Co-Simulation of IP Network Models in the Cyber-Physical Systems Context, using a DEVS-based Platform

Julien Vaubourg, Vincent Chevrier
Laurent Ciarletta, Benjamin Camus

SpringSim'16 Pasadena
Communications and Networking Symposium
April 4th, 2016

Cyber-Physical Systems (CPS)

1. **Communication (IP) networks**, with various technologies (Ethernet, PLC, GPRS, etc.)

2. Everyday **physical devices** connected to information systems, through these IP networks (heatings, fridges, etc.)

3. Connected devices providing **input and output ports** (remote control commands, monitoring data, etc.)

Issues

We are interested in **CPS simulation**, but:

1. **No big CPS simulator** able to model:
   - □ IP networks with all existing technologies
   - □ whole diversity of existing everyday devices
   - □ all kinds of information systems

2. Rewriting all models available for different simulators is **too long, too expensive and requires too many different skills**

Challenges

1. Modeling CPS **using already existing models**, currently available for different incompatible simulators

2. Connecting these models together in a same a **multi-model**, integrating **all different forms of heterogeneity**

Solution

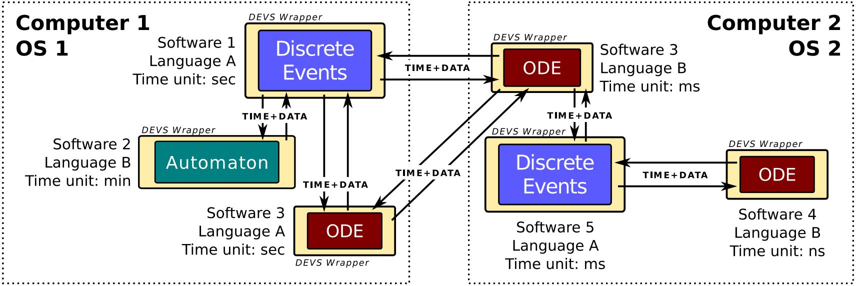
Co-simulation is the retained solution:

- Every **required simulators are executed independently**, with communications between each other (time synchronization, data exchanges)
- The **MECSYCO co-simulation platform** was chosen

MECSYCO

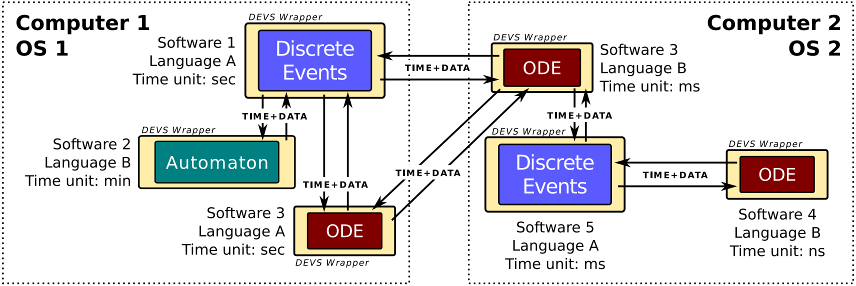
- **Co-simulation platform based on the DEVS formalism** [Ziegler et al., 2000] and *theoretically* able to integrate all other formalisms
- Simulators executing models are **connected to DEVS wrappers**
- Integrating a new simulator into MECSYCO **means creating a new DEVS wrapper**
- Time synchronization is ensured using **CMB algorithm** [Chandy et al., 1979] (fully decentralized execution)

MECSYCO



MECSYCO enables us to **meet our heterogeneity challenges**

MECSYCO



MECSYCO enables us to **meet our heterogeneity challenges** but does **not provide DEVS wrappers for IP simulators**

Contributions

DEVS wrapping IP simulators in order to use them in a DEVS co-simulation, requires to be able to:

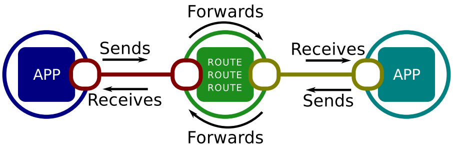
1. **Describe an IP topology (+ apps) with a multi-model**, equivalent to the same description using only one model

   $\Longrightarrow$ add **DEVS ports** to models

2. **Control the execution** (event processing) of an IP model during the simulation, with external event exchanges

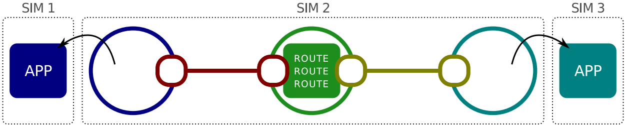
   $\Longrightarrow$ add **DEVS Simulation Protocol** functions to simulators

