The process is shown in below Figure 1.

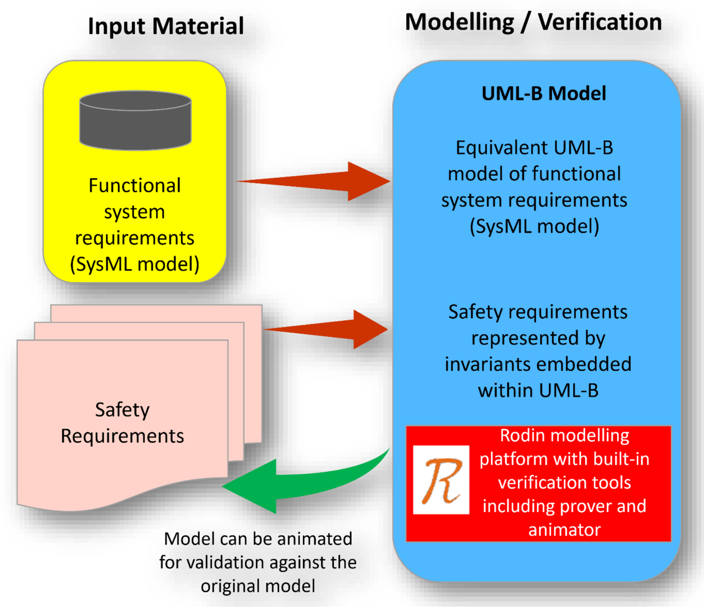

Figure 1. EULYNX V & V approach using Manual Transformation

- EULYNX V & V approach of formally verifying the **SysML**[2] model by manually transforming to a formal model has a few challenges and disadvantages.
- The process of manual transformation is **time consuming**. If the state-machine (system behaviour) is complex, then creating UML-B model takes time and the created model can be erroneous.
- Implementation of transformation rules on metamodels to perform the model-to-model transformation.
- **Automating** the transformation process will significantly improve the V & V process by speeding up of creation of formal model as well as without errors.
- In this poster, we propose a case study of **automatically** transforming the Semi-formal model i.e., **SysML model** to formal model **Event-B** model to support formal verification of EULYNX functional system requirements.

