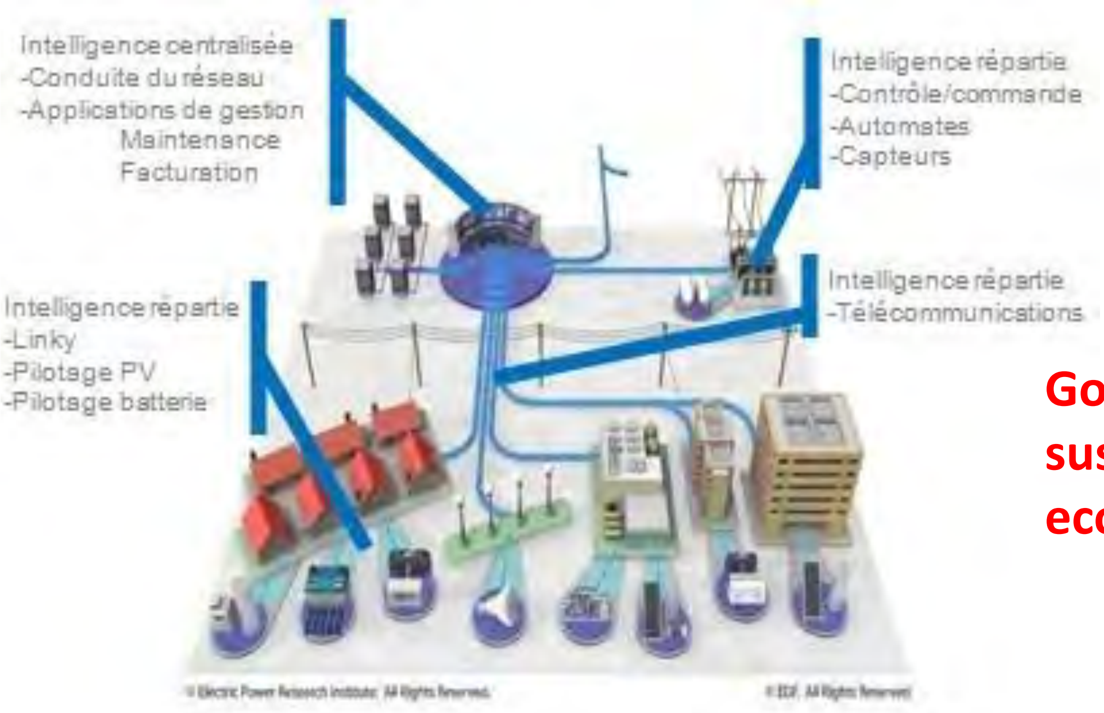
Plate-forme de co-simulation DACCOSIM 2016

Jean-Philippe TAVELLA — EDF R&D département MIRE & RISEGrid, Clamart, France
Cherifa DAD — CentraleSupélec & UMI GT-CNRS 2958 & RISEGrid, Metz, France
Stéphane VIALLE — CentraleSupélec & UMI GT-CNRS 2958 & RISEGrid, Metz, France

What is RISEGrid ?

Research Institute for Smarter Electric Grids

- **A French research institute dedicated to the Smart Grids**
  - *Launched by CentraleSupélec and Electricité de France in Dec. 2012*
  - *About 20 people (academic & industrial researchers, PhDs, post-docs)*
  - *For the study, modelling and simulation of Smart electric Grids*



Goal: efficiently supply a sustainable energy in an economical & secure environment

What is RISEGrid ?

Research Institute for Smarter Electric Grids

- A French research institute dedicated to the Smart Grids
  - *Launched by CentraleSupélec and Electricité de France in Dec. 2012*
  - *About 20 people (academic & industrial researchers, PhDs, post-docs)*
  - *For the study, modelling and simulation of Smart electric Grids*

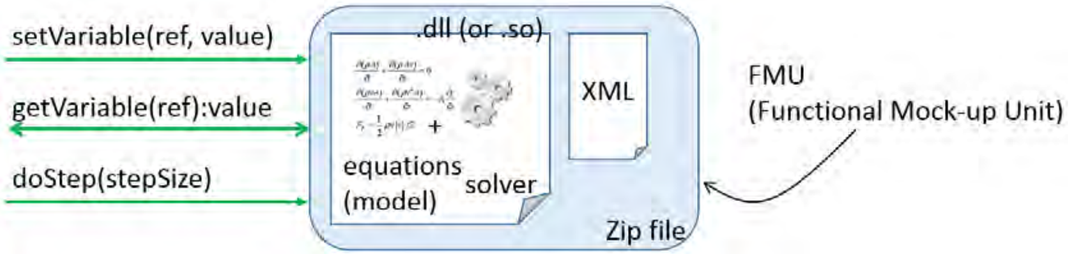
- Some of the research topics with PhDs in progress
  - *Operation of the power systems with lots of renewables*
  - *Stability and P=C balance with the help of the storage & load control*
  - *PLC use in the context of the smart metering*
  - *Harmonics contribution of an installation connected to the grid*
  - *Models and tools for a distributed cosimulation of the Smart Grids*
  - *Simulation of the information systems in Smart Grids*

http://www.supelec.fr/342_p_38091/risegrid-en.html

Advanced M&S for the Smart Grids?

- **Model the Smart Grids (Cyber-Physical complex System)**
  - *EDF R&D multi-disciplinary modeling*
  - *Modelica to unify the physics modeling thanks to internal business libraries (GridSysPro, BuildSysPro, PlantSysPro & ThermoSysPro)*
  - *Dymola as the main modeling tool for the Physics*
  - *IEC 61131-3, UML/SysML & Simulink models for the cyber part*

- **Simulate the Smart Grids**
  - *Dymola & Control Build for the FMUs export (FMI 2.0 for CoSimulation)*
  - *Next Papyrus with a Moka module to export FMUs*
  - *DACCOSIM as a distributed Master Algorithm for cosimulation with strongly coupled FMUs*



Why DACCOSIM?

