



FMI 3.0 Development

-

Status, Roadmap, Layered Standards

Andreas Junghanns (Synopsys)
Torsten Blochwitz (ESI Group)
Christian Bertsch (BOSCH)

FMI 3.0: Status

- 340+ issues closed on FMI 3.0
- Draft available
 - <https://github.com/modelica/fmi-standard/tree/master>
 - Highest quality of all standard versions we have released (validated C-code, XML, XSD...)
- Currently FMI 3.0 Beta 2
 - fixed feature list
 - mostly stable API and XSD
- 2 PlugFests held in 2021 (Jul, Sep)
 - Participating: 3DS, Altair, AVL List, BOSCH, ETAS, dSPACE, ESI ITI, PMSF, Siemens PLM, Synopsys
- 29 FMUs in Sandbox
 - https://github.com/modelica/fmi-design/tree/fmi3-plugfest/FMI_3.0/sandbox/fmus/3.0-beta.2
- More available in reference FMUs
 - <https://github.com/modelica/Reference-FMUs>
- FMPy already supports the current FMI 3.0 Beta 3 import
 - <https://github.com/CATIA-Systems/FMPy>

FMI 3.0: Roadmap

- Currently working towards FMI 3.0 Beta 3
 - Minor changes to the state diagram
 - Clarifications
 - Driven by prototype implementations!
- 2 more PlugFests planned:
 - October 26th and 27th: Beta 3
 - November 30th and December 1st
 - YOU! are welcome to join with your prototype!
- **We are planning to release FMI 3.0 early 2022**
 - **...if all features are sufficiently tested in prototype implementations!**

Layered Standards

- Layering standards on top of FMI allows us to keep the core FMI standard clean and free of domain-specific features
- Layered standards will define extensions **compatible** with FMI 3.0 for domain specific FMUs
- Examples:
 - Domain specific terminals
 - vECUs with XCP support:
https://github.com/modelica/fmi-standard/blob/master/docs/X_layered_XCP.adoc
 - FMUs with network communication:
<https://github.com/modelica/fmi-standard/pull/1177>
 - Including 3D-Visualization in FMUs which represent multi-body simulation models
 - Inclusion of meta data

Beyond FMI 3.0

- Split focus between core features and layered standards
 - Layered standards may also be developed outside the Modelica Association (and later potentially endorsed or adopted)
- Coordination with other Modelica Association Projects (Modelica Language, DCP, SSP, eFMI)
 - ...to ensure compatibility and interoperability of all Modelica Association Standards
- Potential core standard improvements:
 - Performance, e.g.
 - Lifetime of objects?
 - Access to Sparse matrices?
 - New features, e.g.
 - PDEs?
 - “Cloud-native” FMUs?