HAL
open science

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

Julien Vaubourg, Yannick Presse, Benjamin Camus, Christine Bourjot, Laurent Ciarletta, Vincent Chevrier, Jean-Philippe Tavella, Hugo Morais

To cite this version:

Julien Vaubourg, Yannick Presse, Benjamin Camus, Christine Bourjot, Laurent Ciarletta, et al.. Multi-agent Multi-Model Simulation of Smart Grids in the MS4SG Project. PAAMS'15, Jun 2015, Salamanca, Spain. pp.12, 10.1007/978-3-319-18944-4_20 . hal-01171428

HAL Id: hal-01171428

https://inria.hal.science/hal-01171428v1

Submitted on 3 Jul 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

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

Julien Vaubourg[1,3] Yannick Presse[1,3] Benjamin Camus[2,3]
Christine Bourjot[2,3] Laurent Ciarletta[1,3]
Vincent Chevrier[2,3] Jean-Philippe Tavella[4]
Hugo Morais[4]

3rd June, 2015

[1] {Julien.Vaubourg; Yannick.Presse; Laurent.Ciarletta}@inria.fr
Inria, Villers-lès-Nancy, F-54600, France

[2] {Benjamin.Camus; Christine.Bourjot; Vincent.Chevrier}@loria.fr

[3] Université de Lorraine, LORIA, UMR 7503, Vandœuvre-lès-Nancy, F-54506, France / CNRS, LORIA, UMR 7503, Vandœuvre-lès-Nancy, F-54506, France

[4] {Jean-Philippe.Tavella; Hugo.Morais}@edf.fr
EDF - R&D MIRE/R44 1, avenue du Général de Gaulle BP 408 F-92141 Clamart cedex

Abstract

This paper illustrates how the multi-agent approach, or paradigm, can help in the modeling and the simulation of smart grids in the context of MS4SG (a joint project between LORIA-INRIA and EDF R&D). Smart grids simulations need to integrate together pre-existing and heterogeneous models and their simulation software; for example modeling tools of the power grids, of telecommunication networks, and of the information and decision systems. This paper describes the use of MECSYCO as a valid approach to integrate these heterogeneous models in a multi-agent smart grid simulation platform. Several use cases show the ability of MECSYCO to effectively take into account the requirements of smart grids simulation in MS4SG.

Contents

1 Introduction **4**

2 Requirements in Smart-Grids Simulation **5**

3 The MECSYCO Approach **6**

4 Building a Multi-Model with MECSYCO **8**

5 Smart-Grids Multi-Simulation **9**
5.1 Software Integration in MECSYCO : defining model artifacts . . 9
5.2 Single Domain Example 10
5.3 Power Grid and Communication Network 10
5.4 Power Grid, Communication and Information System Domains . 12

6 Discussion **13**

7 Conclusion **13**

1 Introduction

The 2020 Climate and Energy Package of the European Union (20% renewable, 20% energy savings and 20% reduction in emissions of greenhouse gases) has led to the rapid development of production from intermittent energy sources (wind, solar photovoltaic -PV-). This is certainly the phenomenon the most significant of the evolution of electrical systems in the past five years. In France, the installed capacity of the wind farms increased from 3.5 GW in early 2009 to almost 8 GW in late 2013. In the same period, the installed PV capacity has been multiplied by 50 (4.5 GW in 2013). The emergence of renewable energy, but also of new uses of electricity (heat pumps, electric vehicles), the control of peak consumption and the desire to constantly improve the provision of quality led to a necessity to develop more intelligent systems, especially in distribution networks. Such systems are usually called smart grids.

The French DSO (Distribution System Operator) ERDF operates distribution networks having a cumulative total length of 1,250,000 kilometers, mainly in rural areas with overhead power lines. To face these new challenges, ERDF opted several years ago to perform demonstrator systems in some regions of France (e.g. the demonstrator VENTEEA in Aube nearby Troyes which intends to test innovative technologies for the medium voltage grid where Enel Green Power France operates lots of wind farms) and in the French islands (e.g. the demonstrator MILLENER in La Reunion where the lack of interconnection of the grid increases the risk of failure during peak periods). However, despite the advantages of real prototypes, it is not easy to find a local area for experimentation and it is long and expensive to enroll industrial or residential customers. In this sense, the smart grid simulation is an attractive technological solution to test new distributed algorithms (e.g. advanced voltage management) or original operating mode (e.g. islanding) before their use in real prototypes and even in the real networks.

