An Approach to Improve SysML Railway Specification Using UML-B And EVENT-B

Abdul Rasheeq; Randolf Berglehner; Georg Holtmann

Introduction

- **European digitalisation initiatives** in the Command Control and Signaling (CCS) domain such as **EULYNX** [1] aim at a **reference CCS system architecture**[8] in which the system elements are equipped with **standardised** interfaces.
- This new approach requires the creation of **understandable high-quality specifications** and sophisticated methods to **verify** and **validate** them.
- To meet these challenges, an **MBSE Specification framework** (MBSE SF) that facilitates a **holistic model-based seamless** description of complex **CCS** systems is under development. It uses the popular **Systems Modelling Language** (SysML) [2].
- The **EULYNX MBSE approach** has already led to **significant improvements** in the quality of created specifications although it does not allow yet the **formal verification** of system properties.
- In this poster, we present a case study of the integration of **formal methods** into the EULYNX MBSE approach using **UML-B** [3] and **Event-B** [4] as one of the formal methods currently evaluated.

EULYNX MBSE Approach

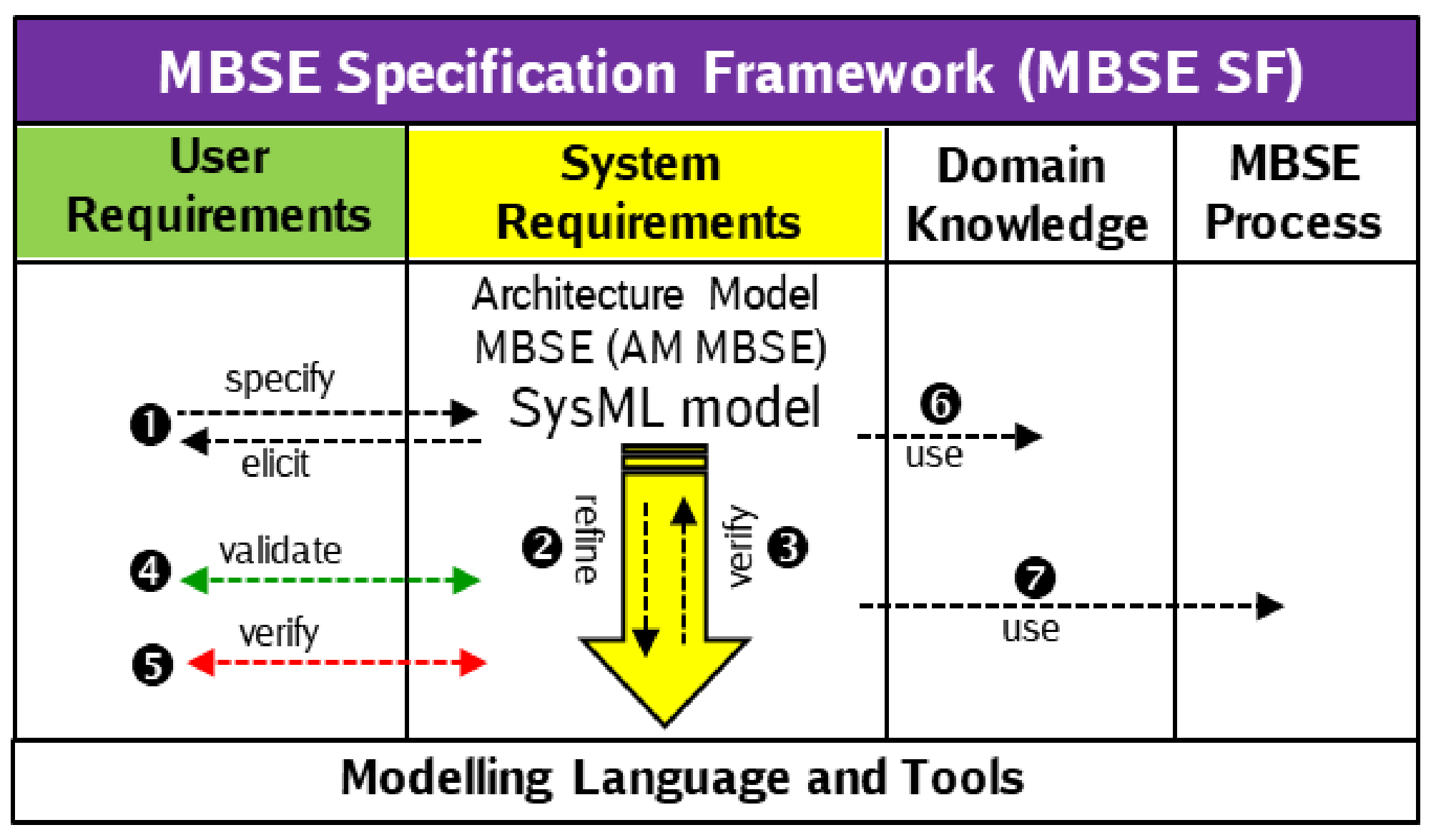

Figure 1. EULYNX MBSE Specification Framework

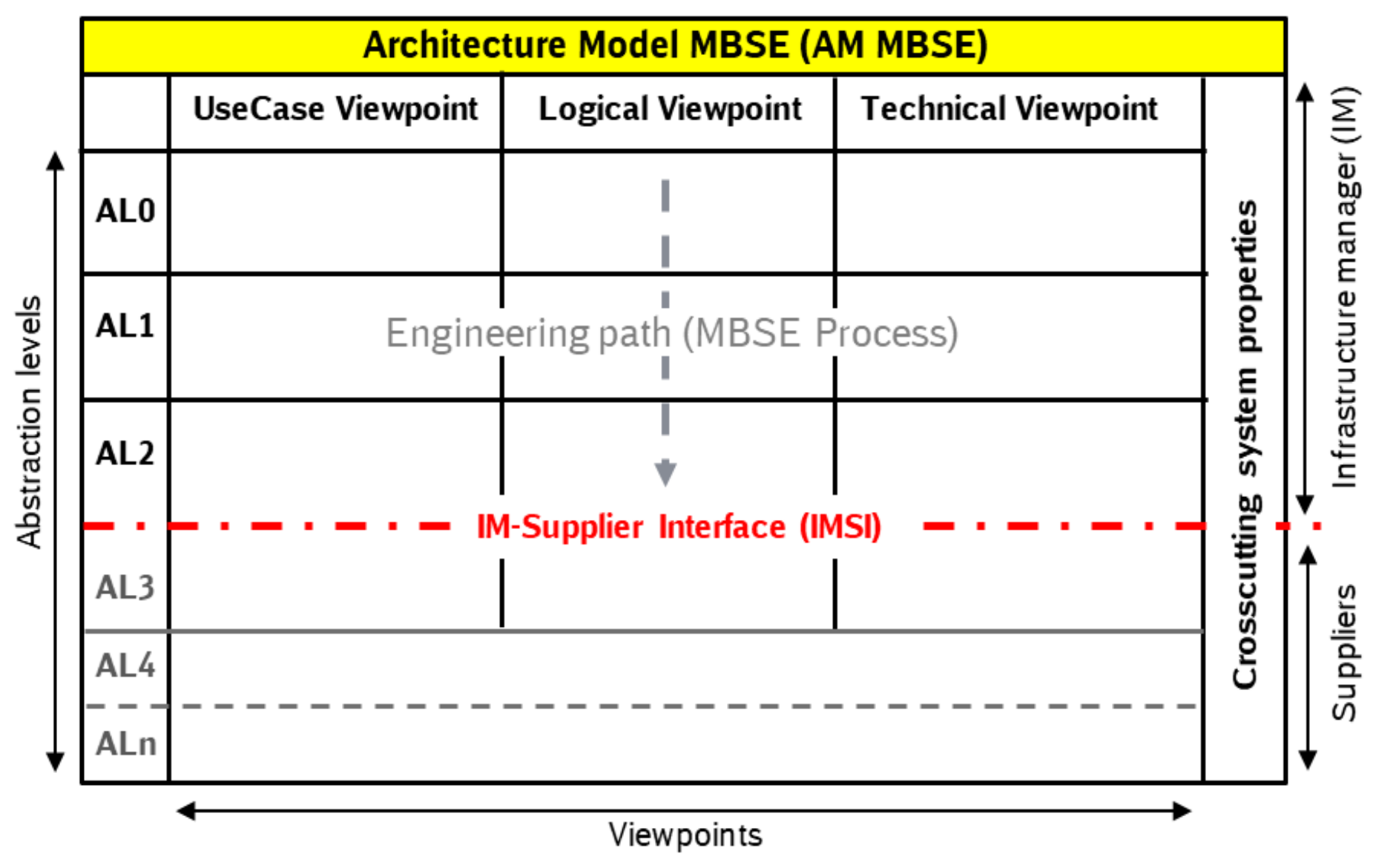
Architecture Model MBSE (AM MBSE)
UseCase Viewpoint
Logical Viewpoint
Technical Viewpoint
AL0
AL1
AL2
AL3
AL4
ALn
Engineering path (MBSE Process)
IM-Supplier Interface (IMSI)
Abstraction levels
Crosscutting system properties
Infrastructure manager (IM)
Suppliers
Viewpoints

