MODAPTO [101091996]: Modular Manufacturing and Distributed Control via Interoperable Digital Twins

6.1.1 Introduction standard FMI

2025-08-11

The Functional Mock-up Interface (FMI) standard represents a revolutionary advancement in simulation technology, enabling unprecedented interoperability between modeling and simulation tools from different vendors. The Functional Mock-up Interface (FMI) is a free standard that defines an interface to exchange dynamic models using a combination of XML files, binaries and C code. It's supported by 240+ tools and maintained as a Modelica Association Project.

Developed through international collaboration, FMI addresses the fundamental challenge of model exchange and co-simulation in increasingly complex cyber-physical systems.

The genesis of FMI stems from the recognition that modern products, particularly in automotive and aerospace industries, require integration of multiple engineering domains. For many years, product modeling and simulation have established themselves as core elements of product engineering for advanced components and complex systems. Therefore, we use many specialized simulation tools, and it is a hard task to exchange models between different departments and partners. In order to reach optimal system behavior, it is essential to create overarching simulation models consisting of combined models of the different components and subsystems.

Each domain typically uses specialized tools optimized for specific physics or engineering disciplines, creating integration challenges when system-level behavior needs to be analyzed.

FMI 3.0, released in May 2022, represents a major evolution of the standard, addressing limitations identified through years of industrial application. On May 10, 2022, eight years after the release of FMI 2.0, version 3.0 of the FMI standard was released! This is a major step forward, as this resolved the previous shortcomings and enables totally new use cases.

The new version introduces several groundbreaking features including support for array variables, advanced event handling through clocks, binary data type support, and enhanced numerical robustness through intermediate variable access.

The standard defines three distinct interface types, each serving specific simulation needs. Model Exchange (ME) provides a standardized way to describe differential algebraic equation systems, requiring the importing tool to provide numerical solvers. This interface is ideal for situations where tight integration with existing solver infrastructure is desired. Co-Simulation (CS) encapsulates both the model and its solver, enabling distributed simulation where each FMU maintains its own internal state and time integration. The FMI Standard provides three interface types for difference aspects of models: FMI for model exchange, FMI for co-simulation, FMI for scheduled execution (since FMI 3.0)

The introduction of Scheduled Execution (SE) in FMI 3.0 addresses the growing need for real-time simulation and hardware-in-the-loop applications. Synchronous Clocked Simulation aims at scenarios where the cause and exact time of multiple simultaneous events can be unambiguously conveyed. At the same time, scheduled execution facilitates real-time simulation among black-box models by giving practitioners a fine-grained control (compared to version 2.0 of the FMI standard) over when specific tasks inside the black-box simulation unit can be executed.

This interface type enables precise control over execution timing, making it suitable for integration with real-time operating systems and embedded platforms.

The technical architecture of an FMU follows a well-defined structure. At its core, an FMU is a ZIP archive containing an XML model description file, compiled binaries for one or more platforms, and optional source code. The modelDescription.xml file provides comprehensive metadata about the model, including variable definitions, unit specifications, dependencies, and capability flags. This self-contained packaging ensures that all necessary information for model execution travels with the model itself, simplifying deployment and version management.

Checking FMUs for standard conformity is crucial for ensuring reliable interoperability. The FMI standard provides detailed specifications for both static (structural) and dynamic (behavioral) requirements. Tools like the FMU Compliance Checker validate structural conformity by examining the modelDescription.xml schema compliance, verifying required function presence, and checking consistency between declared capabilities and actual implementation. Dynamic validation involves executing test scenarios to verify correct behavior of state initialization, time integration, event handling, and data exchange protocols.

For analytical applications, FMUs offer unique advantages in encapsulating domain-specific knowledge while maintaining standardized interfaces. The main underlying class of models for this Submodel are DAEs, models described by differential algebraic equations. It describes rudimentarily the content of the model.

Analytics-focused FMUs might implement statistical models, machine learning algorithms, optimization routines, or specialized calculations like Life Cycle Assessment. The standardized interface ensures these analytical capabilities can be seamlessly integrated into larger simulation workflows.

The practical implementation of FMUs for analytics requires careful consideration of computational efficiency and numerical stability. Since analytical FMUs often process large datasets or perform iterative calculations, optimization of the internal algorithms is crucial. The FMI standard's support for direct memory access through binary data types in version 3.0 significantly improves performance for data-intensive analytical applications. Additionally, the ability to expose intermediate variables

allows downstream tools to monitor convergence and numerical health of analytical algorithms.

Integration of FMUs into standardized Digital Twins represents the convergence of simulation technology with Industry 4.0 architectural principles. The AAS opens up new possibilities for digital twins. A virtual asset could simulate the physical asset and be connected via the AAS to a virtual or real production system. This would enable virtual testing and commissioning of production modules, lines and plants, as also that of production control, automation and orchestration layers. The Asset Administration Shell standard specifically includes provisions for simulation models through the “Provision of Simulation Models” submodel template (IDTA 02005-1-0), which defines how FMUs should be described, stored, and accessed within the AAS framework.

The implementation process for creating analytics-enabled FMUs follows established software engineering practices adapted for simulation models. Development begins with mathematical model formulation, followed by implementation in a suitable programming language (typically C/C++ for performance). The FMI SDK or similar frameworks provide templates and utilities for implementing the required callback functions. Careful attention must be paid to memory management, thread safety, and numerical robustness. Testing should cover not only functional correctness but also performance under various operating conditions and compatibility with different importing tools.

Version compatibility remains a critical consideration in FMI deployments. For FMI 1.0/2.0 and 3.0 interoperability the bridging between the different variable types of 1.0/2.0 and 3.0 and the handling of arrays needs to be taken into account. In projects that employ the SSP standard, the variant handling facilities of SSP can be used to package multiple versions of an FMU into one package.

Organizations must carefully plan their migration strategies, potentially maintaining multiple FMU versions during transition periods. The System Structure and Parameterization (SSP) standard provides mechanisms for managing multi-version deployments and complex system configurations.

References

1. Blochwitz, T., et al. (2011). “The Functional Mock-up Interface for Tool independent Exchange of Simulation Models”. Proceedings of the 8th International Modelica Conference, Dresden, Germany.

2. Modelica Association (2022). “Functional Mock-up Interface 3.0 – Standard Document”. Available at: https://fmi-standard.org/docs/3.0/

3. Gomes, C., et al. (2018). “Co-simulation: A survey”. ACM Computing Surveys, 51(3), Article 49.

4. IDTA – Industrial Digital Twin Association (2023). “IDTA 02005-1-0: Submodel Template for Provision of Simulation Models”. Version 1.0.

5. Krammer, M., et al. (2019). “The FMI 3.0 Standard Interface for Clocked and Scheduled Simulations”. Proceedings of the 13th International Modelica Conference.

6. Asset Administration Shell Reading Guide (2022). “Submodel Templates and Their Application”. Platform Industrie 4.0.

7. Bertsch, C., et al. (2021). “FMI 3.0 – New Features and Applications”. Proceedings of the 14th International Modelica Conference.

8. System Structure and Parameterization (SSP) Specification 1.0 (2019). Modelica Association Project.