- **A solution for simulating IP protected models**
  - *FMUs are “black boxes”, only interfaces are known*

- **A solution for simulating large-scale systems**
  - *Time-continuous simulations in parallel possibly distributed over several cluster nodes*

- **A solution for simulating multi-physics systems**
  - *Generic interfacing between multi-physics components*

- **A (future) system of Russian dolls**
  - *A cluster of FMUs as a parallel FMU*

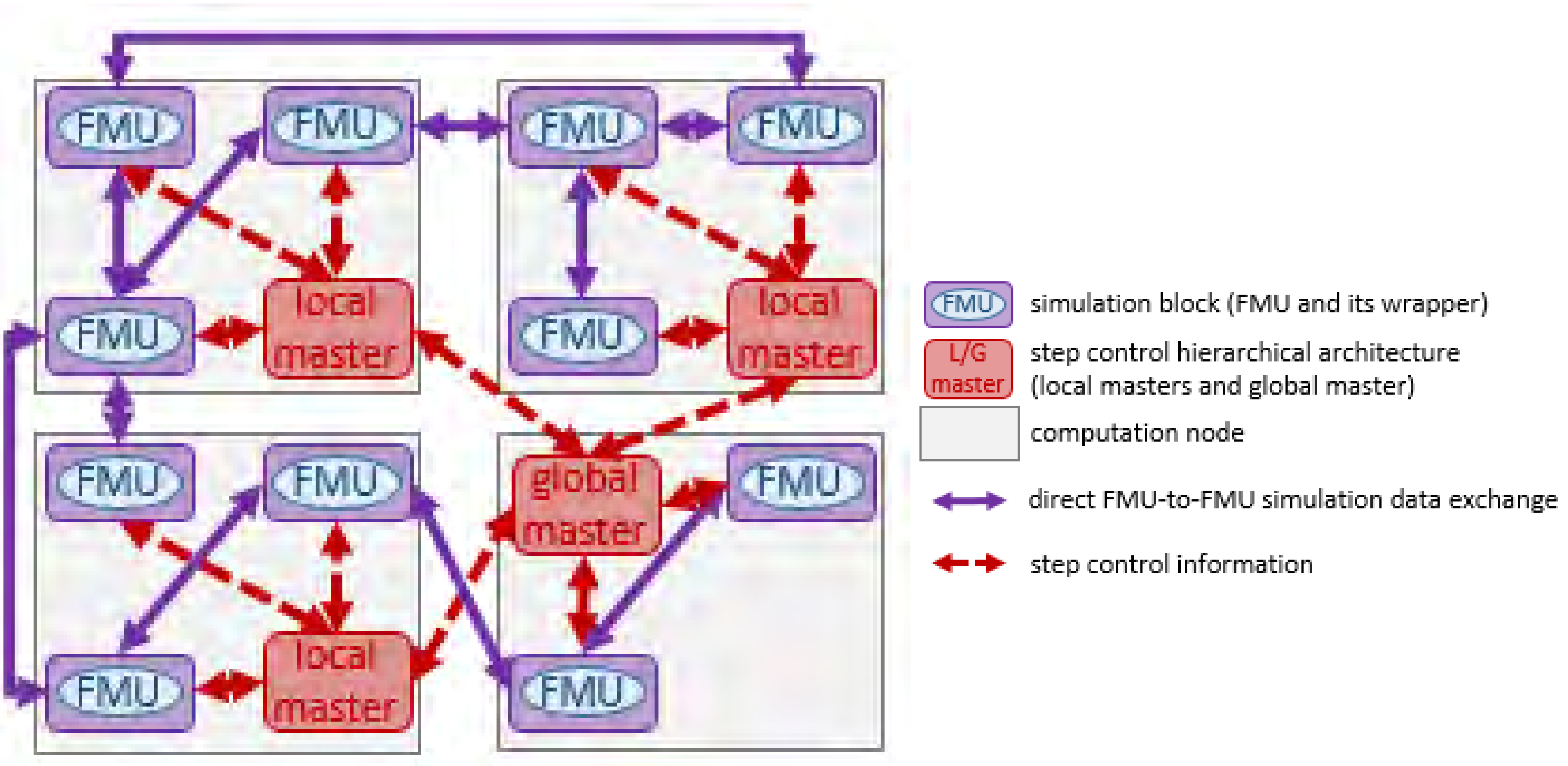
What is DACCOSIM?

- **A parallel & distributed Master Algorithm fully compliant with the FMI 2.0 for cosimulation standard providing**
  - *Co-initialization of the calculation scheme*
  - *Multi-threaded and distributed execution (deployment in parallel Python)*
  - *Variable time step methods*
  - *Adams-Bashforth multiple steps methods recently developed*

- **A Graphical User Interface making easier the construction of case studies**
  - *Calculation graph design, configuration and validation*
  - *Automatic case study execution*

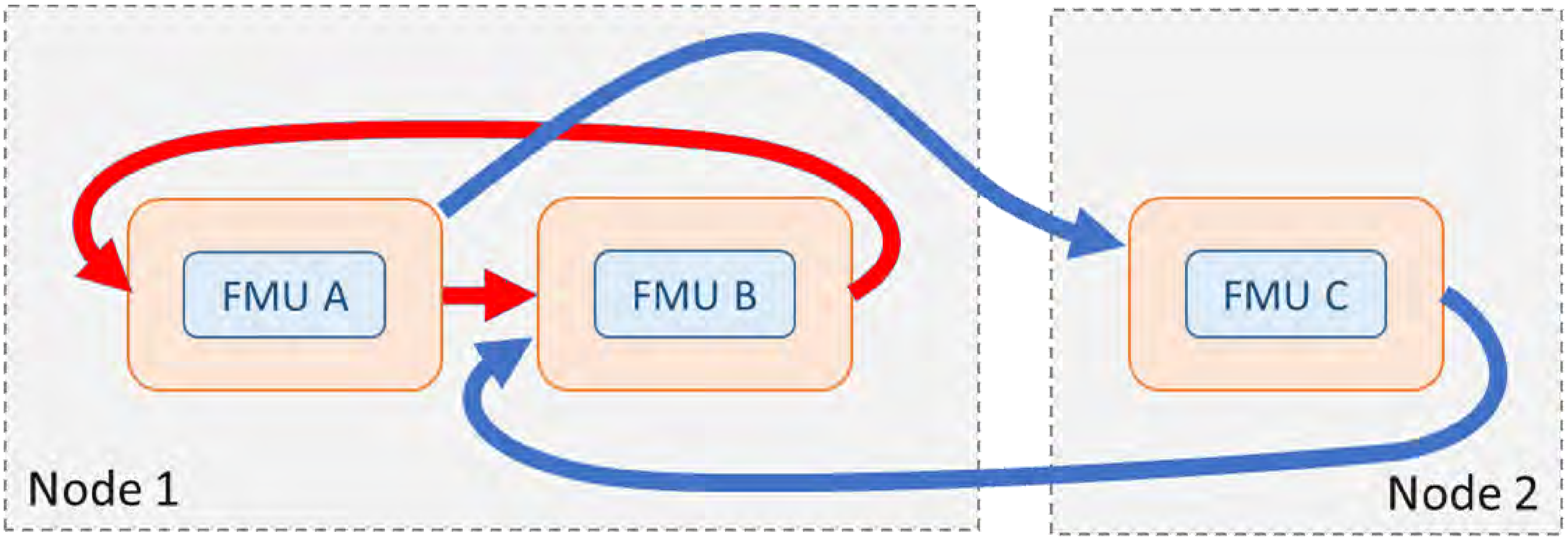
- **A free distribution**
  - *Windows/Linux 32/64 bits under the AGPL license*
  - *One version per year, last one is DACCOSIM 2015, Oct. 2015*