Figure 2. EULYNX Architecture Model MBSE

- **Architecture Model Model-based system engineering** (AM MBSE) enables the seamless top-down description of the abstract solution of a CCS system. It defines different **abstraction levels** (AL), **viewpoints** and **views**.
- The functional system requirements are defined using **executable SysML state machines**.
- The **transitions** of the state machines represent the **mandatory functional system requirements**.

Simulation-based V&V

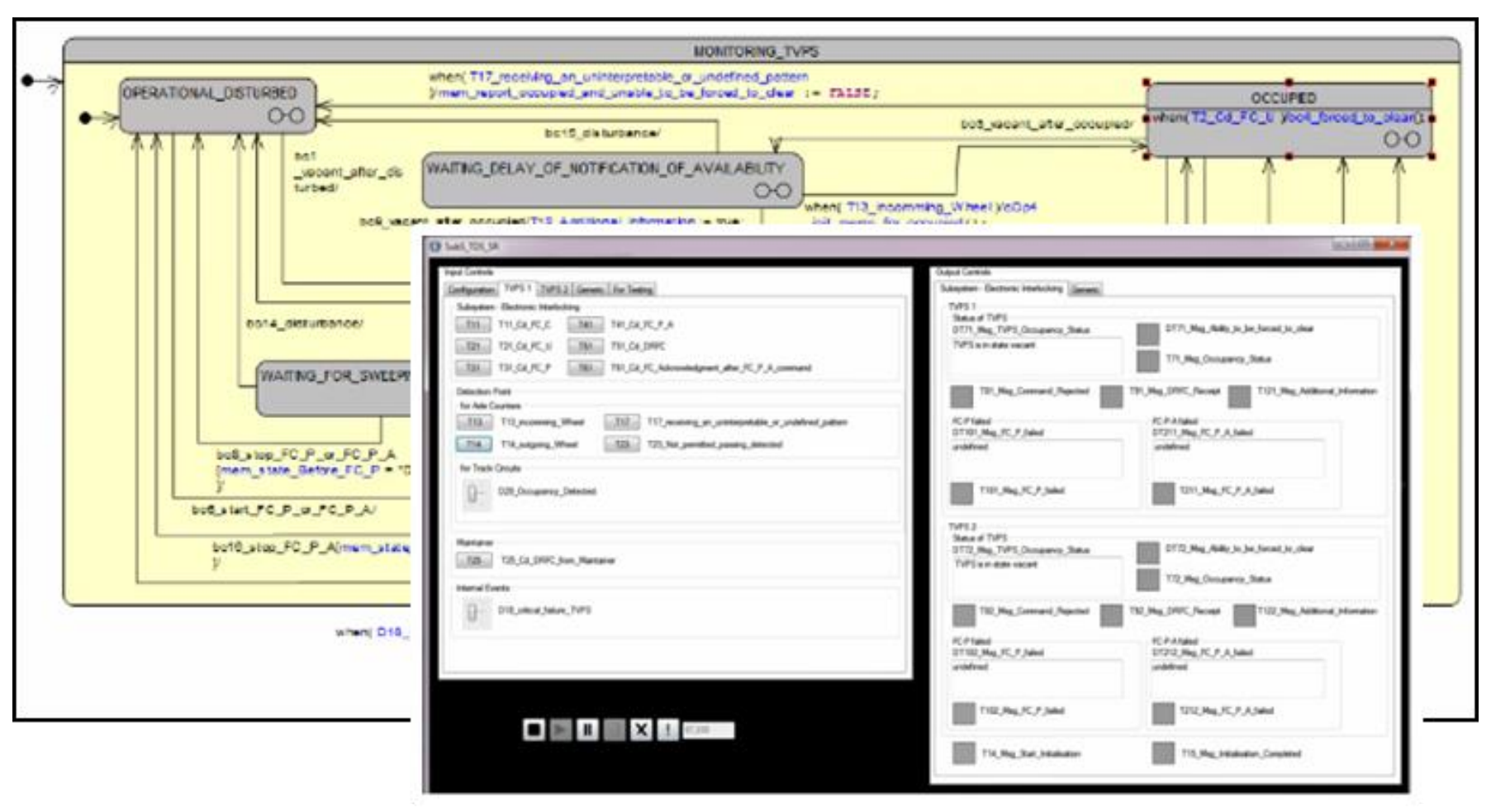

Figure 3. Simulation-based testing of a virtual prototype.

In the current EULYNX approach, the **validation** and **verification** (V&V) of functional system requirements based on user requirements are performed using **simulation-based testing** of a **virtual prototype** (executable state machines).

Formal Methods

As with simulation, it is difficult to prove that the specifications meet **safety-critical requirements.** The EULYNX MBSE approach shall be improved using **formal methods**. The idea is visualised in the process illustrated in Figure 4.