Adding DEVS ports



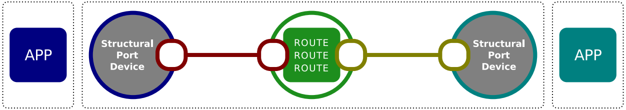
- **IP Network Model:** model composed of **submodels** representing nodes, applications, links, etc.
- **End-Node:** node with applications, producing and receiving data (e.g. client, server)
- **Transit-Node:** node without application, forwarding data from and to other nodes (e.g. router, switch)

Adding DEVS ports



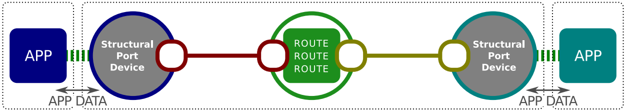
- **Application submodels** (data processing) offered by IP simulators are **restricted**
- Exporting of the **application submodels** for **using external specialized models** (e.g. information system models)
- 3 models with **3 simulators** are now used

Adding DEVS ports



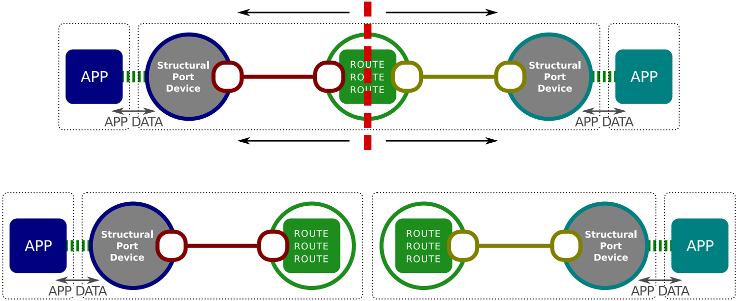
- **Reusing of the empty nodes** for creating Structural Port Devices
- **Structural Port Device:** application installed on an End-Node, for catching internal application layer data and injecting external application layer data

Adding DEVS ports



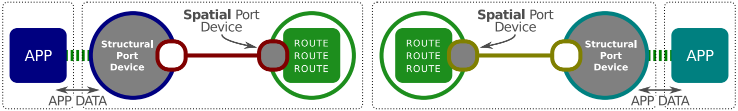
- Port Devices can be used as **DEVS Ports** for creating Structural Couplings between the IP model and others
- **Structural Coupling:** interconnection between an IP model (representing the *network layer*) and another model (representing the *application layer*)

Adding DEVS ports



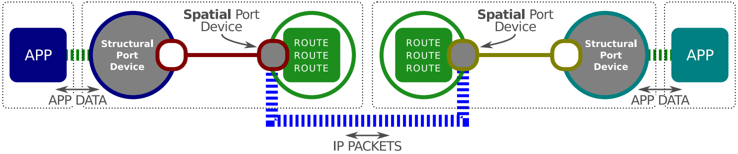
- No IP simulator offers submodels for **all existing IP technologies**
- **Splitting the IP model** into 2 separate models for being able to use 2 different simulators

Adding DEVS ports



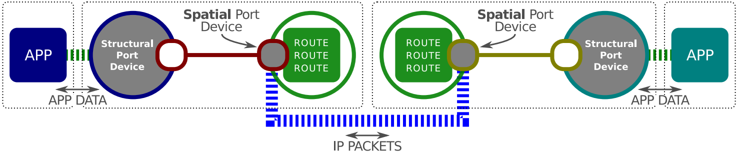
- **Using of the network device submodels** for creating Spatial Port Devices
- **Spatial Port Device:** modeled device (e.g. network device) used for catching internal IP packets and for injecting external IP packets

Adding DEVS ports



- New Port Devices can be used as **DEVS Ports** for creating Spatial Couplings between the IP models
- **Spatial Coupling:** interconnection between two IP models, each one representing a part of an IP network

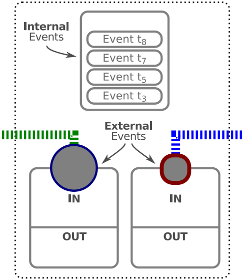
Adding DEVS ports



- New Port Devices can be used as **DEVS Ports** for creating Spatial Couplings between the IP models
- **Spatial Coupling:** interconnection between two IP models, each one representing a part of an IP network

$\Longrightarrow$ We are now able **to describe an IP topology with apps**, using a multi-model

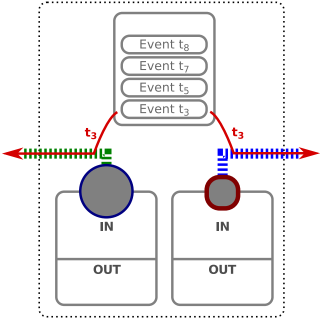
Adding DEVS Simulation Protocol functions