Parallel & Distributed Master Architecture



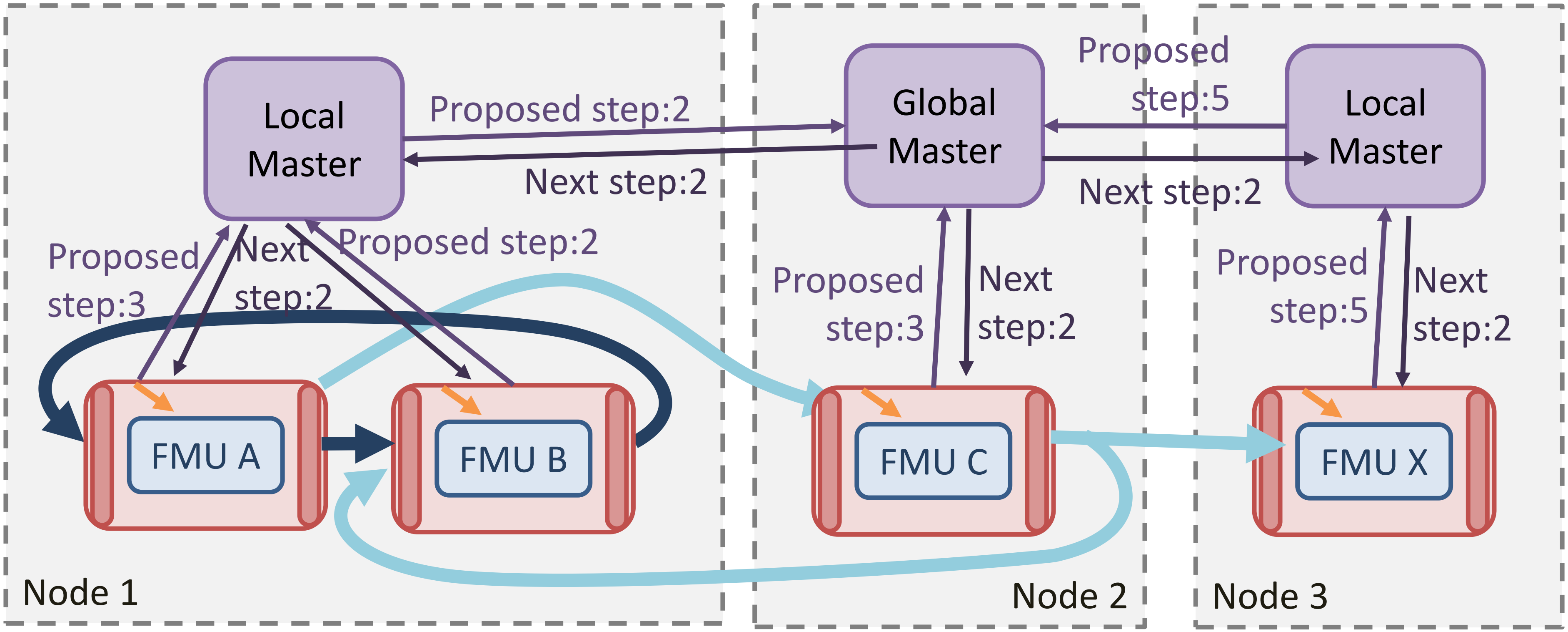
DACCOSIM master: key features

- **Direct FMU-to-FMU data communication**
  - ***Each FMU is embedded in a wrapper (C++ or Java)***
    - Allows asynchronous communication in parallel with computation
  - ***FMUs can be on the same machine or on separate cluster nodes***
    - ***FMUs on separate nodes: communication with TCP***
    - ***FMUs on same node: shared queue or ØMQ communication with inproc***