Automatic Translation Process

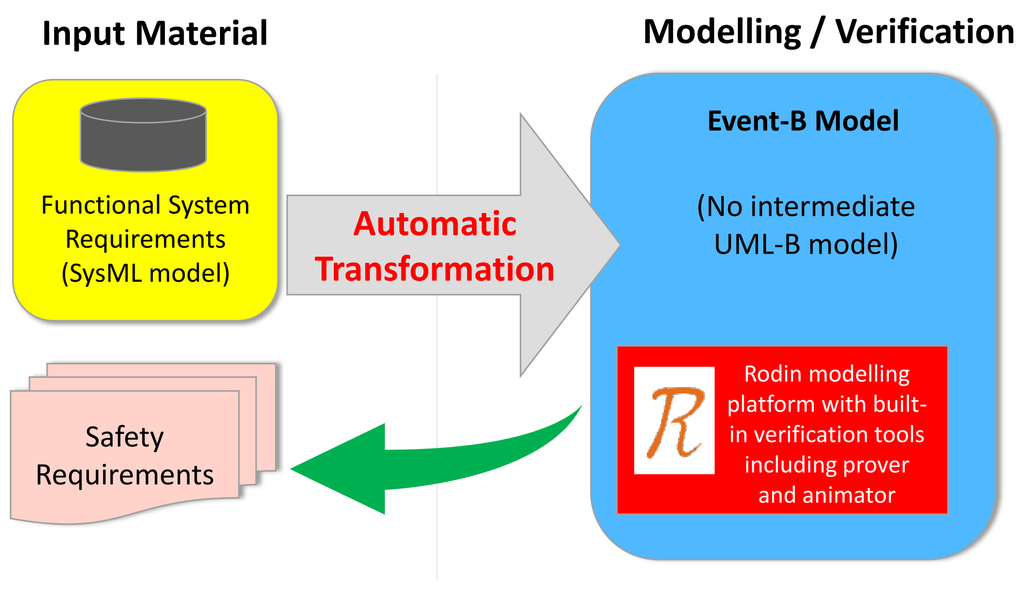

Figure 2. Schematic Block Diagram of Improvised Modelling Approach by Automation

- The Figure 2. represents the automatic translation process, where the intermediate representation i.e., the UML-B model is removed from the process.
- This approach will transform the SysML model directly to the Event-B as the target Model.

Tool Chain

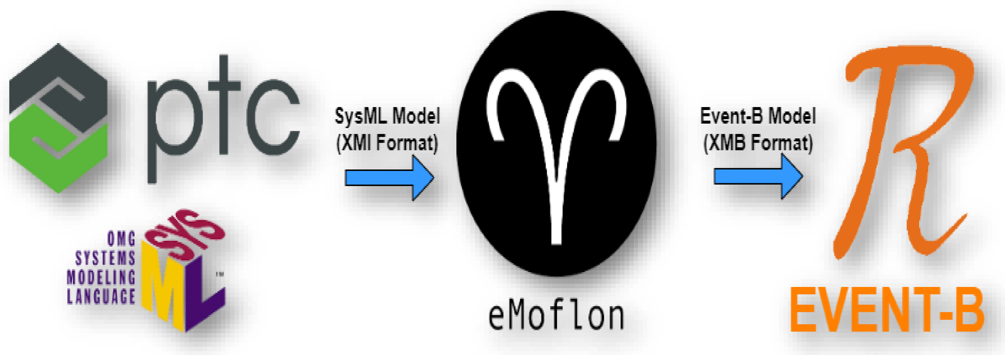

Figure 3. Software Architecture for Automatic Transformation

- The proposed **methodology and toolchain** enables the automatic transformation of SysML specification models into formal models (Event-B) and the generated Event-B Models can be verified.
- It maintains traceability between informal requirements and the modeled system, specifically for the safety properties.
- It reduces the efforts needed for the manual transformation of a SysML semi-formal model to a formal model.
- It supports a modular railway signalling architecture with standardized interfaces.

