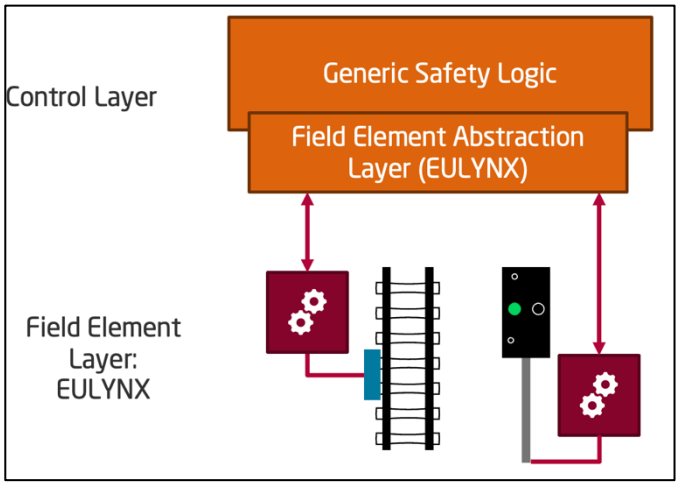
Platform-independent Safety: Building a Portable Railway Interlocking Software

Master Project, Winter 2024/2025

Railway interlockings are safety-critical systems that control the movement of trains over a railway network. Modern software-based interlocking systems are faced with a trend of centralization and virtualization ('cloud' datacenters) and use of commercial-off-the-shelf (COTS) hardware.



This means that the railway safety logic for an interlocking can be developed largely independently of the eventual execution platform. Performance constraints regarding the execution speed do not play a significant role since the safety logic is not computationally demanding and no embedded CPUs must be used in centralized datacenters.

This project therefore aims to establish a code generation and compiler toolchain for a portable digital railway interlocking software. As inputs, a formalized specification of an interlocking safety logic as well as model-based protocol definitions for the field element subsystem interfaces (Light Signals, Turnouts, Train Detection Systems) exist. The hardware platform that should be used for the initial deployment of the interlocking safety logic is based on the ARINC 653 safe segregating operating system specification from the avionics domain.

Possible project tasks include:

- Translating a deterministically specified interlocking safety logic into an IBM Rhapsody-compliant UML representation. IBM Rhapsody subsequently generates SIL 4-certifiable C code (SIL 4= highest *safety integrity level* for railway applications).
- Integrating the generated interlocking code with segregated application partitions for field element interactions on an ARINC 653-compliant operating system (Aviotech SCORPOS).
- Deploying the compiled application to the 2-out-of-2 redundant compute module using an ARINC 615-compliant data loader
- Investigating a WebAssembly-based virtualization of safe program interpretation on the compute module platform.
- Demonstration and presentation of the interlocking system in a real-life railway test field.

Participants can benefit from previous experiences with embedded and operating systems and systems programming languages. On the other hand, experience with model-based software engineering (MBSE) and higher-level programming languages is useful for the model transformation and safety analysis-related aspects of the project.

Contact:
Robert.Schmid@hpi.de (C-1.13), Andreas.Polze@hpi.de