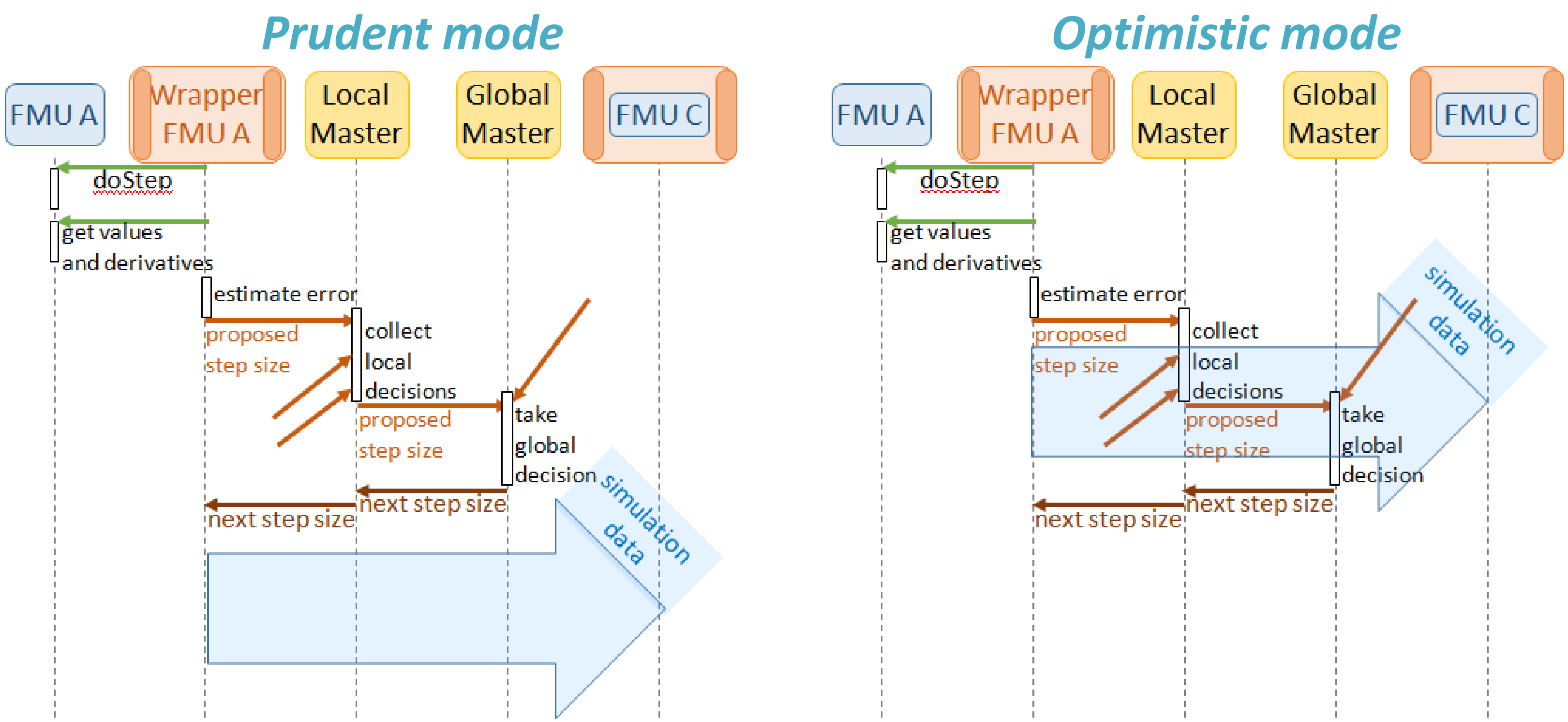
DACCOSIM master: key features

- **A hierarchy of control masters to synchronize rollbacks**
  - *Under the constraint all FMUs must share the same step size*



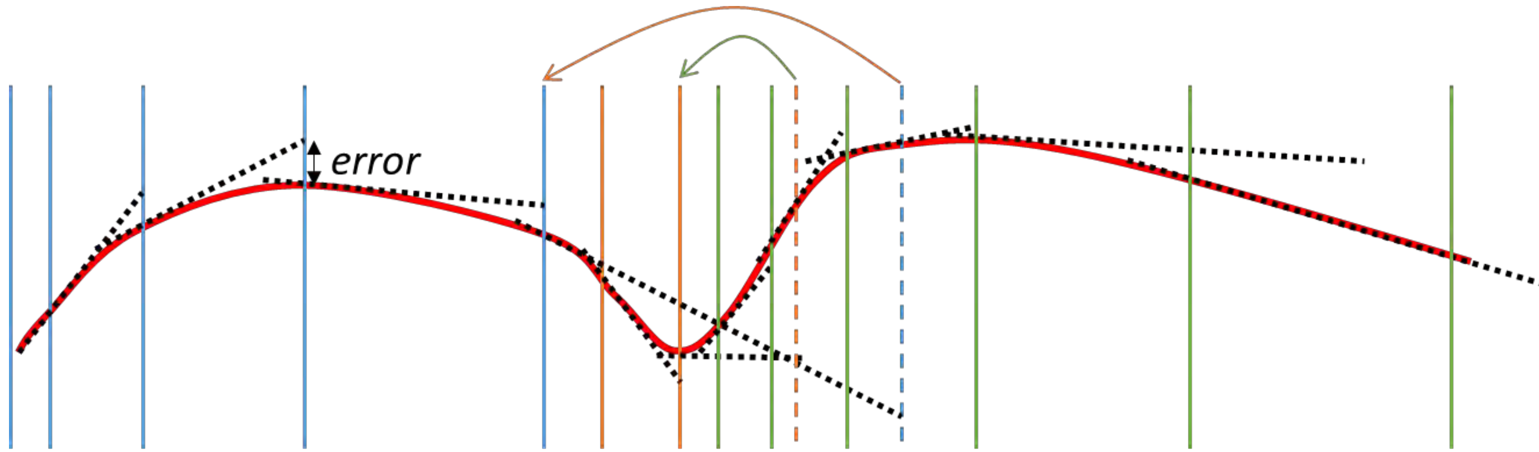
DACCOSIM master: key features

- **Common step size negotiation**
  - ***All FMUs share the same step size***
  - ***The simulation timing is controlled by the hierarchy of masters***
  - ***Two data synchronization methods: prudent & optimistic modes***



DACCOSIM master: key features

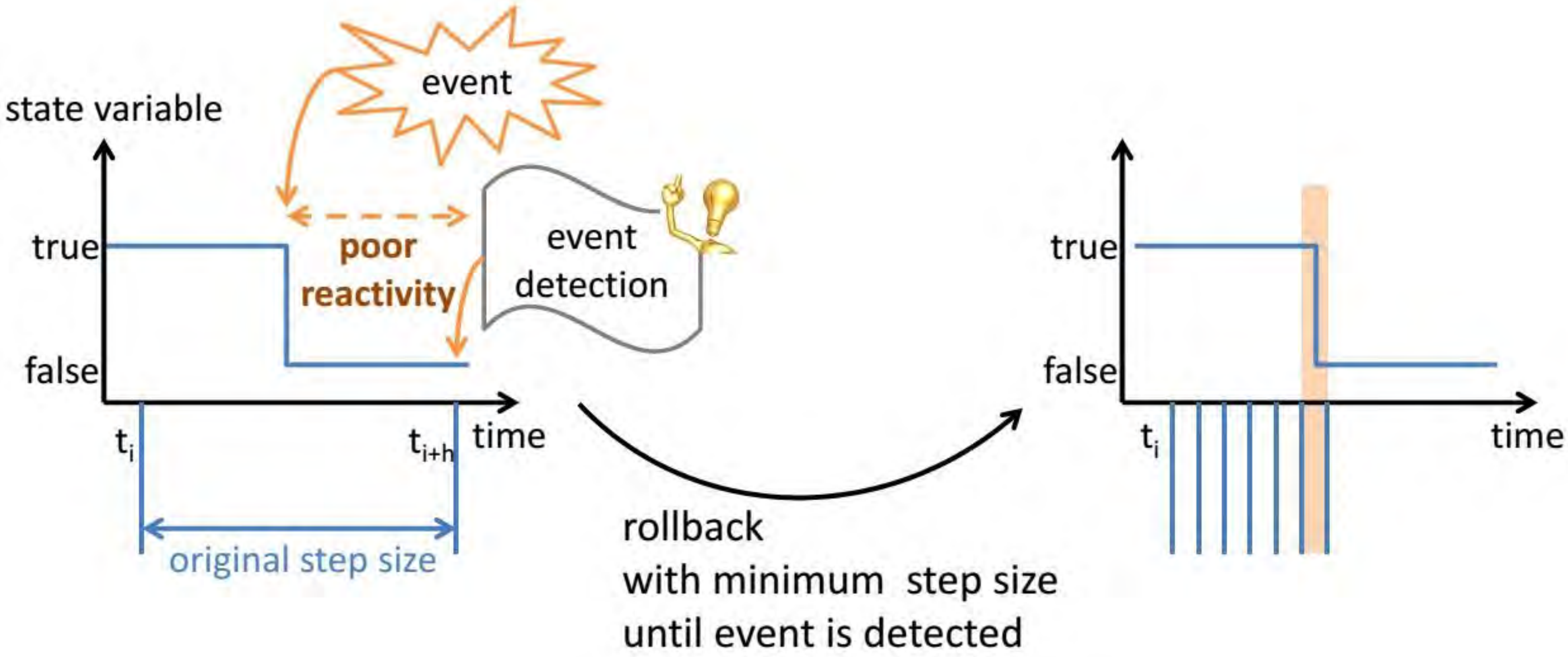
- **Several types of time step control strategies**
  - ***Constant step size***
  - ***Euler & Richardson adaptive step size (exploit fully FMI 2.0 interface)***