Methodology

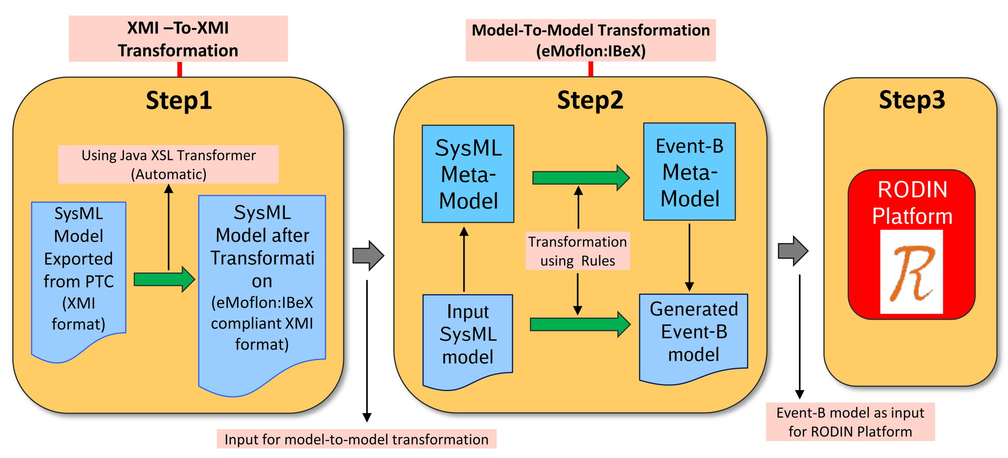

Figure 4. Schematic Block Diagram of Transformation Methodology

- The methodology behind automatic transformation is visualized in Figure 4.
- The input for our Transformation engine is XMI formatted SysML model pre-processed in **Step 1.**
- Then the pre-processed model is transformed to target Event-B model in **Step 2.**
- Finally, the transformed formal model is imported in Event-B for verification in **Step 3.**
- Our methodology uses **model-to-model transformation** techniques, which uses **Rule** based transformation approach.
- The tool **eMOFLON:IBeX** [7] supports model-to-model transformation using **Triple Graph Grammar(TGG)** [7] Rules, the rules are **bi-directional** [7] in nature.
- TGG is a well-known approach to consistency management with the unique advantage of being declarative enough to address multiple consistency management operations with the same specification.

"When applying Model-Driven Engineering(MDE), metamodels with similar objectives are then inescapably created. A recurrent issue is thus to turn compatible models conforming to similar metamodels, for example to use them in the same tool." *[8]*

Semantic Mapping and Rules

- For the transformation process, it is important to understand the **semantics** of both languages and map those semantics to generate the target model.
- In this approach, we identified "seven" constructs that are semantically similar between SysML and Event-B for the case study below.

SysML Concept	Event-B Concept
State Machine	Machine(Project)
States	Variables
Guards	Guards
Transitions	Events
Effects	Actions
States	Default Invariants
Ports	Variables

Table 1. List of Semantic Mapping

- Figure 5 illustrates one of the rules we developed for the transformation.

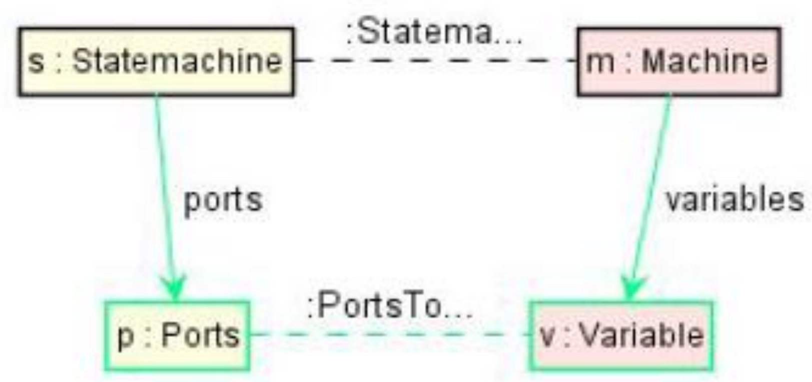

Figure 5. Rule to Transform Ports to Variable

Case Study

- Case study considers **Point Machine** behaviour with only one state machine. The state machine in **XMI format** is provided as an input to generate **Event-B** model. Figure 6 illustrates the case study.

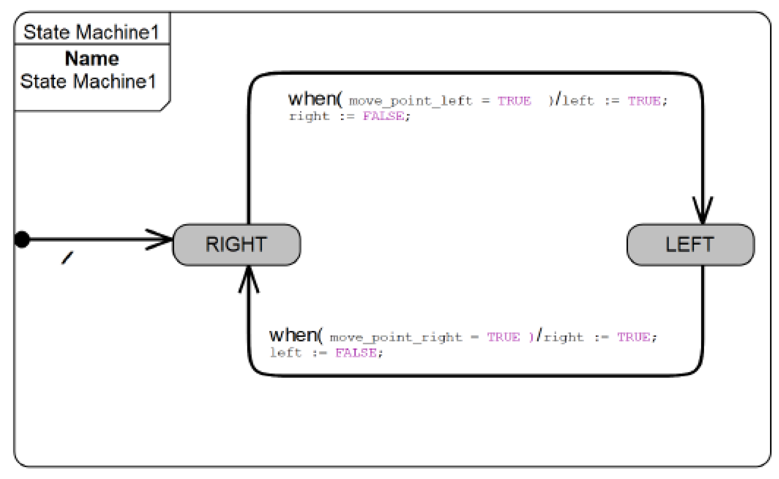

Figure 6. Case Study (Point Machine Behavior)

- After the application of "**transformation rules**" using transformation engine, we successfully generated the Event-B model illustrated in Figure 7. below.

