Multi-Agent Multi-Model Simulation of Smart Grids in the MS4SG Project

J. Vaubourg, Y. Presse, B. Camus, C. Bourjot, L. Ciarletta,
V. Chevrier, J.-P. Tavella and H. Morais

PAAMS
3rd June 2015

Multi-Agent Multi-Model Simulation of Smart Grids in the MS4SG Project

Multi-Agent Multi-Model Simulation of Smart Grids in the MS4SG Project

J. Vaubourg, Y. Presse, B. Camus, C. Bourjot, L. Ciarletta, V. Chevrier, J.-P. Tavella and H. Morais

PAAMS
3rd June 2015

- We will present you our work about modeling and simulation for smart grids.
- This work is part of the MS4SG project (*Multi-Simulation For Smart Grids*).
- The MS4SG project is a joint project between researchers from Inria, Loria and the University of Lorraine, and industrials from EDF, the French electricity utility.

Plan

1. **Smart Grids**
2. **Simulation Goals & Challenges**
3. **MECSYCO: Co-Simulation Platform**
4. **Co-Simulation Building Example**

Multi-Agent Multi-Model Simulation of Smart Grids in the MS4SG Project

Plan

- We will begin this presentation with a short overview of the smart grids, and its goals for the future.
- Then we will speak about the goals and the challenges with smart grids simulations, illustrating the issues thanks to a real use case.
- Then we will introduce the MECSYCO co-simulation platform, our solution for meeting these challenges.
- Finally, we will propose you a co-simulation building example, for better understanding the main concepts of MECSYCO.

Smart Grids

What? Why?

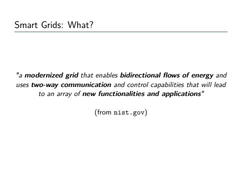
Smart Grids: What?

*"a **modernized grid** that enables **bidirectional flows of energy** and uses **two-way communication** and control capabilities that will lead to an array of **new functionalities and applications**"*

(from `nist.gov`)

Multi-Agent Multi-Model Simulation of Smart Grids in the MS4SG Project

└Smart Grids: What?



- According to the American *National Institute of Standards and Technology* (NIST), a smart grid is a modernized power grid, with bidirectional flows of energy and a two-way communication, leading to new functionalities and applications.

Smart Grids: Why?

- **Renewable and intermittent** energy sources
- **Production & Consumption balance**
- **Voltage control**
- **Electric vehicles** increase

New algorithms or original operating modes to test

Smart Grids: Why?

- **Renewable and intermittent** energy sources
- **Production & Consumption balance**
- **Voltage control**
- **Electric vehicles** increase

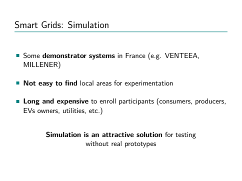
New algorithms or original operating modes to test

- More concretely, why?
- For integrating renewable and intermittent energies, like solar photovoltaic or wind energies.
- For smoothing comsumption peaks, with a better Production & Comsumption balance.
- For better controlling the stability of the voltage on the power grid, for example by monitoring the received voltage rather than the sent voltage.
- And after all, for anticipating the electric vehicles increase, that will result in significant fluctuations on the power grid. For example, in France, the French government plans to introduce 1 million of electric vehicles by 2020.
- Currently, smart grids are not widely deployed but researchers and industrials need to test these new algorithms and operating modes, on this new type of grid.

Smart Grids: Simulation

- Some **demonstrator systems** in France (e.g. VENTEEA, MILLENER)
- **Not easy to find** local areas for experimentation
- **Long and expensive** to enroll participants (consumers, producers, EVs owners, utilities, etc.)

Simulation is an attractive solution for testing without real prototypes



- For these purposes, some real (and physical) demonstrators exist in France and in the rest of the world.
- But it is very hard to find new areas for building new demonstrators.
- And, above all, it is a very long and expensive process to enroll all the required participants.
- For these reasons, smart grids simulation is an attractive solution for testing without requiring real prototypes.

