

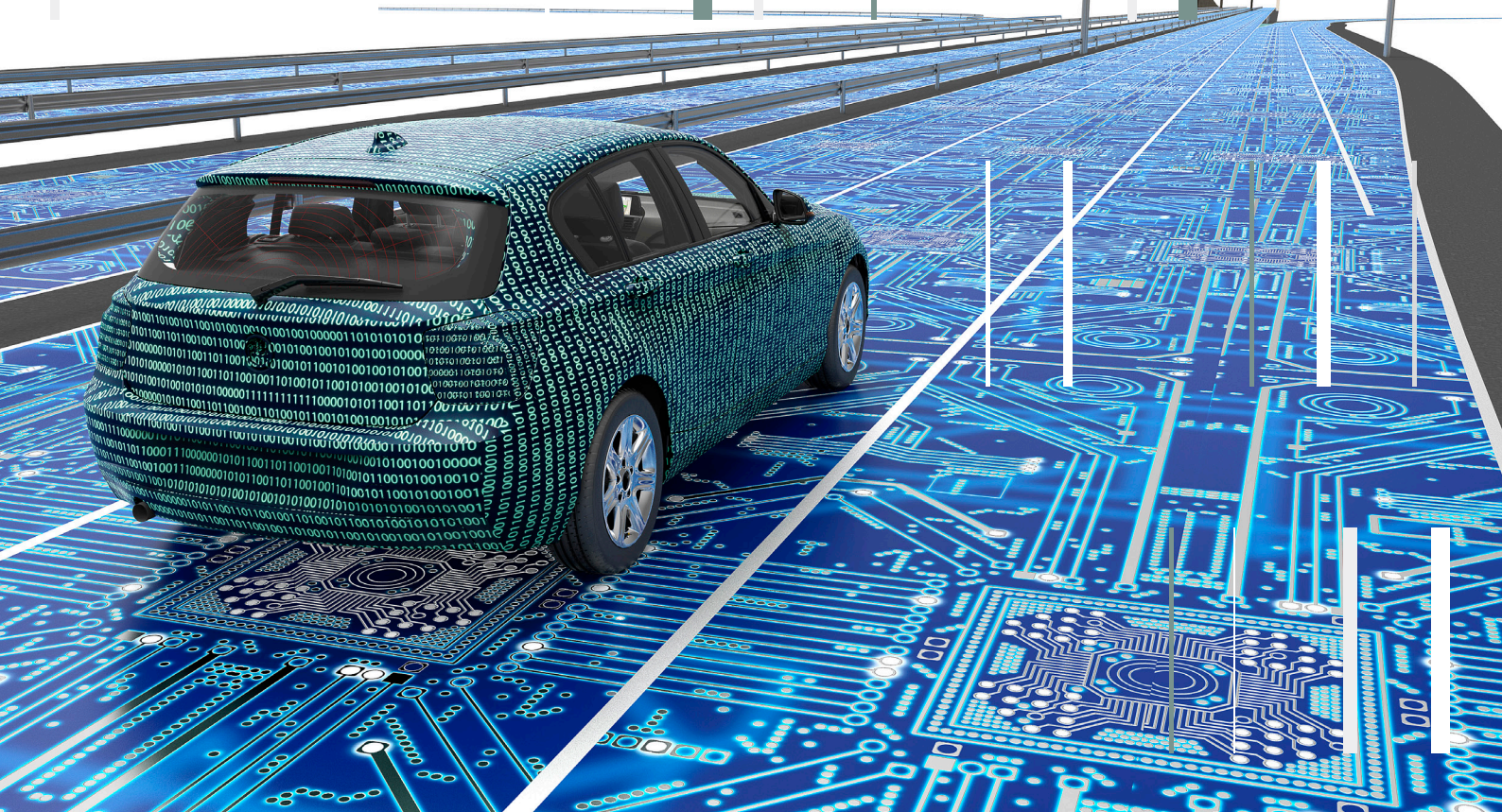
EMPHYSIS

Bridging the gap between digital simulation and embedded software with eFMI[®]

In the realm of developing new functionalities, engineers often discover that their profound understanding of the physics governing their products falls short when it comes to crafting functions for embedded targets. This is due to the demanding certification requirements of safety-critical software and the diversity in embedded ecosystems and their harsh resource restrictions on real-time timings, available computational power and memory. While this might result in a never-to-be-realised excellent initial idea of operating and controlling their product in a much smarter way, at the same time it reveals the need for a link between the digital simulation of real-world physics and embedded software leveraging on such physics models.