- **Generic master design** that enables :
  - Either to ***execute DACCOSIM as a standalone application***
  - Or to ***control DACCOSIM from an higher level middleware***
    - *Work in progress : DACCOSIM as a "super-FMU"*

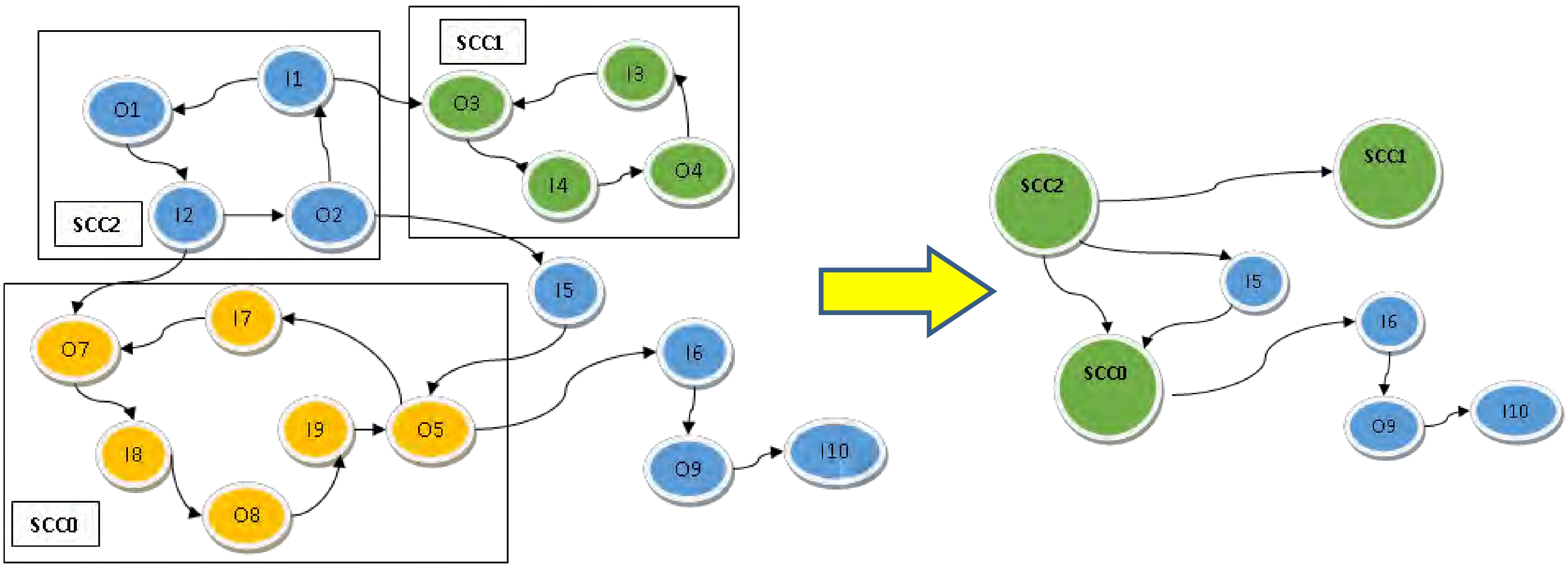
DACCOSIM master: key features

- **State event detection**
  - *Provisional solution till next FMI 2.1 is released (hybrid cosimulation)*