This paper illustrates how the multi-agent paradigm can help in the modeling and the simulation of smart grids in the context of MS4SG (Multi-Simulation for Smart-Grids), a joint project between LORIA-INRIA and EDF R&D. The main goal of the proposed platform is to model real systems, allowing the simulation of different domains of the smart grids at the same time, namely the electrical domain, the information domain and the decision domain. To demonstrate the capacity of the proposed platform to perform simulation with different domains, three case studies are proposed considering three distinct problems. In the first case, a real high voltage network of La Reunion Island is simulated and an agent is used to detect congestion situations. This agent will transmit the congestion information to the decision level which is responsible to find a best solution to solve the problem. The second case is focused in the MV network with two overhead feeders under a fault situation. The main goal is to show the interoperability between the electrical domain and the telecommunication domain under a very stressing scenario, namely a ground phase default. Finally, a third case study will be presented to show the interoperability of the three domains of a smart grid. In this example the system should determine the load shedding

considering two house equipped smart-meter with bi-directional communication systems and equipments to control the equipments and appliances. The entire platform was developed considering complex scenarios, real operation situations and the future hierarchical decision structure.

In the next part, we explain the requirements of smart grids simulation especially the ability to design a system as a set of heterogeneous interacting subsystem (namely multi-modeling). We then introduce MECSYCO [1] (Multi-agent Environment for Complex SYstems CO-simulation) – formely based on AA4MM (Agents & Artefacts for Multi-Modeling) – a multi-agent approach for multi-modeling and multi-simulation, and explain how it enables to design a multi-model and its execution. The next section details different use-cases of smart grid simulations developed in the MS4SG project, before discussing the examples and concluding.

2 Requirements in Smart-Grids Simulation

As stated in the previous section, modeling and simulating a smart grid system should integrate different domains of expertise; at least power grid, communication, information and decision systems. In MS4SG, each domain uses different tools. The electricity one uses executable Modelica [1] models exported from Dymola [2] or EMTP-RV (ElectoMagnetic Transient Program, Restructured Version [3]) as modeling tools of power grid components; the telecommunication network one may use NS-3 [4] or OMNeT++ [5] depending on the protocols requirements, and decision systems can be modeled with UML-oriented tools such as Enterprise Architect [6]. Moreover, operators such as ERDF already own heterogeneous business models designed on different simulation softwares, potentially non-interoperable together.

This state is our starting assumption: tools exist and they must be integrated together. The central problem is then the multi-simulation that intends to simulate the whole as the coordinated simulation of several heterogeneous and interacting simulators. Handling heterogeneity and enabling the interaction between components can be envisaged at different levels:

- software interoperability level: software exist in each domain and must interoperate even though they are not conceived with that purpose ;
- formalism level: each domain uses the most convenient formalism(s) for itself [7], ordinary and differential equations for power grids, event-based in telecommunication networks, etc ;
- time management level: as simulation of smart grids involves several simulators, the time management in each simulator is potentially different (with constant or variable time-step, event-based, ...);

[1] The MECSYCO library is available under the Affero General Public License v3 on `http://mecsyco.fr`.

- information representation level: each software uses its own representation and data exchanges imply an information mapping between tools ;
- programming languages level: API can be in different languages ;
- hardware level: basically, for legal reason (e.g. non-replicable licenses) some software must be executed on a specific machine. More generally, some software may need specific resources and require to use dedicated hardware ;
- co-simulation norms and standards integration level: as smart grids are in an industrial finality, standards should be used in the simulation.

Different solutions exist (for example HLA [8] for software interoperability) at the different levels but they are mainly conceived independently and taking into account all the requirements imposes to integrate all of these solutions [9]. For instance, [10], [11] and [12] are focused on the coupling between power grid models and communication models. Regarding the requirements listed above, they only try to bring solutions for the integration of the different formalisms used by these two kinds of models.

In addition to these requirements, the activity of modeling and simulation brings its own (classical) requirements (calibration, validation, etc.). Moreover, the integration of several heterogeneous components brings new requirements.

The first is about incremental system design: starting from simple use cases to more elaborate ones. This implies to be able to add new models (and then simulators) in the system, but also to remove or exchange components [9].

The second is about tools supporting modeling and simulation activity. As the activity gathers several domains [13], levels of concerns have to be separated: domain experts have to focus on the modeling phase, specialists of the theory of modeling and simulation will focus on formalism integration and computer scientists on software interconnection. As a consequence, there is a need of support for domain experts in modeling to fill the gap between system domains and the co-simulation of software.