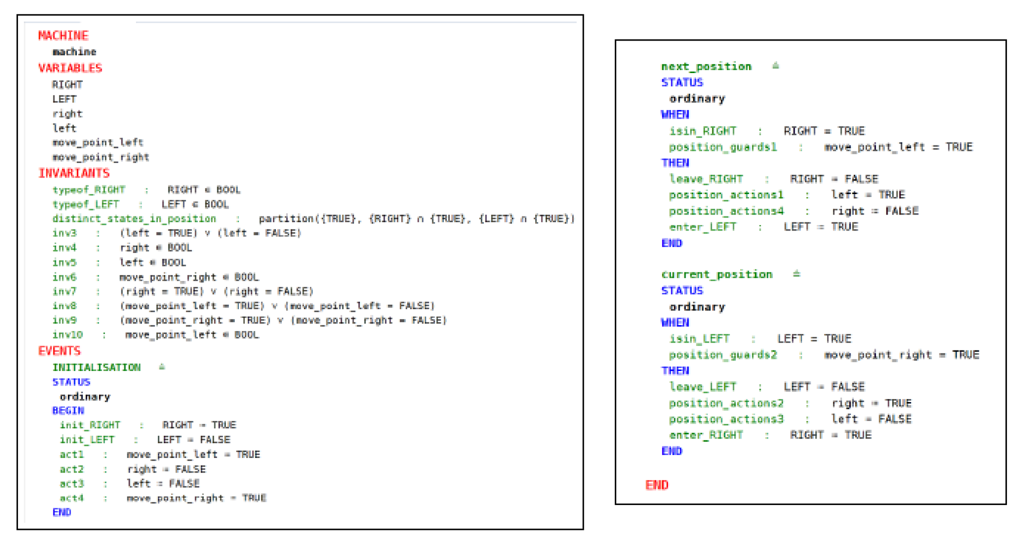

Figure 7. Event-B model

Contact

neovendi GmbH
projects I engineering I digital transformation
Website: www.neovendi.com
Tel: +49 2824 99891 75
Mail: info@neovendi.com

DRSS Digital Rail Summer School

References

1. EULYNX . https://eulynx.eu/
2. SysML . https://sysml.org/
3. Colin Snook. iUML-B statemachines. In Proceedings of the Rodin Workshop 2014, pages 2930, Toulouse, France, 2014. http://eprints.soton.ac.uk/365301/
4. Event-B. http://www.event-b.org/
5. Jean-Raymond Abrial, Michael Butler, Stefan Hallerstede, Thai Son Hoang, Farhad Mehta, and Laurent Voisin. Rodin: An open toolset for modelling and reasoning in Event-B. Software Tools for Technology Transfer, 12(6):447466, November 2010.
6. Salunkhe, Shubhangi, Randolf Berglehner, and Abdul Rasheeq. "Automatic Transformation of SysML Model to Event-B Model for Railway CCS Application." *International Conference on Rigorous State-Based Methods*. Springer, Cham, 2021.
7. Nils Weidmann, Anthony Anjorin, Lars Fritsche, Gergely Varró, Andy Schürr, and Erhan Leblebici. Incremental bidirectional model transformation with emoflon::Ibex. in 2019 (ceur workshop proceedings), james cheney and hsiang-shang ko (eds.), vol. 2355. ceur-ws. org, 45–55, 2019.
8. Falleri, JR., Huchard, M., Lafourcade, M., Nebut, C. (2008). Metamodel Matching for Automatic Model Transformation Generation . In: Czarnecki, K., Ober, I., Bruel, JM., Uhl, A., Völter, M. (eds) Model Driven Engineering Languages and Systems. MODELS 2008. Lecture Notes in Computer Science, vol 5301. Springer, Berlin, Heidelberg.