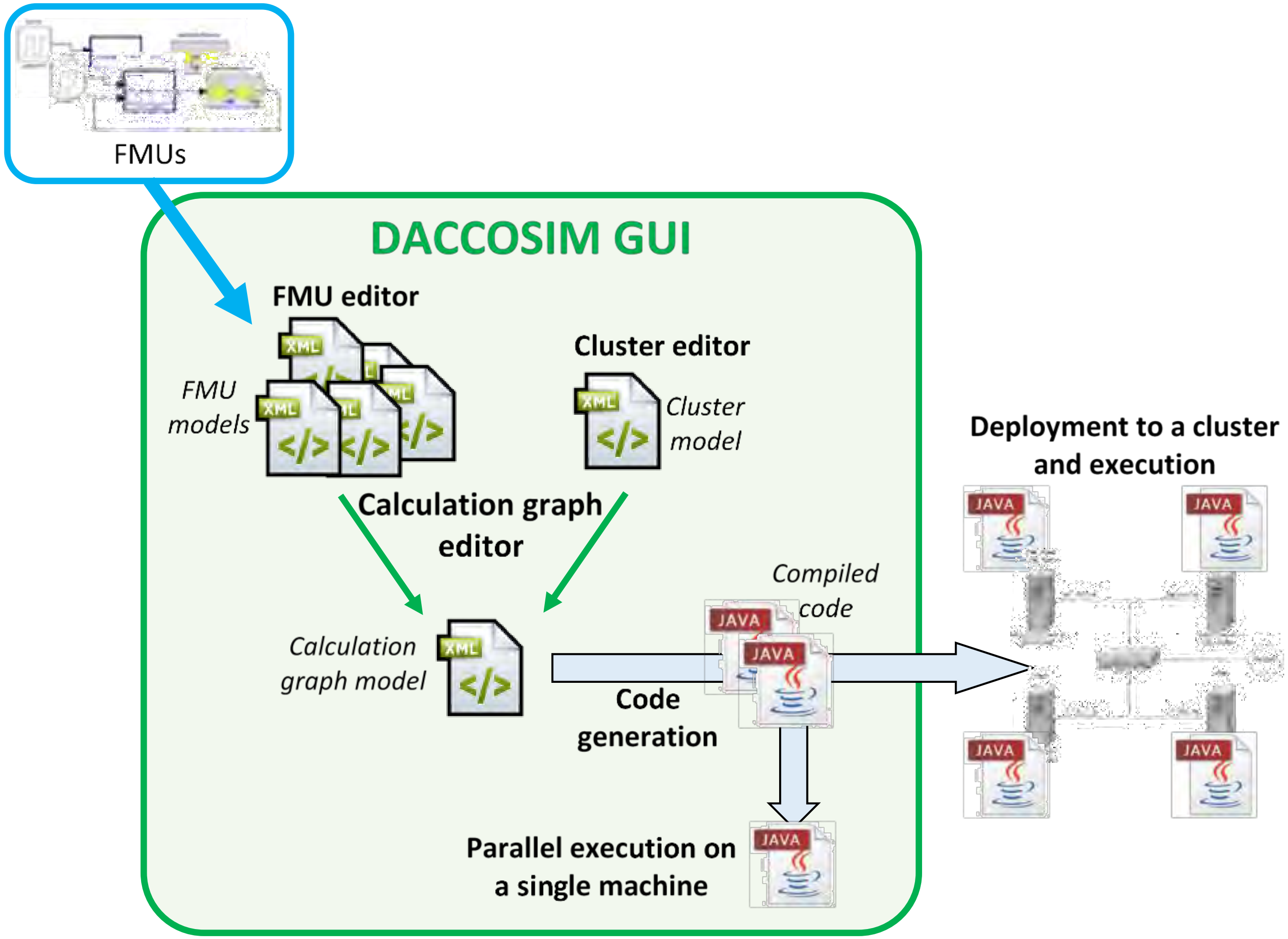


DACCOSIM master: key features

- **Co-initialization**
  - ***Automatic computation of the causality graph***
    - **Algebraic loops are detected using a Tarjan algorithm**
    - **Transformation of the causality graph into an acyclic oriented graph**
  - ***Dissociate treatment of the algebraic chains and parallelized loops***

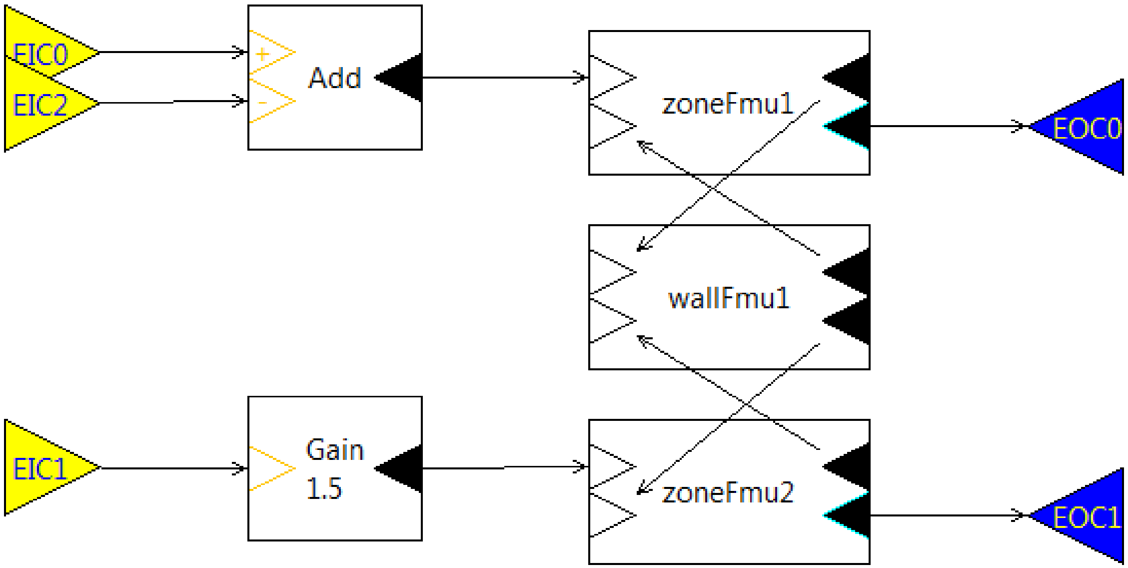


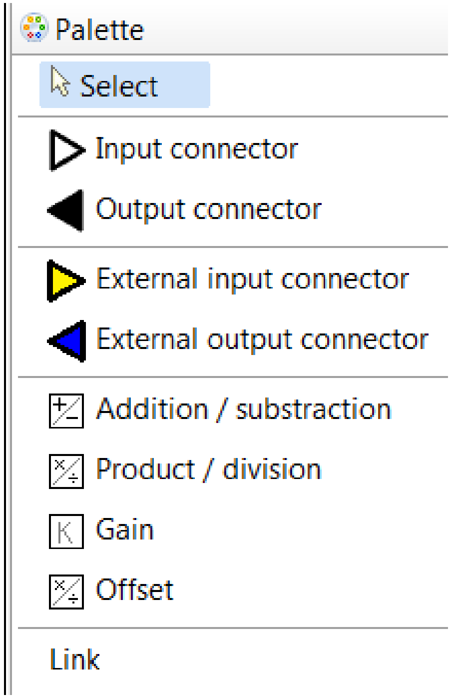
DACCOSIM GUI:
Making multi-simulation easy



Main GUI features

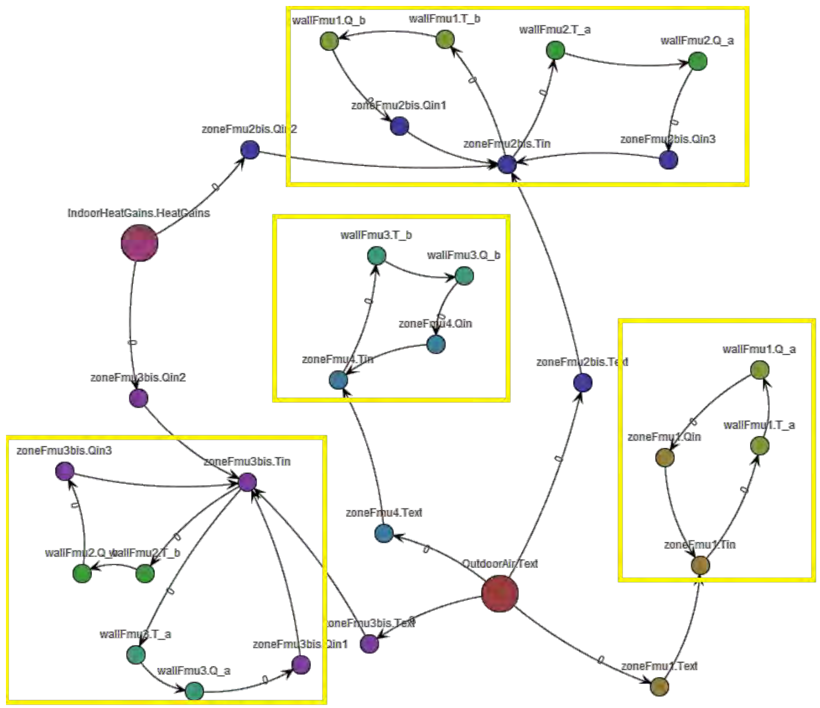
- **Easy creation, configuration and validation of a calculation graph**
  - *FMU, operator and external connector blocks creation*
  - *Inter-block exchange definition*
  - *Sophisticated Copy-Paste function*
  - *Cluster resource association*