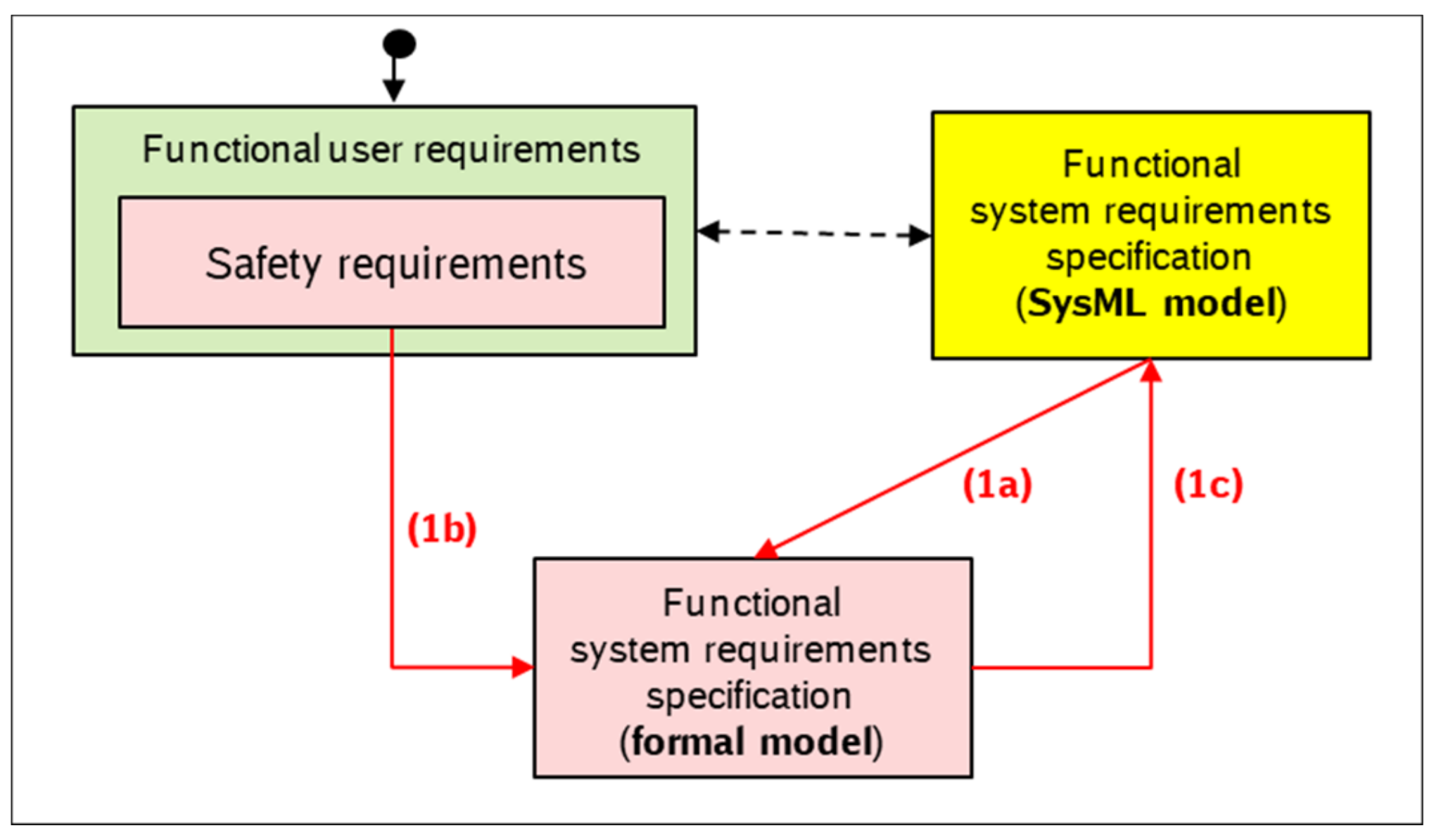
Functional user requirements
Safety requirements
Functional system requirements specification (SysML model)
(1a)
(1b)
(1c)
Functional system requirements specification (formal model)

Figure 4. Illustration of the principle using formal methods.

(1a) Transformation of the SysML model into a formal model based on defined transformation rules and **verification of the transformation**.

(1b) Formal verification of the formal model based on safety requirements (a subset of functional user requirements).

(1c) Correction of the SysML model as appropriate.

The process starts again with **(1a)** until no errors are found anymore.

UML-B / Event-B

- The integration of **formal methods** into the EULYNX MBSE approach is demonstrated using **UML-B** and **Event-B**.
- **UML-B** is a UML-like **graphical front-end** for **Event-B** that provides support for object-oriented and state-machine modelling concepts, which are not supported in Event-B.
- **Event-B** was developed as an **alternative to classical B** in order to support modelling at a **systems level**.

"Railway signalling has been considered as one of the most fruitful areas of intervention by formal methods." *[7]*