Simulation
Goals & Challenges

in the context of Smart grids

Fields of technology

Power Grid

Communication Network

Information System

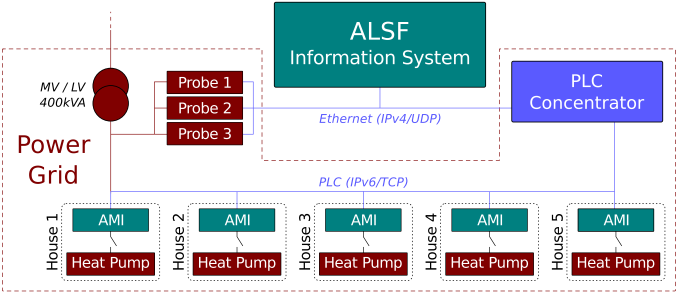
Smart grids are composed of
3 main fields of technology

Fields of technology

Smart grids are composed of
3 main fields of technology

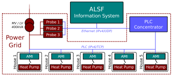
- So, we need smart grids simulations.
- A smart grid is composed of 3 main fields of technology.
- A power grid of course, but also communication networks and information systems.

Use Case Illustration



Concept Grid: Real use case provided by EDF (French electricity utility)

Use Case Illustration

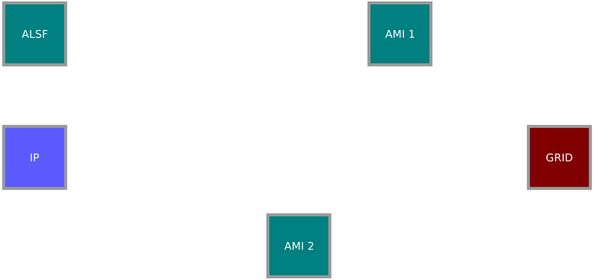


Concept Grid: Real use case provided by EDF (French electricity utility)

- We will find these 3 different fields on the real use case on which we will base the rest of this presentation.
- This use case, named *Concept Grid*, was provided by EDF, the French electricity utility.
- The system to simulate contains a power grid with a power generator, and five consuming houses.
- We want to distribute the generated power between the five houses, asking them to consume one by one, at specified times, during the day.
- For that, we use an information system corresponding to an *Advanced Load Shedding Function* (ALSF).
- We need a communication netwoork (in this case, an IP network), for enabling the information system to send orders (time slots) to the houses.
- Each house is equipped with a smart smeter, for executing the information system instructions.
- And, heat pumps will represent the power consumption, and will be enabled or disabled during the day, depending on the information system orders.
- Finally, we added 3 probes, in charge to provide some feedbacks about the current current & voltage on the power grid, to the information system.
- This is the system we want to model and simulate.

Goals

Goals: Reusability Support



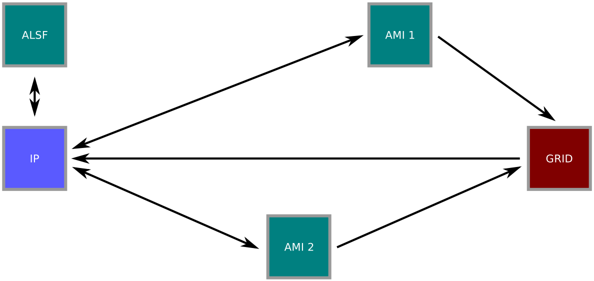
Various existing **well-tested, proven or industrial models** has to be reused

Goals: Reusability Support

Various existing **well-tested, proven or industrial models** has to be reused

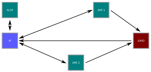
- EDF provided us the power grid model, representing the whole electrical network topology, and the smart meter models, representing their application layer.
- We developed the information system model and the communication model, representing the whole IP network topology.
- More generally, various well-tested, proven or industrial models already exist and we need to reuse them, rather than rewriting all the required models in a smart grid simulation.
- So, our first goal is the model reusability.

Goals: Interactions among models



Models have to be **connected and executed together** in a same **co-simulation**

Goals: Interactions among models

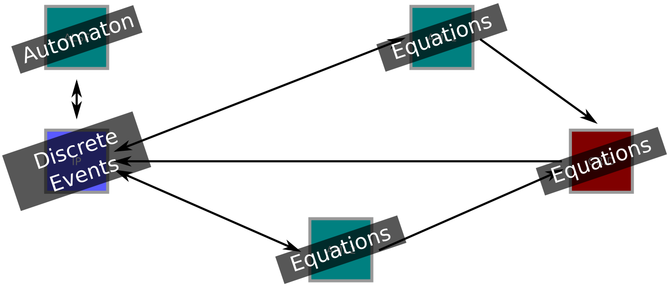


Models have to be **connected and executed together** in a same **co-simulation**

- Secondly, for simulating the complete smart grid, we need to connect all of these models together and execute them in a same co-simulation.