Success story





The ITEA project EMPHYSIS, which stands for 'Embedded systems with physical models in the production code software', was set up to overcome this challenge. The project successfully ran from 2017 to 2021, uniting 26 partners from Belgium, Canada, France, Germany and Sweden. EMPHYSIS aimed to establish a new open standard laying the foundation to develop innovative tools, facilitating the realisation of model-based functions directly in embedded software with improved code efficiency. The project carried forward the highly successful Functional Mock-up Interface (FMI®) standard for model exchange and simulation and developed a new standard, the *Functional Mock-up Interface for embedded systems* (eFMI®).

Revolutionising control function development

eFMI emerges as an open standard, offering a systematic approach to developing advanced control functions tailored for safety-critical and real-time targets. For example, it enables a faster-than-real-time capable embedded virtual twin simulation from a drivetrain model in Modelica® that can be deployed on the car's embedded control unit (ECU) for error diagnostics, automatic gear switching, backlash reduction and other drive

experience improvements. At its core, eFMI serves as a standardised workspace – a common ground – for information exchange and collaboration among stakeholders and their tooling, working on different abstraction levels and viewpoints of a common cyber-physical product. The eFMI workflow empowers developers to model systems at a higher level of abstraction, starting with a reusable, high-level, component-oriented and physics equation-based model. eFMI tools automatically transform this into a solution suited for embedded software, enabling a wide variety of advanced model-based approaches for control and diagnosis. This includes the ability to solve differential equations describing a virtual twin of a cyber-physical system that can be used, for example, as a hardware sensor replacement (virtual sensor) or provide highly advanced real-time system diagnostics for optimised control, significantly reducing overall costs.

At the end of this article, you can find the 'Technical insights' for a full understanding of the EMPHYSIS project.

The main benefit of eFMI is to provide automated processes whereas the

current state-of-the-art still suffers from organisational and technology-space gaps. In particular, engineering communities can reuse and provision their existing physics model knowledge – potentially already at an early product design stage – to the embedded software development communities tasked with the implementation of advanced control functions for a common product. Assuming tools support eFMI, each community can leverage on its methods and well-established tooling, without fostering silos. In that context, eFMI is an enabler for cooperative development and knowledge exchange.

The main benefits are:

- Overcoming vendor lock-in thanks to eFMI being an open standard.
- Accelerated development time and reduced costs thanks to automated toolchains.
- Enhanced utilisation of domain experts and their tooling by providing well-defined plug-in points for the physics modelling and embedded domains.
- Facilitating new ways for OEM-supplier collaborations with complementary viewpoints and respective toolsets along the eFMI workflow.

Project start
September 2017

Project end
February 2021

Project leader
Oliver Lenord
Robert Bosch GmbH, Germany

More information
<https://emphysis.github.io/>

Outperforming use-cases

EMPHYSIS showcased its prowess through 11 industrial use-cases in the automotive domain, a Modelica open source library with 22 examples comprising about 40 real-time simulation configurations, and 13 tool prototypes supporting eFMI of which 6 have been commercially released in the meantime (see Figure 1). The 11 industrial use-cases comprised such advanced scenarios like a semi-active damping controller with a nonlinear inverse model and nonlinear Kalman filter (DLR, e.fs TechHub), a transmission model of the whole drivetrain as virtual sensor (Volvo Cars), a dual-clutch transmission diagnosis virtual sensor (Mercedes-Benz) and an advanced emergency braking system controller (Dassault Systèmes).