Translation to UML-B

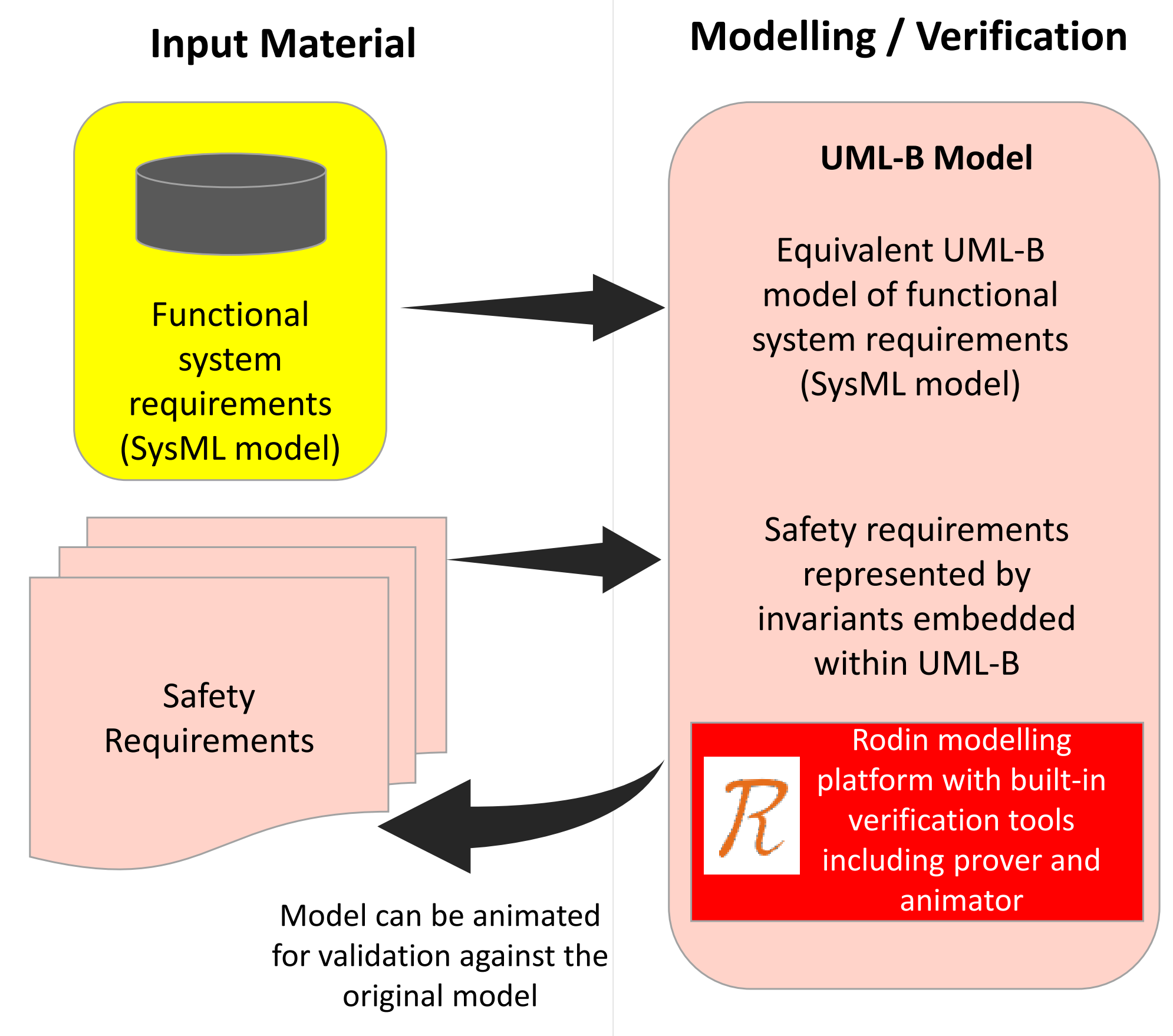
Input Material
Modelling / Verification
Functional system requirements (SysML model)
Safety Requirements
UML-B Model
Equivalent UML-B model of functional system requirements (SysML model)
Safety requirements represented by invariants embedded within UML-B
Rodin modelling platform with built-in verification tools including prover and animator
Model can be animated for validation against the original model

Figure 5. Schematic block diagram illustrating the translation of the SysML model and associated safety invariants into the UML-B notation

UML-B provides a diagrammatic modelling notation equivalent to those used in **UML** [5] (i.e. Class and State-machine) but with significant **semantic** and minor **syntactic** differences.

Proving a Safety Invariant

Most of the **proof obligations** from the Event-B model are **discharged automatically** by the **Rodin provers** [6]. It ensures that the model is constructed **correctly** in a **consistent** manner but do not prove anything about how the model behaves.