In MS4SG, we chose the MECSYCO approach, that answers these requirements by using the multi-agent paradigm, to develop simulations of smart grids.

3 The MECSYCO Approach

MECSYCO proposes concepts and tools to describe a system and to simulate it as a set of interacting heterogeneous models/simulators. It relies on the multi-agent paradigm to envisage a multi-model as a set of interacting models/simulators: each couple model/simulator corresponds to an agent, and the data exchanges between the simulators correspond to the interactions between the agents. Originality compared to other multi-agent multi-model approaches is to consider the interactions in an indirect way within the Agents and Artifacts (A&A) paradigm [14, 15].

Within this paradigm, artifacts support interactions between models as processes outside of the models and express them independently of the models' internal functioning. As a consequence, the simulators interoperability issue is managed by the artefacts. The information representation mapping issue is managed as a transformation service of the artefact in charge of the interaction between models.

MECSYCO proposes a meta-modelling approach based on the multi-agent metaphor to describe a heterogeneous multi-model. The MECSYCO multi-agent concepts are represented graphically and associated with semantic and syntactic constraints guaranteeing a non ambiguous description. These concepts are formalized with DEVS [16] (Discrete Event System Specification) operational specifications enabling to derive from a graphical representation to an executable multi-simulation. The DEVS formalism ensures the integration of different formalisms [17] with discrete and continuous dynamics [18], and include simulation algorithms. An example of the integration of equation based model with DEVS can be found in [17] and [19].

MECSYCO relies on four concepts to describe a multi-model. A **model** m_i is a partial representation of the target system implemented in a simulator (Figure 1d). It has a set of input and output ports. The models are the pre-existing parts we want to interconnect to build a multi-model. An **m-agent** $\mathcal{A}_i$ manages a simulation model m_i and is in charge of interactions of this model with the other ones (Figure 1a). The behavior of an m-agent is specified by the DEVS simulation protocol to enable agent coordination. A **model artifact** $\mathcal{I}_i$ reifies interactions between a simulation m-agent $\mathcal{A}_i$ and its simulation model m_i (Figure 1c). An interaction from $\mathcal{A}_i$ to $\mathcal{A}_j$ is reified by a **coupling artifact** $\mathcal{C}^i_j$ (Figure 1b). A coupling artifact $\mathcal{C}^i_j$ that has two roles: for $\mathcal{A}_i$, it is an **output coupling artifact**, whereas for $\mathcal{A}_j$ it is an **input coupling artifact**. The coupling artifacts can transform the data exchanged between the models using **operations**.

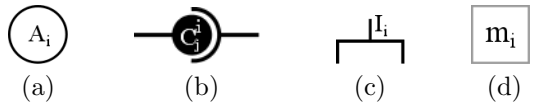

Figure 1: Symbols of the MECSYCO components for simulation (a) m-agent $\mathcal{A}_i$, (b) coupling artifact $\mathcal{C}^i_j$, (c) model artifact $\mathcal{I}_i$, (d) model m_i.

In our implementation of MECSYCO , coordination between models is achieved in a fully decentralized way thanks to the m-agents behavior. This behavior corresponds to the parallel conservative DEVS simulator based on the Chandy-Misra-Bryant (CMB) algorithm [20]. Detail on the integration of the CMB algorithm in the multi-agent paradigm of MECSYCO can be found in [21].

4 Building a Multi-Model with MECSYCO

The key steps to build a multi-model with the MECSYCO approach assume the existence of the different simulators and of the MECSYCO library (see (a)). The

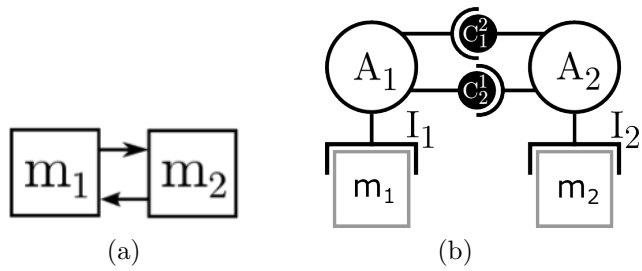

Figure 2: (a) Graph of relations between models. (b) The corresponding MECSYCO diagram.

starting point is to define the structure of the multi-model, that is how models are connected and what kind of information are exchanged between models. The next step is to define the equivalent of the intuitive graph with the MECSYCO primitives (see Figure 2). Each model corresponds to an m-agent with its model artifact and its simulator. Each information exchange is reified by a coupling artifact. This implies two aspects to be defined: i) model artifacts (see (b)); ii) information types and transformation operations between types (see (c)). From the specifications of these components, it becomes possible to have the equivalent code and then to deploy the system (see (d)).