BOSCH further conducted an intensive performance assessment to compare eFMI solutions with state-of-the-art handcrafted solutions for six representative application scenarios: a simple PID speed controller, an inverse drivetrain model to reduce drivetrain vibrations, an air-flow control model that is stiff, a nonlinear slider crank model for optimised crank movement, a rectifier model for battery charging requiring very compact ODE representations, and

a thermal heat transfer model requiring automatic finding of a reduced order model for high-dimensional maps. Four of the six benchmark cases surpassed the handcrafted solutions and the top eFMUs requiring 9% less data memory on the targeted BOSCH MDG1 multicore ECU. Five benchmark cases excelled in ECU runtime performance, exceeding the state-of-the-art by an average of 26% for the best-performing eFMUs, with the highest speedup being 40%. Productivity gains were evident, with a reduction in development time for five cases averaging 90%, including a remarkable 93% for the PID controller, 92% for the drivetrain controller, and 88% for the slider crank controller. The versatility of eFMI was highlighted by the air system use case, where modelling time remained constant, but embedded implementation and validation efforts plummeted, resulting in a 52% overall increase in productivity.

These figures are backed by further, non-public industry assessments performed since the end of EMPHYSIS by third parties like the Institute of Vehicle Engineering (ivH) and others. The indicators of the assessments are auspicious: eFMI reduces the time and risk to market of embedded control software for complex cyber-physical systems, increases collaboration and knowledge exchange between physics-based engineering and embedded software domains, and increases productivity. The virtual sensor industrial demonstrators – like the drivetrain transmission model of Volvo Cars – are in particular disruptive technologies since such applications simply had not been considered feasible; the development effort and risk of errors in handwritten respective C production code solutions are unacceptable.

Continuous success

The results of the EMPHYSIS project were transferred to the Modelica Association Project eFMI (MAP eFMI) founded in 2021 under the umbrella of the non-profit Modelica Association to prepare them for open access standardisation and publication, and the organisation of a volunteer community promoting and developing eFMI. Eleven of the original EMPHYSIS project partners immediately joined

MAP eFMI, demonstrating their strong confidence in and commitment to the eFMI technology. Subsequent membership applications of Altair Engineering Inc., Institute of Vehicle Engineering CO LTD, Mercedes-Benz AG and Mitsubishi Electric Research Laboratories further validate the project's relevance across diverse domains. The project is open for membership applications to continue further expansion.

A major milestone has been the provisioning of a complete eFMI toolchain backed by first commercial tool releases available on the market and covering the whole eFMI workflow from physics modelling in Modelica to production code on dedicated embedded platforms. The respective tools and their interplay are shown in Figure 1. Most are already well-known and established tooling in the physics modelling or embedded software domain, giving professionals a chance to plug-and-play with eFMI. Being well-established implications:

- **A long history and reasonable user community:** Dymola for example is available since 1993, is a de-facto reference tool for physics-based system simulation using the Modelica language and accompanied by a significant ecosystem of commercial and open-source (third-party) libraries for all kinds of engineering disciplines like electrics, batteries, fluids, thermal & HVAC, mechanics, process modelling, automotive industry, aerospace industry, naval industry, building industry etc.
- **A high-level of industry standard compliance and ecosystem integration:** TargetLink for example integrates seamless with MathWorks® Simulink®/Stateflow® models, supports a plethora of embedded targets and ecosystems like the AUTOSAR Classic and Adaptive Platforms, is certified for ISO 26262, ISO 25119 and IEC 61508 and generates production code satisfying MISRA C:2012. Likewise, TPT satisfies ISO 26262 and supports testing of ECU software and embedded control systems in all development phases such as model-in-the-loop (MiL), software-in-

the-loop (SiL), processor-in-the-loop (PiL) and hardware-in-the-loop (HiL) in various embedded ecosystems and hardware, from the AUTOSAR Classic and Adaptive Platforms to dSPACE SCALEXIO etc.

A key enabler to achieve commercial releases has been the continued tool-vendor collaboration in MAP eFMI, whose crosscheck test suite is used to conduct extensive interoperability testing. MAP eFMI also maintains two supporting open-source tools for packaging and accessing eFMUs as well as ensuring compliance with the eFMI Standard: the eFMI Container Manager and the eFMI Compliance Checker.