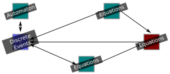
Challenges

Challenges: Formalisms Heterogeneity Integration



Different formalisms have to co-exist

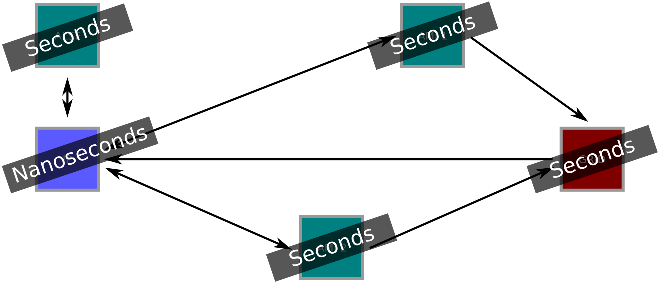
Challenges: Formalisms Heterogeneity Integration



Different formalisms have to co-exist

- These goals led us to some challenges.
- The first one is the formalisms heterogeneity integration.
- Because the models are from different fields of technology, they correspond to different areas of expertise.
- This is why we have to integrate different formalisms in a same co-simulation: here, equation-based models, a discrete-events-based model and an automaton.

Challenges: Representations Heterogeneity Integration

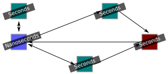


Simulated time and exchanged data can have
different representations

Multi-Agent Multi-Model Simulation of Smart Grids in the MS4SG Project

Challenges: Representations Heterogeneity Integration

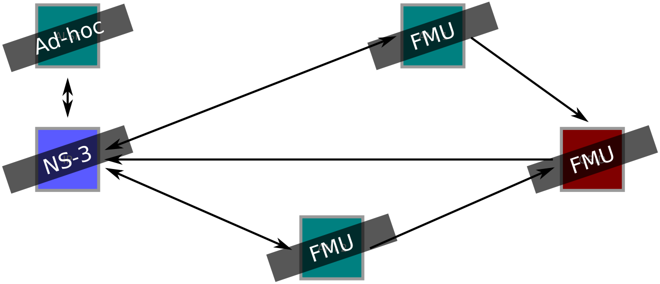
Challenges: Representations Heterogeneity Integration



Simulated time and exchanged data can have **different representations**

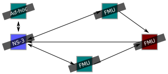
- The second challenge is the representations heterogeneity integration.
- Indeed, models can work with different time scales or with different and inconsistent data representations.

Challenges: Simulators Heterogeneity Integration



Models implementations are usable with **different simulators**
(often **not interoperable** together)

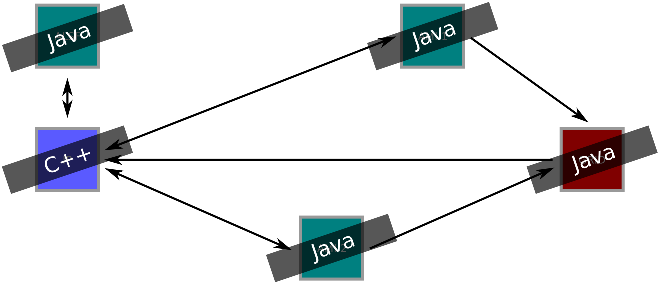
Challenges: Simulators Heterogeneity Integration



Models implementations are usable with **different simulators** (often **not interoperable** together)

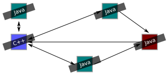
- Then, the model implementations are often available for a specific simulation software (simulator), and these ones are often not interoperable together.
- FMU stands for *Functional Mockup Unit* (from the *Functional Mockup Interface* industrial standard) and NS-3 is a popular IP network simulator.
- So, we need to integrate the simulators heterogeneity in the co-simulation.

