Understanding FMI CS and ME

Overview, comparison and lessons learned

Functional Mock-up Interface

Dr. Clément Coïc

Understanding FMI – Co-Simulation & Model Exchange

Goal

To enable cross-tool exchange of dynamic simulation models

Mindset

Open and broad adoption while protecting IP

(Many tool vendors contributed to the FMI specification – initially during the Modelisar ITEA 2 European project. Anyone can join the project and contribute.)

Bonus

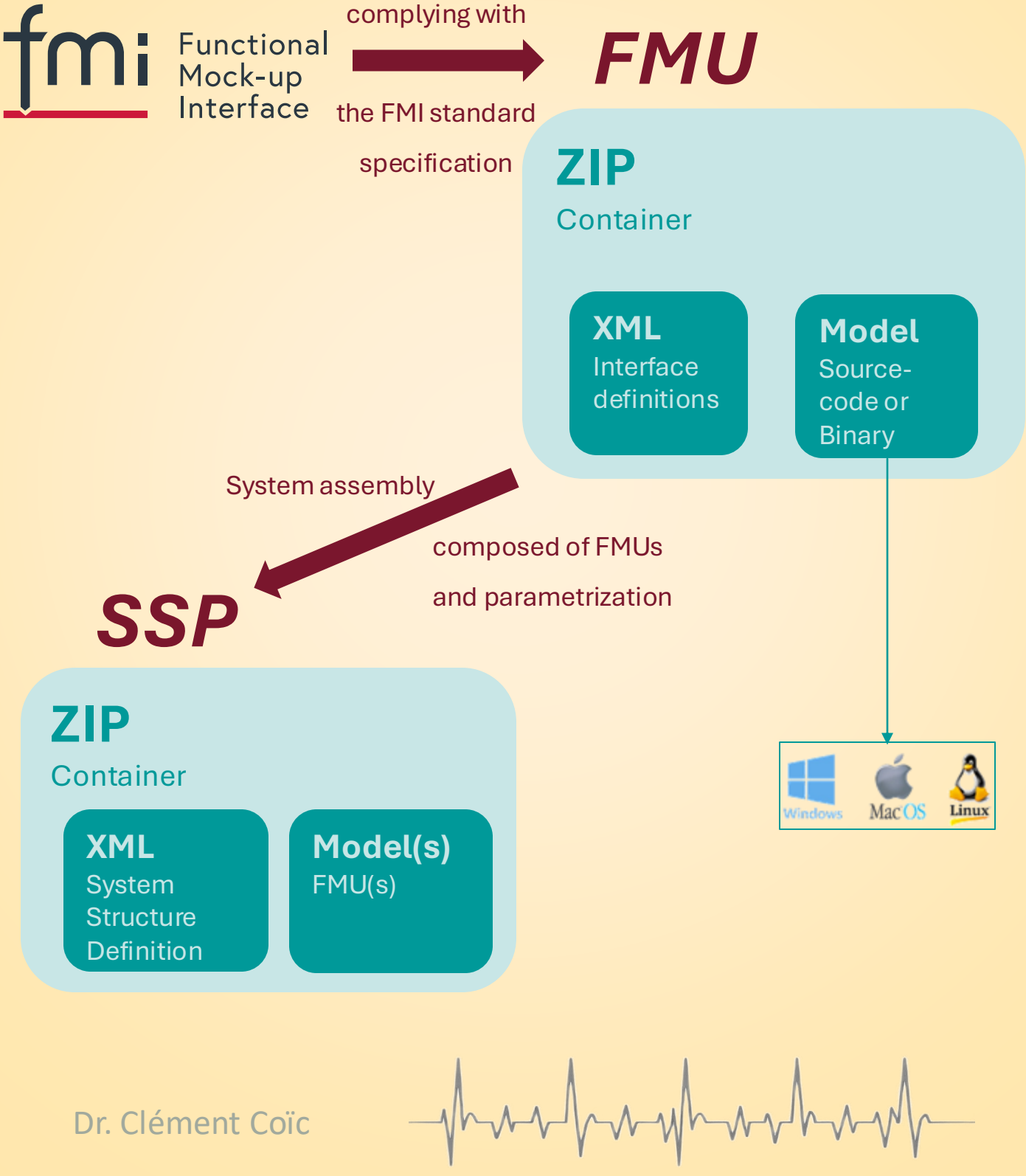
Also used for model export on real-time target

Useful links

- FMI website
- Tools support
- Specification (3.0.1)
- SSP website

Dr. Clément Coïc

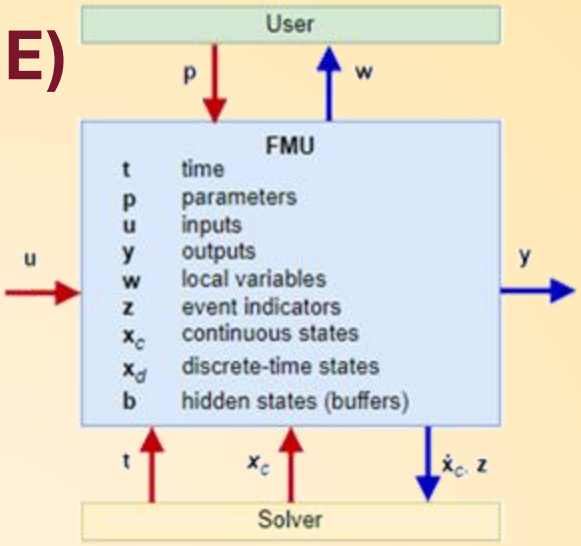
Understanding FMI – Co-Simulation & Model Exchange



