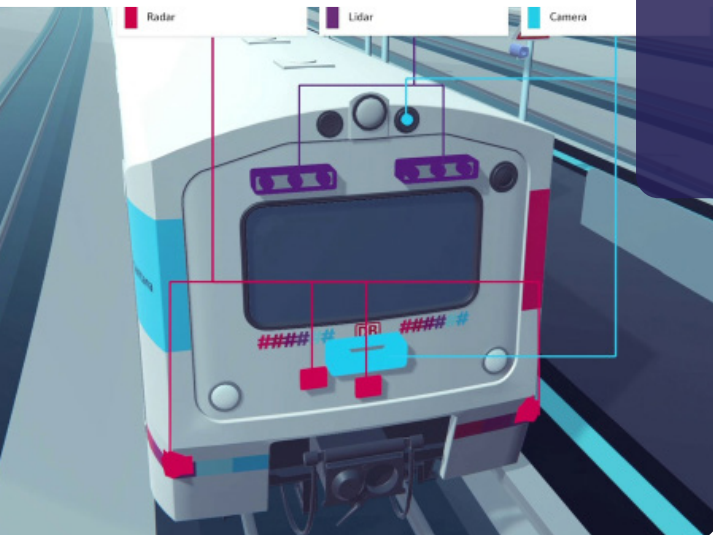


Model-centric Engineering Environment to Design the Future Railway System



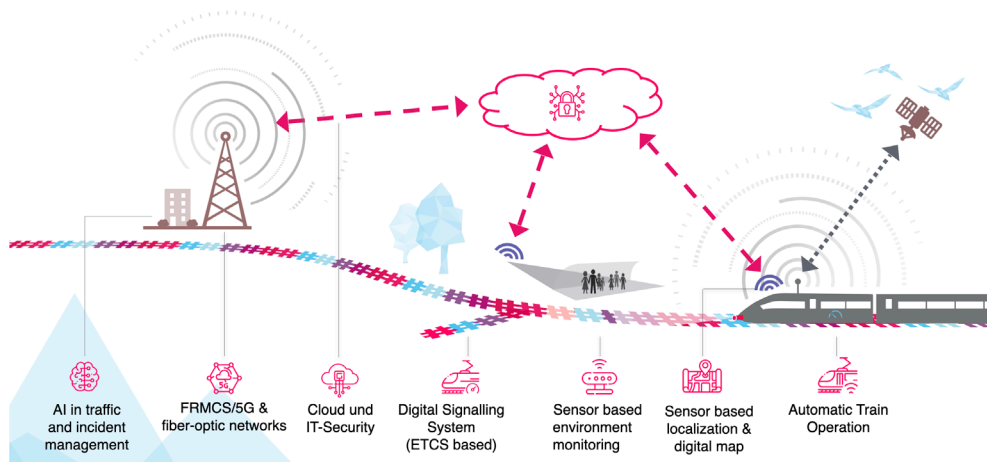
We started looking into Capella about 2 years ago and realized that it was a great match for our needs. The effort to implement the tools was not so huge in comparison to the benefits we received. - Viktor Kravchenko

Context

Today a number of EU railway operators are on a journey to define what the future of railway operations should look like. In Germany, Deutsche Bahn AG works within the sector initiative [Digitale Schiene Deutschland](#).

Next to the implementation of ETCS/DSTW technology in the first stage, the initiative aims in the second stage to improve the performance, quality and efficiency of the railway system by integrating higher degrees of automation in traffic management, train driving and infrastructure operation. This requires implementation of new technologies such as artificial intelligence, localization and perception sensors, cloud computing and 5G connectivity.

The challenge is pretty serious - to modernize and digitalize something as complex, they need a solid understanding of the railway system itself, the surrounding actors, the operational processes / workflows, functions and responsibilities. But they also need a way to design automation concepts and test those against the operational needs (i.e. The Reference Architecture).



Viktor KRAVCHENKO

Viktor Kravchenko is a Systems Architect with 10+ years of experience in multiple domains (aerospace, railways). Last 5 years mostly focused on MBSE and use of it in automation /simplification of systems engineering workflows. With DB Netz / Digitale Schiene Deutschland since 2018 leading the MBSE Toolchain team and SE / architectural work in the Sensors4Rail project.

Solution

An analysis of that scale requires knowledge and expertise across multiple domains, and involves a pretty large number of stakeholders. The ARCADIA method was just the right thing, and as Capella is freely available it could easily be tested.

The reference architecture is being developed according to a detailed process that elaborates on the basic steps of ARCADIA to give specific guidelines on every step of engineering. The process is broadly structured in the layers of ARCADIA, then influenced by the safety and RAM standard EN 50126, the systems engineering standard ISO 15288, and some specific situations. The process is described according to architecture framework standard ISO 42010 with an emphasis on precise definition of the contents of each modeling artifact.

Documents and engineering artifacts are automatically derived from the model and shared with various internal and external stakeholders. To validate the design decisions and assumptions made in the Reference Architecture, when a simulation is insufficient, prototype projects are created.

Result

Early results of the deployment of this detailed process have been positive, with reductions in the complexity of the model, and clearer, more precise and more consistent functions being identified against a reusable pattern.

The Sensors4Rail prototype project was designed in Capella nearly from the start. And for a number of common engineering reasons it was designed bottom up. It is a textbook example of a very complicated and ever-changing project with disruptive modifications, priority shifts and very high level of uncertainty. And thanks to the flexibility of Capella and ARCADIA the project was conducted with the model being the place where most of the engineering decisions were taken - be it functional, software or hardware engineering.

Today there are a number of CI/CD pipelines that deliver documents, engineering artifacts and Confluence page updates on every commit. A framework enables the projects to derive engineering artifacts from the model quickly, traverse it in all possible directions and render in any shape and form (IDL, pdf, xlsx). With Team for Capella, this framework is made available as a self-service web-based management environment.

To learn more about how they transform models into engineering deliverables you may see some [public examples](#).