Challenges: Languages Heterogeneity Integration



Simulator bindings are proposed with **different languages**

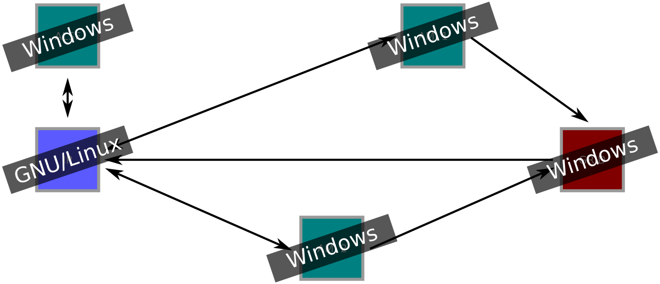
Challenges: Languages Heterogeneity Integration



Simulator bindings are proposed with **different languages**

- In the same manner, bindings with these simulators are possible using different languages.
- Thus, we also need to handle the languages heterogeneity.

Challenges: Platforms Heterogeneity Integration

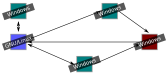


Simulators are available for **different platforms**

Multi-Agent Multi-Model Simulation of Smart Grids in the MS4SG Project

Challenges: Platforms Heterogeneity Integration

Challenges: Platforms Heterogeneity Integration



Simulators are available for **different platforms**

- Finally, the simulation softwares are often not available for the same operating system.
- So, the platforms heterogeneity has to be integrate.

Challenges: Synthesis

Managing the **heterogeneity** of a multi-model:

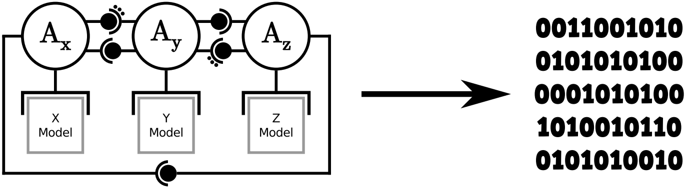
1. **Models Issues**: integrating different **formalisms and representations**
2. **Software Issues**: ensuring the **simulators interoperability** for the models reusability

MECSYCO

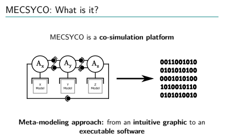
Multi-agent Environment for Complex SYstems CO-simulation

MECSYCO: What is it?

MECSYCO is a **co-simulation platform**



Meta-modeling approach: from an **intuitive graphic** to an **executable software**

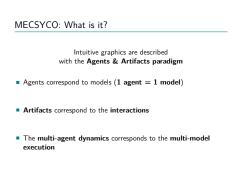


- MECSYCO is our simulation for meeting these challenges. MECSYCO stands for *Multi-agent Environment for Complex SYstems CO-simulation*.
- MECSYCO is a co-simulation platform based on a meta-modeling approach.
- From an intuitive graphic, we are able to produce an executable software, corresponding to the co-simulation to execute, producing some simulation results.

MECSYCO: What is it?

Intuitive graphics are described
with the **Agents & Artifacts paradigm**

- Agents correspond to models (**1 agent = 1 model**)
- **Artifacts** correspond to the **interactions**
- The **multi-agent dynamics** corresponds to the **multi-model execution**



- MECSYCO intuitive graphics are described with the *Agents & Artifacts* paradigm, a multi-agent paradigm.
- In MECSYCO, agents are autonomous, and corresponds to models (we have 1 agent for 1 model).
- Artifacts, usable as tools for the agents, correspond to the interactions, between the models and the agents and between the agents themselves.
- Finally, the multi-agent dynamics correspond to the multi-model execution.

MECSYCO: What is it?

Specifications are based on **the DEVS formalism**

- **Any formalism can be mapped** in DEVS
- Agents' behavior is defined with the **Chandy-Misra-Bryant algorithm**

MECSYCO: What is it?

Specifications are based on **the DEVS formalism**

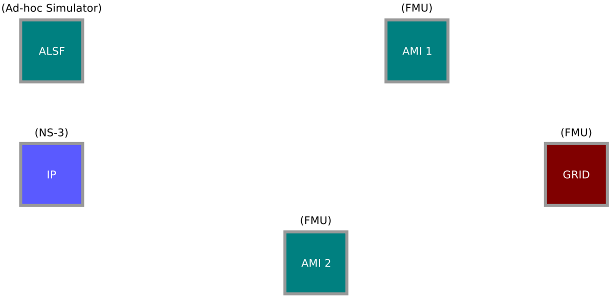
- **Any formalism can be mapped** in DEVS
- Agents' behavior is defined with the **Chandy-Misra-Bryant algorithm**

- MECSYCO specifications are based on the DEVS formalism.
- It is proven that DEVS is able to integrate any other formalisms.
- We defined the agents' behavior thanks to the *Chandy-Misra-Bryant* algorithm, a DEVS-compliant and decentralized algorithm for the global time synchronization.