Last, but not least, MAP eFMI has launched its official website <https://www.efmi-standard.org/>, offering insights into the upcoming standard drafts, eFMI technology overview, available tooling, documentation, introductory material, example eFMUs and project organisation. A comprehensive 2.5 hour tutorial describing the motivation for eFMI, introducing its container architecture and providing a hands-on training in Dymola, Software Production Engineering and TargetLink was presented at the 15th International Modelica Conference 2023, and is available online at YouTube (<http://tinyurl.com/fr35hxhv>).

Bridging the engineering and embedded software development domains

In essence, EMPHYSIS was a journey of innovation and collaboration that has paved the way for a transformative technology for embedded software, and MAP eFMI is set to carry the torch forward.

The major challenge of eFMI lies not in its technological capability or feasibility – which has been demonstrated by extensive assessments as explained in the highlighted section – but in its strength to bridge the engineering and embedded software development domains: such a bridge implies bringing communities together which requires respective organisational processes. Reorganisation and redistribution of

Technical insights

The general eFMI workflow

Figure 1 summarises the general eFMI workflow for the virtual drivetrain example – one of 11 industrial use-cases of EMPHYSIS – using tooling developed in the EMPHYSIS project and now commercially available.

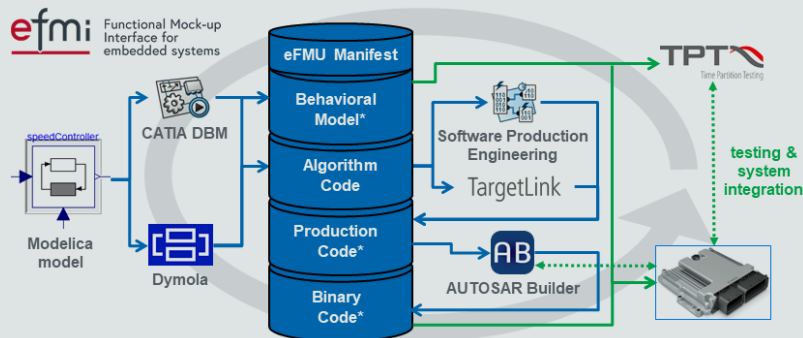


Figure 1: eFMI container architecture and workflow

The starting point is a physics equation-based Modelica model of the drivetrain, developed using the Dymola Modelica IDE (Dassault Systèmes). Besides the virtual design and verification via offline simulations, Dymola can generate a new eFMU (embedded Functional Mock-up Unit) with an algorithmic solution of the drivetrain model in an eFMI Algorithm Code container. Although deterministically computable, such algorithmic solutions are still of a high, target-independent abstraction level. Dedicated eFMI production code generators have to translate them to C code suited for the actual target platform. In the figure above, TargetLink (dSPACE) and Software Production Engineering (Dassault Systèmes) are used. The resulting C11 production code is added as an eFMI Production Code container to the eFMU. To integrate this production code into the system and deploy it on the target hardware, AUTOSAR Builder (Dassault Systèmes) adds required ©AUTOSAR Adaptive Platform code in an eFMI Binary Code container, ultimately provisioning the faster-than-real-time drivetrain simulation as an AUTOSAR component. Offline simulations of the original Modelica model in Dymola are further used to generate reference trajectories for software- and hardware-in-the-loop (SiL, HiL) tests, added as an eFMI Behavioral Model container to the eFMU. The final tests on the target hardware are conducted with the TPT (Synopsys) testing IDE. Note, that the eFMI Standard requires that each container, besides the actual implementation artefacts, provisions a dedicated manifest with meta-information like checksums and dependencies on other eFMU containers; an eFMU manifest lists all the containers.

The eFMU is a standardised workspace where different stakeholders and tools contribute towards a final solution by means of updating existing or adding further development containers derived from the results of higher abstraction levels along the chain from physics-based equations down to target specific binaries ready for deployment. The eFMI Standard thereby only defines “what” the representations and results of an abstraction level are, not the “how” to produce such; the actual magic is in the generating tools that can be well-established expert tools in their respective domain.

responsibilities to break silos in huge organisations is a sensitive social, political and bureaucratic subject.

Put simply: Investing in eFMI is a must to tackle the future challenges in embedded control.