IP simulators generally work with an **internal event stack**

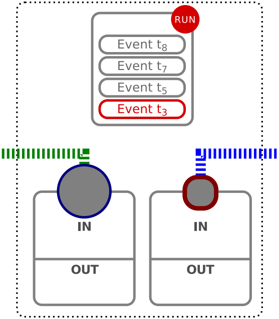
- Internal **event loop has to be controlled** (RUN vs. PAUSED)
- Port Devices have to work with **dedicated external event stacks**

Adding DEVS Simulation Protocol functions



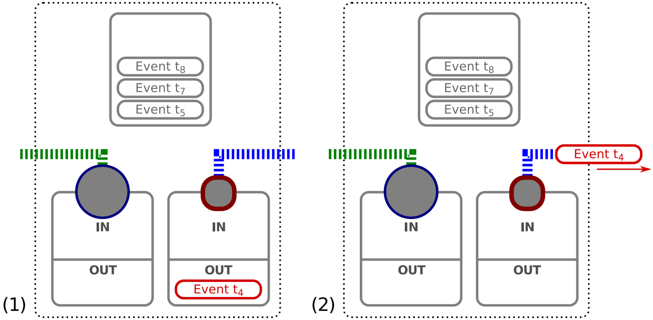
1/4 **Getting next internal event time**: consulting the event stack

Adding DEVS Simulation Protocol functions



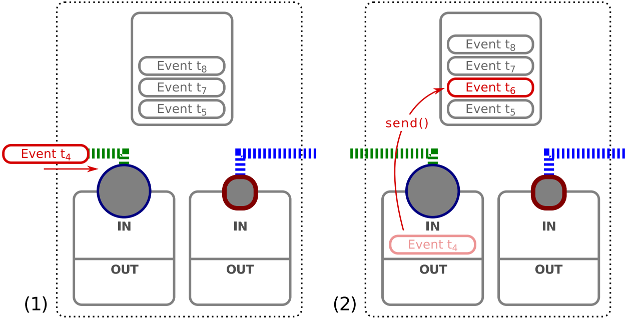
2/4 **Processing an internal event:** executing the main event loop

Adding DEVS Simulation Protocol functions



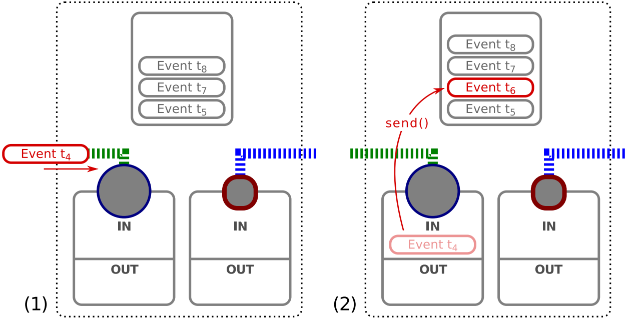
3/4 **Getting an output event:** using Port Devices event stacks

Adding DEVS Simulation Protocol functions



4/4 **Processing an external event:** filling and executing Port Devices event stacks

Adding DEVS Simulation Protocol functions



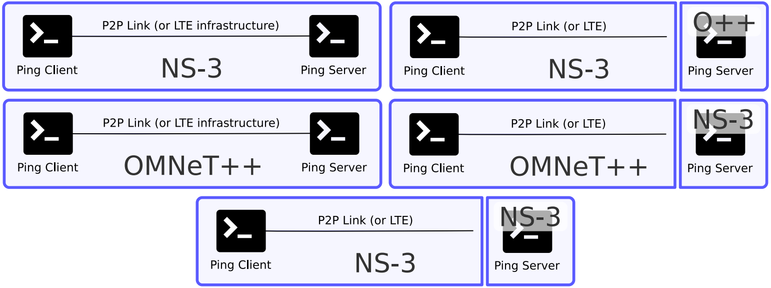
4/4 **Processing an external event:** filling and executing Port Devices event stacks

⟹ We are now able to **integrate IP models and simulators** to DEVS co-simulations

Proof Of Concept

- All **proposed concepts (and others presented in the paper) have been implemented** for NS-3 and OMNeT++/INET
- **DEVS wrappers for NS-3 and OMNeT++/INET** have been added to the MECSYCO library
- **Core source-code of NS-3 and OMNeT++/INET was not changed** (regular software implementations are still usable)

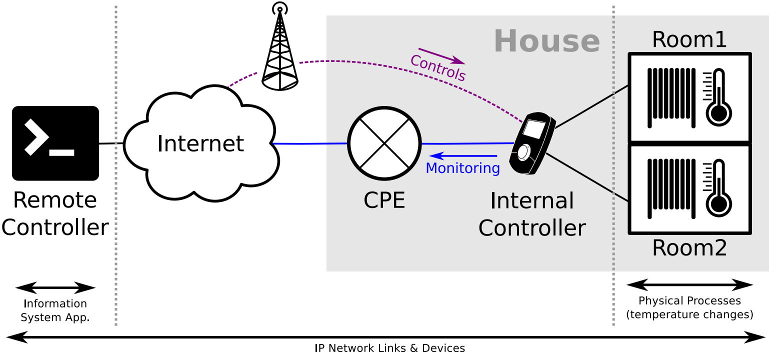
Proof Of Concept



Same simulation results, with or without Spatial Couplings.