Co-Simulation Building Example

With MECSYCO

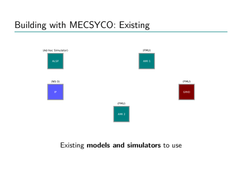
Building with MECSYCO: Existing



Existing **models and simulators** to use

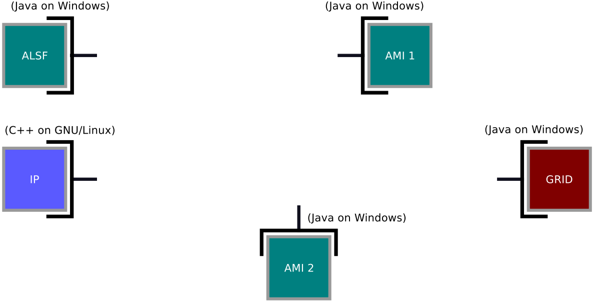
Multi-Agent Multi-Model Simulation of Smart Grids in the MS4SG Project

└─Building with MECSYCO: Existing



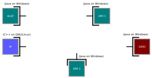
- Back to our use case: we have different models, with different formalisms, time scales, for different simulation softwares, operating systems, and so on.
- We want to build a co-simulation with these multi-heterogeneous models, for simulating the system introduced before.

Building with MECSYCO: Model Artifacts



Model Artifacts: ensure software **interoperability** and manage **formalism** integration

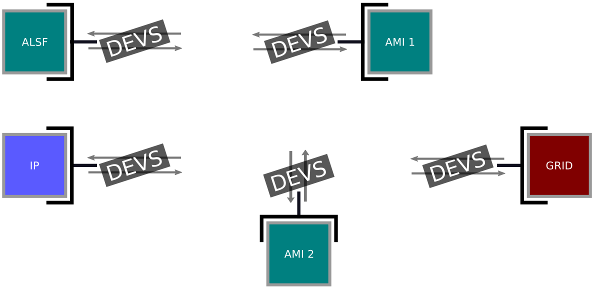
Building with MECSYCO: Model Artifacts



Model Artifacts: ensure software **interoperability** and manage **formalism** integration

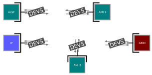
- We solved the software interoperability and the formalism issues thanks to the MECSYCO model artifacts.
- A model artifact is just a piece of code, developed for each simulation software.
- So, once a model artifact was developed for a simulation software, all of its models can be integrated to a MECSYCO co-simulation.

Building with MECSYCO: Model Artifacts



Model Artifacts act as **DEVS** wrappers

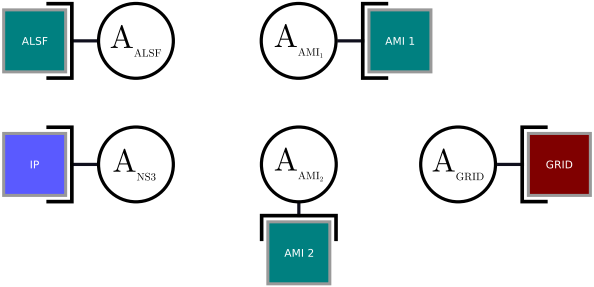
Building with MECSYCO: Model Artifacts



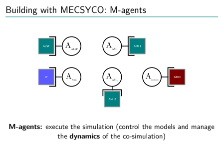
Model Artifacts act as **DEVS** wrappers

- In fact, model artifacts act as DEVS wrappers, for integrating different formalisms.
- The piece of code to develop for creating a new model artifact contains 5 functions to implement, corresponding to the *DEVS Simulation Protocol* specifications.