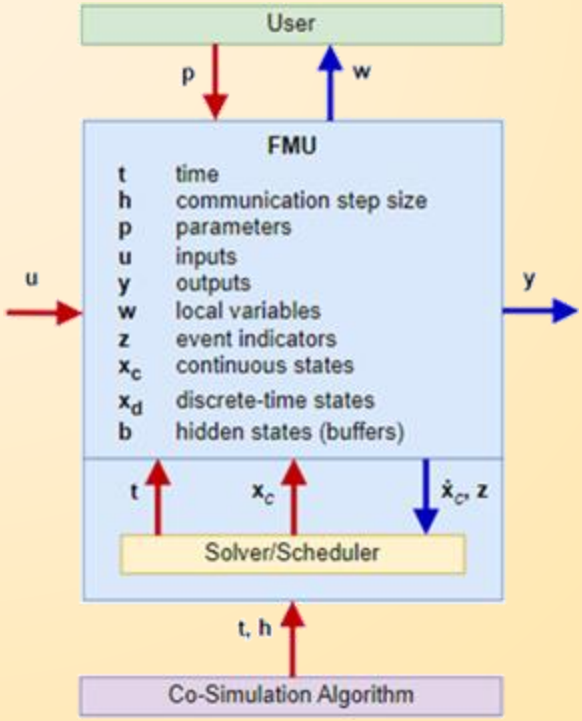
Dr. Clément Coïc

Model Exchange (ME)

Solver not include
Derivatives at the interfaces
Requires integration



fmi Functional Mock-up Interface



Co-Simulation (CS)

Solver included
Can be coupled
Can be directly simulated

When to use CS or ME?

Particular cases	ME	CS	Comment
Importing tool support	🔗 Tools support		Not all tools support all types of FMU
Solver availability	❌	✅	E.g. Finite Element solver might not be available in system simulation tool
Model coupling with different dynamics	❌	✅	It might be preferrable not to make all models run at the highest dynamics
Model numerics require specific solver	❌	✅	ODE vs DAE, also each solver has its stability region. Locking solver in FMU.
Model includes sample data	❌	✅	Not ODE so not an exposed interface
Direct feedthrough in model	✅	❌	Not allowed for CS
Unsure about usage	✅	❌	ME is more flexible yet let robust

Lessons learned.

Few “Top-Level” parameters can be modified.

- In modelDescription.xml, search for `causality = parameter` and `variability = fixed` or `tunable`.
- Make sure to propagate the parameters of interest at top level before exporting your model as an FMU.
- Some parameters might become `Structural`, which means you cannot change them anymore after compilation – even if at top-level. Typically, this happens for conditions to `if-statements` or to structural changes (could be the size of vectors, conditional connect statement, etc.).

Some parameters can be pointing to an external file

- Needs to be set up before FMU generation
- How to do it

A model in a tool might not give the same result than its FMU in the same tool

- Especially true if the inputs are not connected → FMI standard specifies that unconnected inputs are set to `0`
- Especially true if the inputs are not differentiable → a tool might be able to smoothen inputs while the FMU might not
- The solver might not be the same by default when running an FMU versus an embedded model.

Lessons learned.

Different tools can give slightly different results for edge cases

- The solver might not be the same by default. And even a given solver might be implemented differently.
- A tool might be constrained by its modeling language.
- Precision / Accuracy might be different…

Co-Simulation means discretized inputs

- FMUs exchange data at each communication step and values are hold in between
- The FMU sees steps at every changing input, hence discontinuous.
- Be cautious that your model does not require a derivative of the input.

FMUs can require tool-vendor license calls

- Based on vendor implementation, it could also require an installation of the tool on the target
- Based on the vendor, the license call can be removed for a fee
- Some vendors do not require a license, and some add legal (EULA) limitation

Lessons learned.

FMUs is not the response to everything

- The model is compiled or in source code
- The model causality is fixed
  (You would need two different FMUs for V=R*I and I=V/R)
- Hard or impossible to modify with the model afterwards
- Hard to understand what a model does (correlation used, fidelity, etc.) as it is not necessarily possible to look inside
- No structural changes can be done, e.g. add a friction between two elements.

An assembly of FMUs has its limitations

- The model structure level cannot be optimized (e.g. two resistances in a row cannot be lumped into one equivalent)
- Algebraic FMUs can create numerical issues
  (requiring additional non-linear solvers)
- Communication between FMUs can create discontinuities

Recommendations

- Stay in modeling tool as long as possible to benefit from the tool flexibility and features
- Export in the right format (CS/ME) when needed to interface with other models from another tool or dynamics
- When possible, stick to one FMU imported in another environment rather than integrating two FMUs together.

Conclusion

FMI is great:

- For coupling models that are coming from different tools
- For coupling models that have different dynamics
- For sharing models as “black-box”

FMI brings some limitations:

- On model usage – acausal vs. causal
- On model evolution – such as structure or parametrization
- On model coupling stability

Missing something?

Comment if you need any further clarification or insights.