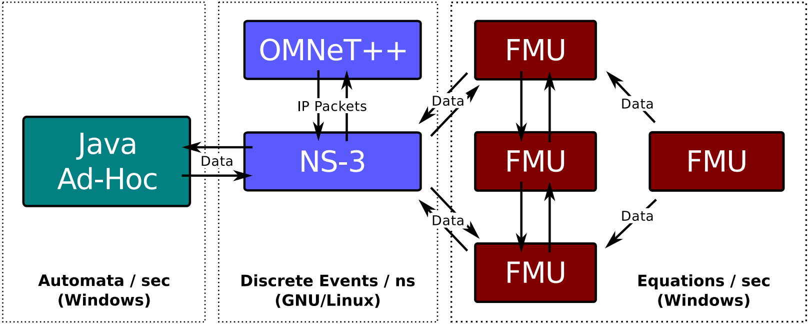
$\Longrightarrow$ **Multi-model versions are equivalent** to single-model ones

Example



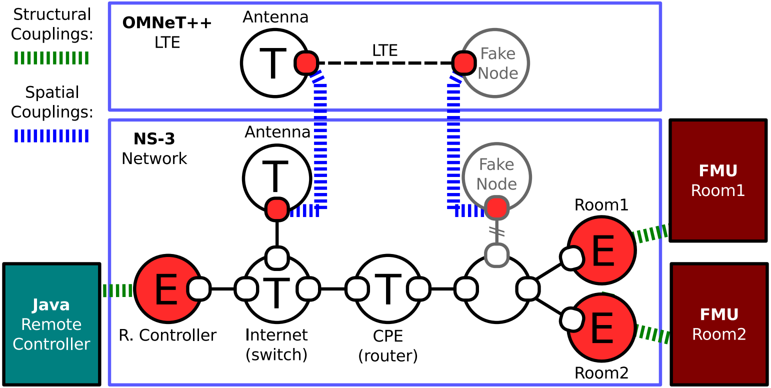
Example: Smart-Heating to model and simulate

Example



Corresponding **multi-model**

Example



IP model with **structural and spatial couplings**

Conclusion

- IP networks models can be **integrated into a DEVS co-simulation**

Conclusion

- IP networks models can be **integrated into a DEVS co-simulation**
- IP models can be **connected to any other models** already integrated to MECSYCO

Conclusion

- IP networks models can be **integrated into a DEVS co-simulation**
- IP models can be **connected to any other models** already integrated to MECSYCO
- Plug-n-play simulator replacements in MECSYCO enables to **compare models performance/accuracy**

Conclusion

- IP networks models can be **integrated into a DEVS co-simulation**
- IP models can be **connected to any other models** already integrated to MECSYCO
- Plug-n-play simulator replacements in MECSYCO enables to **compare models performance/accuracy**
- IP topologies can be modeled **mixing IP models provided by different IP simulators**

Conclusion

- IP networks models can be **integrated into a DEVS co-simulation**
- IP models can be **connected to any other models** already integrated to MECSYCO
- Plug-n-play simulator replacements in MECSYCO enables to **compare models performance/accuracy**
- IP topologies can be modeled **mixing IP models provided by different IP simulators**
- Can be used for **parallelizing an existing sequential IP simulator**

Conclusion

- IP networks models can be **integrated into a DEVS co-simulation**
- IP models can be **connected to any other models** already integrated to MECSYCO
- Plug-n-play simulator replacements in MECSYCO enables to **compare models performance/accuracy**
- IP topologies can be modeled **mixing IP models provided by different IP simulators**
- Can be used for **parallelizing an existing sequential IP simulator**
- Used **by the main French electric utility** company (EDF R&D)

Limitations

Assumptions on the IP simulators:

- Dynamics is based on an **event stack**
- Models **meet the published standards** for network protocols
- Nodes and links are **independently modeled**
- **IP packet is the atomic unit of data** that is simulated

[Riley et al., 2004] proposes some solutions for spatially coupling IP models that use incompatible implementation of a same protocol

Perspectives

1. Integrating **more IP simulators** in order to be able to propose the most possible generic concepts

2. Creating a **model-driven environment** for creating IP multi-models using several IP model libraries, with MECSYCO

Questions?