Building with MECSYCO: M-agents

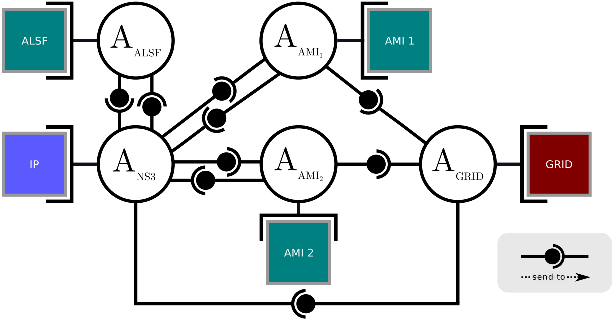


M-agents: execute the simulation (control the models and manage the **dynamics** of the co-simulation)



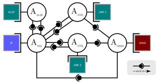
- Our agents - we named M-agents - are in charge to execute the simulation, controlling the models and managing the dynamics of the co-simulation.
- They are also in charge to ensure the global time synchronization.

Building with MECSYCO: Coupling Artifacts



Coupling Artifacts: exchange events between m-agents

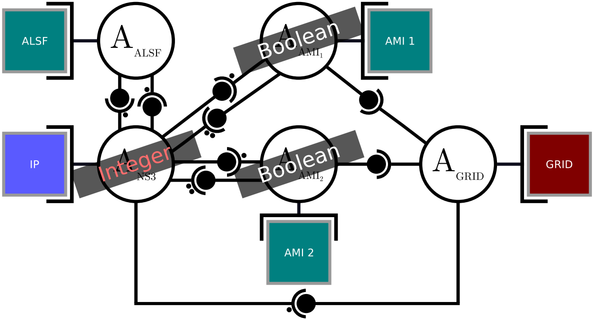
Building with MECSYCO: Coupling Artifacts



Coupling Artifacts: exchange events between m-agents

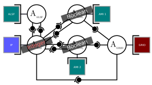
- Finally, for connecting the agents together, we use MECSYCO coupling artifacts, acting as mailboxes for the exchanged events.
- These coupling artifacts are tools for the agents: a m-agent can send events to another m-agent without be worried about the data or the time representation, and without have to ask if the remote m-agent is ready to integrate it.

Building with MECSYCO: Operations



Operations: transform the events
(in this case, convert types)

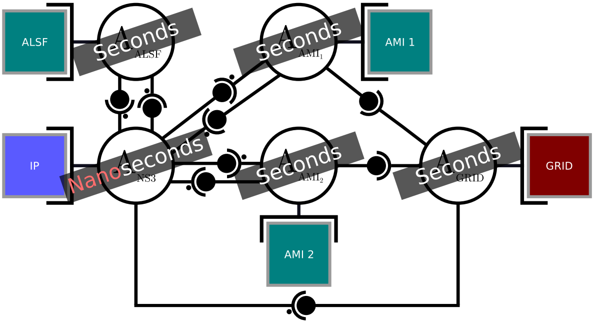
Building with MECSYCO: Operations



Operations: transform the events
(in this case, convert types)

- These coupling artifacts support operations, able to convert the data inside the events...

Building with MECSYCO: Operations



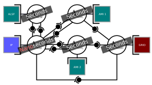
Operations: in this case, change **time scales**

2015-06-15

Multi-Agent Multi-Model Simulation of Smart Grids in the MS4SG Project

└─Building with MECSYCO: Operations

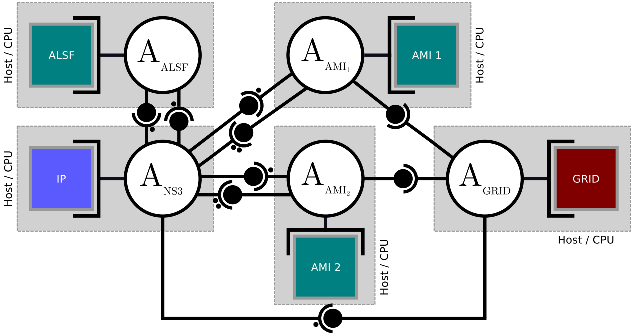
Building with MECSYCO: Operations



Operations: in this case, change **time scales**

- … or convert the time associated to the events from a m-agent to another, solving the representation issues.