Safety Requirement: *"PDI Connection is established only if the Level Crossing and Electronic Interlocking version are equal."*

In UML-B:
Safety Requirement: When version-check fails in Level Crossing, the PDI Connection must not be established in Electronic Interlocking.

In Event-B:
Safety Requirement:
(LX=LX_PDI_VERSION_UNEQUAL) ⇒
(EIL≠ EIL_PDI_CONNECTION_ESTABLISHED)

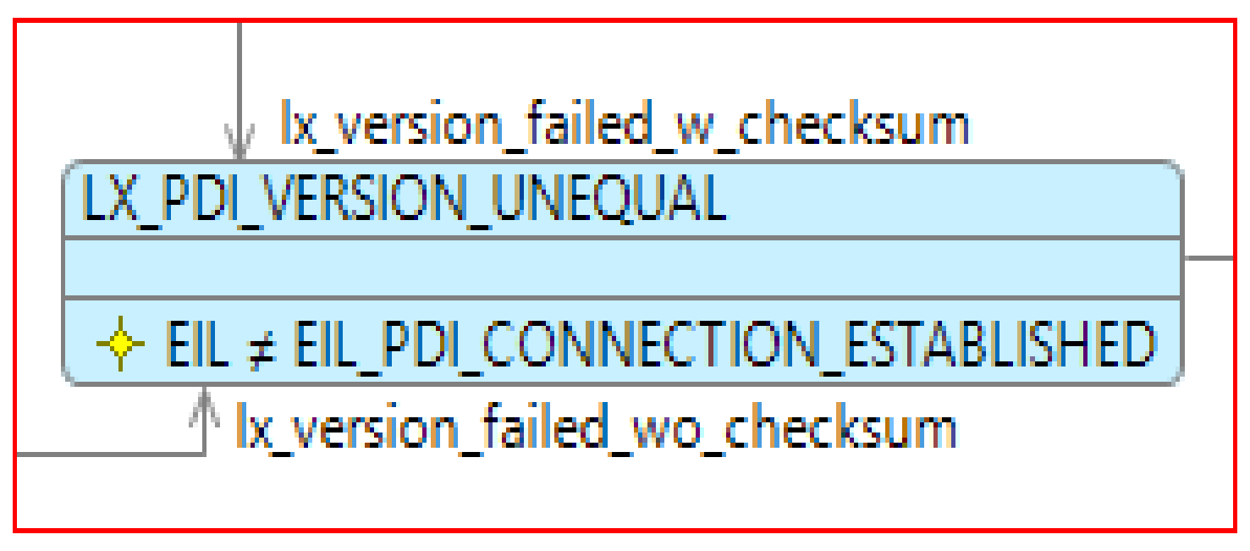
lx_version_failed_w_checksum
LX_PDI_VERSION_UNEQUAL
EIL ≠ EIL_PDI_CONNECTION_ESTABLISHED
lx_version_failed_wo_checksum

Figure 6. State and invariant in UML-B (Level Crossing side)

The **safety invariant** is discharged when all the proof obligations are discharged by Rodin.

Refinements	Proof	Automatic
m0	60	60
m1	28	28

Contact

Abdul Rasheeq; Georg Holtmann
neovendi GmbH
projects | engineering | digital transformation

Email: a.rasheeq@neovendi.com
g.holtmann@neovendi.com
Website: www.neovendi.com
Phone: +49 69 26548806

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. Fantechi, A. (2012b). The role of formal methods in software development for railway applications. In: Railway safety, reliability and security: technologies and system engineering (chapter 12), pp. 282297 (cit. on p. 18).
6. 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.
7. Fantechi, A., Fokkink, W., and Morzenti, A. (2012). Some trends in formal methods applications to railway signaling. In: Formal Methods for Industrial Critical Systems. Hoboken, NJ, USA: John Wiley Sons, Inc., pp. 6184 (cit. on p. 18).
8. ERTMS Users Group. https://ertms.be/workgroups/ccs_architecture