Main GUI features

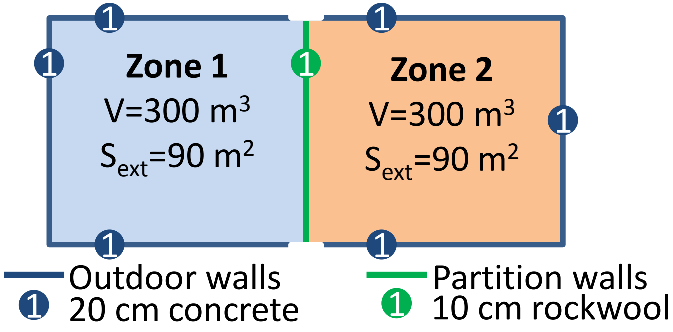
- **Creation and visualization of the inter-block dependencies**
  - *Allows to set consistent system-wide initial values*

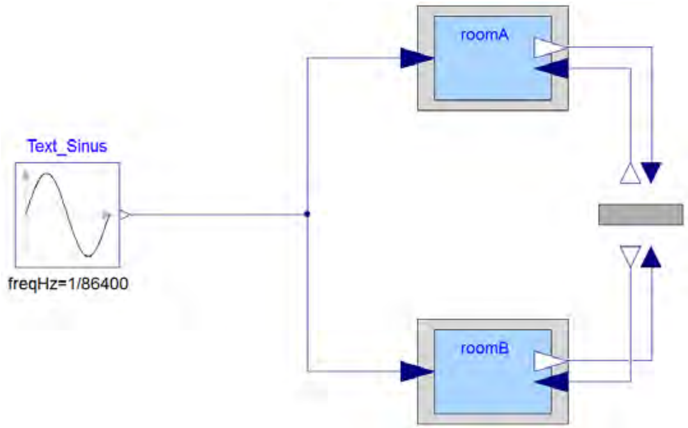


- **Direct generation of the DACCOSIM global and local master codes**
- **Direct execution of a local co-simulation**

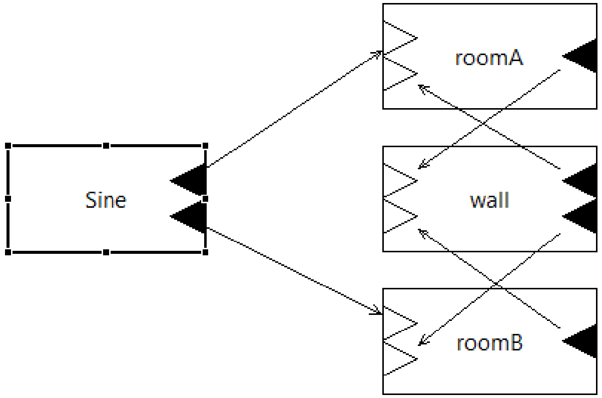
Use Case EnerBat 1 (2014)

- **Simple thermal use-case with continuous-time model**
  - ***Based on Modelica Standard Library, with the HeatTransfer package***
  - ***Two separate thermal zones (rooms) with a wall in between***
  - ***Boundary conditions***
    - *Outdoor temperature described by sine signal*





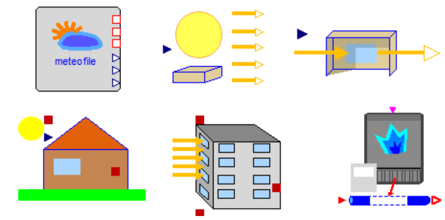
Reference model in Dymola without FMUs

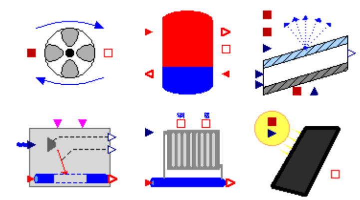


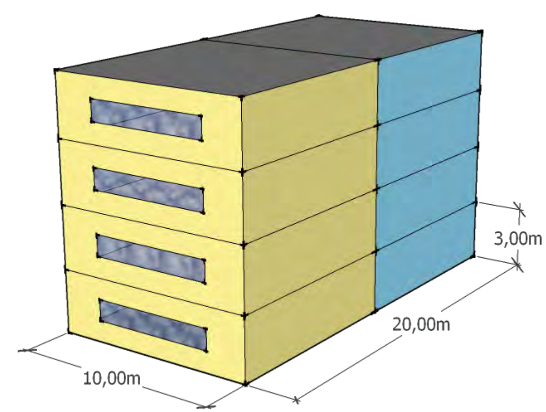
Equivalent model in DACCOSIM

Use Case EnerBat 3 (2015)

- **A realistic test case**
  - *Based on BuildSysPro, EDF Modelica library for modeling energy in buildings*
  - *Multi-layer wall defined by encapsulated arrays of records with changeable size*
  - *Modelling of a residential multi-story building*
  - *Physical phenomena*
    - *Conduction*
    - *Convection*
    - *Long wave radiation (linearized)*
    - *Solar radiation*
  - *Boundary conditions*
    - *Meteorological data*
    - *Specific boundary condition for the crawl space and the top floor*





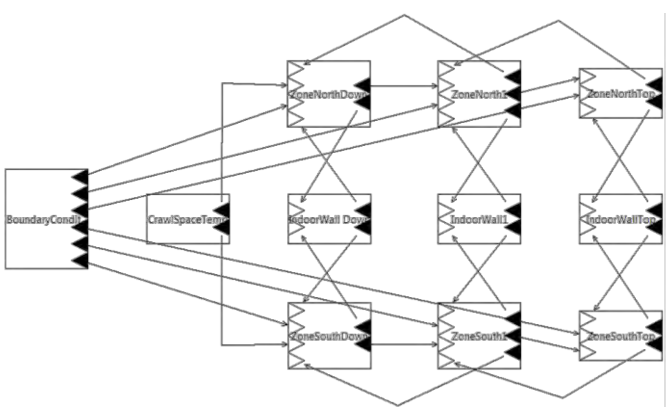


Use Case EnerBat 3 (2015)