Building with MECSYCO: Remote Communications

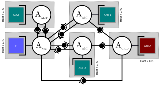


Each simulator instance can use a **dedicated thread or host**

Multi-Agent Multi-Model Simulation of Smart Grids in the MS4SG Project

└Building with MECSYCO: Remote Communications

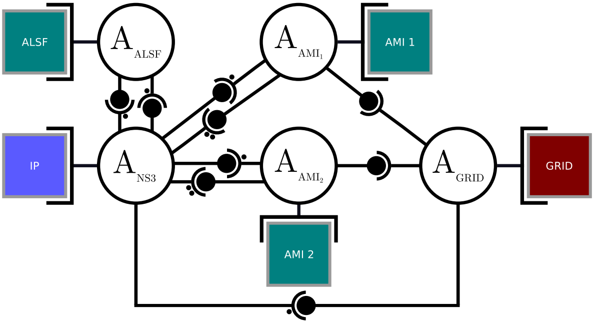
Building with MECSYCO: Remote Communications



Each simulator instance can use a **dedicated thread or host**

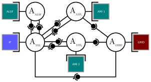
- Because we use a Chandy-Misra-Bryant algorithm for synchronizing the global time, and because we implemented the coupling artifacts with communication middleware software (as DDS or ZeroMQ), a MECSYCO co-simulation can be entirely decentralized.
- So, each simulator instance can run on a different thread or computer.

Building with MECSYCO: Intuitive Graphic



This is the **intuitive graphic** corresponding to our use case

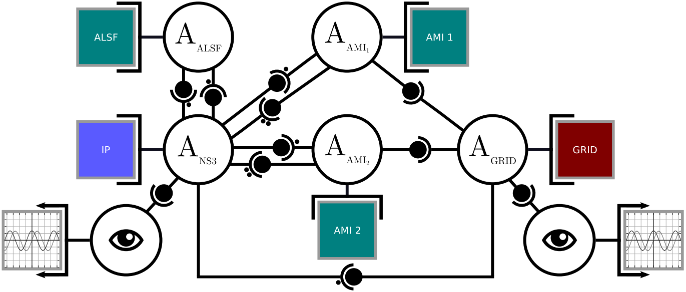
Building with MECSYCO: Intuitive Graphic



This is the **intuitive graphic** corresponding to our use case

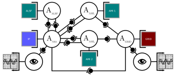
- This is the intuitive graphic, corresponding to our use case, and enabled us to directly produce an executable co-simulation.

Building with MECSYCO: Observing Agents



The Agents & Artifacts paradigm allows us to add **observing m-agents** (plots, traces, etc.)

Building with MECSYCO: Observing Agents



The Agents & Artifacts paradigm allows us to add **observing m-agents** (plots, traces, etc.)

- Because the Agents & Artifacts paradigm is flexible and expressivness enough, we can add other kinds of m-agent.
- For example, we added observing m-agents, in charge to receive the simulation results from the other m-agent, and exploit them for realtime or post-mortem visualizations.

Building with MECSYCO: Results Visualization

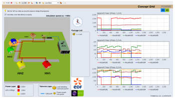
The *European Institute For Energy Research* (EIFER) used our simulation results for building a **visualization software**

2015-06-15

Multi-Agent Multi-Model Simulation of Smart Grids in the MS4SG Project

└─Building with MECSYCO: Results Visualization

Building with MECSYCO: Results Visualization



The *European Institute For Energy Research* (EIFER) used our simulation results for building a **visualization software**

- Thanks to these observing m-agents, the *European Institute For Energy Research* (EIFER) built a powerful and dedicated visualization software, corresponding to this screenshot.
- The simulation results of this use case were validated by EDF R&D, by comparing them to a real demonstrator.

Conclusion

- **Executable co-simulation** created from a set of models, thanks to an **intuitive graphic**
- Integration of **several forms of heterogeneity** (formalism, representation, language, simulator and platform)
- **Simulation results** are directly usable

Conclusion

- Purely **decentralized** execution
- Developed in **Java and C++**
- Developement **framework** available (`mecsyco.com`)
  Free Software: AGPL 2.0

Perspectives

1. **Generating the physical domain** from CIM (*Common Information Model*)

2. Connecting more business tools to **visualize simulations**

3. Long Term: **Experimental Plans**

Questions?