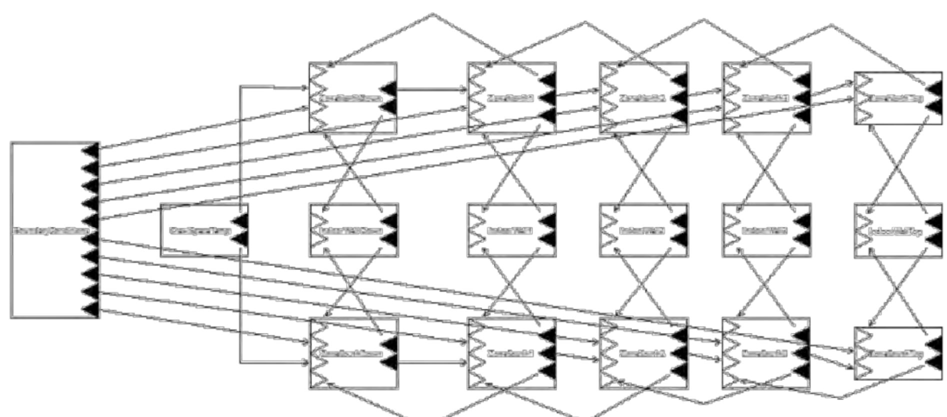
- ***Multi-scale thermal use case***
  - *Variable number of FMUs can be assembled*
  - *5 levels of complexity (size/weight) for each FMU*
- ***Model to FMU orientation***
  - *Acausal heat port to causal interfaces*
    - *Resistive node → input **T** and output **Qflow***
    - *Capacitive node → output **T** and input **Qflow***

	Variable number	
	Walls	Zones
Build 1	5	482
Build 2	10	582
Build 3	25	882
Build 4	50	1382
Build 5	500	10382

3-stories building (11 FMUs)



5-stories building (17 FMUs)

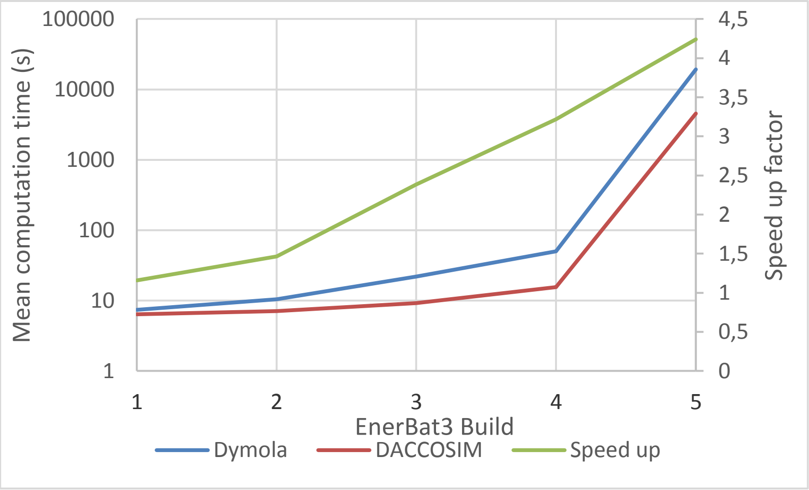


DACCOSIM performance

EnerBat 1	Mean Computation Time (s)		
	Dymola	DACCOSIM	Speed up
Build 1	3.81	2.52	1.51

Dymola (monothreaded co-simulation)
DACCOSIM (multithreaded co-simulation)

EnerBat 3	Mean Computation Time (s)		
	Dymola	DACCOSIM	Speed up
Build 1	7.4	6.39	1.158
Build 2	10.4	7.11	1.463
Build 3	21.93	9.19	2.386
Build 4	50.13	15.58	3.218
Build 5	19230.33	4535.17	4.240



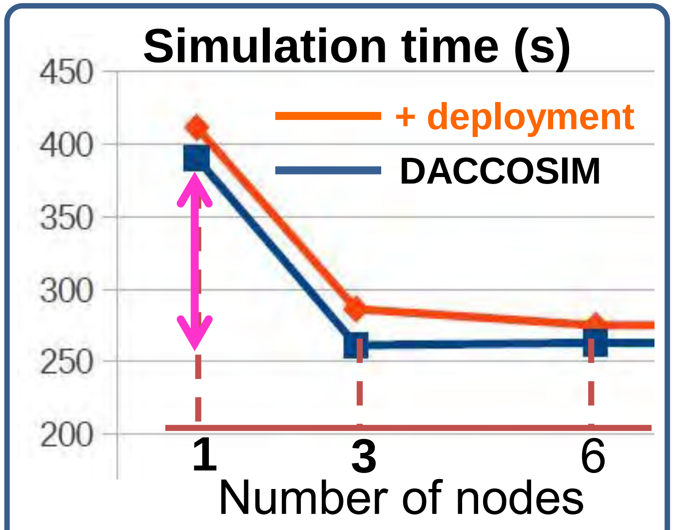
Provided measurements are the average of 10 experiments
Processor: Intel Core i7-3840QM CPU @ 2.80 GHz, 4 physical cores, 8 logical cores
OS: Windows 7 64 bits
RAM: 32.0 GB DDR3 932 MHz

DACCOSIM performance on PC cluster

DACCOSIM on a cluster of 6-core nodes (distributed & multithreaded co-simulation)

Enerbat-3 L5_05 : 11 FMUs
Constant step size: 50 s
Simulated time: 5 h
Simulation time: 400 s → 250 s (3 nodes)

Performance analysis ongoing:
- communication/computation time ?
- load balancing/FMU granularity ?
→ optimization & speedup improvement



Cluster of 6C/12Th nodes
Speedup(3) = 1.5

Next: « 100 heavy FMUs » scalability benchmark on PC cluster

Provided measurements are the average of 3 experiments
Cluster Node: 1x Intel Core i7 CPU920 @ 2.67 GHz, 6 physical cores, 12 logical cores
Interconnect: Ethernet 10 Gbit/s
OS: Linux 64 bits
RAM: 6.0 GB

What's next?

- DACCOSIM as a super FMU (recursive mode)
  - *Final development planned early 2016*
- Further optimizations of:
  - *Inter-block exchanges*
  - *Model splitting and smart FMU deployment*
  - *Bisection search algorithm for event catch*
- Update of the C++ version with a new C++ FMI API
  - *Sezzet developed by SIANI as a clone of JavaFMI*
- MPI implementation for PC clusters
  - *mpiJava version early 2016, MPI C/C++ version further*
- FMI 2.1 support (when published, not planned in 2016)

Demos

GUI Demo **Cluster Demo** **Real time visu**

http://daccosim.foundry.supelec.fr/