(a) Existing Components: Simulators and MECSYCO library The approach assumes the existence of simulators corresponding to the different parts of the multi-model. In order to be compatible with MECSYCO , these simulators are supposed to be compliant with the DEVS formalism.

Each concepts of MECSYCO has operational specifications. This enables to have each basic component to be programmed (currently either in Java or C++) by instantiating predefined classes with the right types and/or operations.

(b) Connecting Simulators to MECSYCO This corresponds to the definition of a model artifact (that acts as a DEVS wrapper [19]) for each type of simulator and coupling used in the multi-model.

Concretely, this means defining five functions corresponding to the primitives of the DEVS simulation protocol[2]:

- $init()$ sets the parameters and the initial state of the model.
- $processExternalEvent(e_{in}, t, x^k)$ processes the external input event e_{in} at simulation time t in the kth input port of the model.

[2]Details on the design of a new model artifact are given in [22].

- $processInternalEvent(t)$ processes the model event scheduled at time t.
- $getOutputEvent(y_k)$ returns e^k_{out}, the external output event at the kth output port of the model.
- $getNextInternalEventTime()$ returns the time of the earliest scheduled internal event of the model.

(c) Defining Operations Operations perform the information mapping between simulators. They have to be defined and attached to coupling artifacts. They can correspond to scale transformation (from meters to kilometres), information reduction (from a list of values to its mean), etc.

(d) Software Deployment We developed several instantiations of the MECSYCO software that allow different deployment choices: from single process (in JAVA or in C++) to an hybrid (JAVA and C++), distributed (on several machines) execution thanks to a specific implementation of the coupling artifacts (with DDS middleware [23]).

5 Smart-Grids Multi-Simulation

This section describes different use cases based on real scenarios of smart grids simulation. It shows the ability of MECSYCO to effectively take into account the requirements of smard-grids simulation. Section 5.1 explains the integration of existing software in the MECSYCO framework. Next three cases (from simple to complete) are detailed and underline the kinds of requirements to meet.

5.1 Software Integration in MECSYCO : defining model artifacts

Grid Domain The design of models in the electrical domain can be done with various tools such as Modelica-based simulators [1], EMTP-RV [3], Matlab, etc. A common characteristic of these tools is their compliance with the Functional Mock-up Interface (FMI) [24]. Such models are equation based (algebraic or differential).

FMI is a standard [25] to handle the coupling of models described by differential, algebraic, and discrete equations. It enables to export a set of equations with its solver as FMUs (Functional Mock-up Units). Simulating with a set of FMUs implies to design a master component; role played by MECSYCO .

EDF R&D made the choice to use this standard to develop their models. As a consequence, we had to define a model artifact for FMU simulators. Once defined any FMU can be connected to MECSYCO . As FMI standard has been conceived for coupling purpose, it was quite easy to design these model artifacts.

Communication Network Domain: Simulators in communication network domain are mainly event-based. The major tools are NS-3 [4] and OMNeT++ [5] that were not designed to interoperate together.

In the case of the NS-3 simulator we developed a model artifact with some efforts to have some genericity [22]. For OMNeT++, we re-use an existing (but more ad-hoc) possibility (see Section 5.3).

5.2 Single Domain Example

In electrical domain, we started with simple examples to demonstrate the integration of a model in MECSYCO . The aim of these examples was i) to show the technical feasibility, ii) to produce the same results as simulations with other approach (classical simulation), and iii) to manage real cases. For this use case, the Réunion island was simulated with a steady state case of the corresponding high voltage grid (modeled with an FMU). Basically, the simulation system involves an m-agent $\mathcal{A}_{REU}$ for simulating the Réunion island power grid and another m-agent $\mathcal{A}_{CHK}$ for detecting traffic congestions by checking the state variables. We were able to integrate this model by developing the FMI generic model artifact.

5.3 Power Grid and Communication Network

A second challenge in the electrical domain, was to design systems with different interacting sub-models. As smart grids simulation requires the use of IP network models for the communication network part (*Telecom*), we worked on an example coupling electrical models and communication network models.

This use case corresponds to a medium voltage (MV) grid with two overhead feeders. When a ground phase default occurs on a feeder, the theory says that it is possible to determine whether the default is upstream or downstream from the measuring point depending on the fact that the residual voltage and the residual current are in phase or in phase opposition. The system (named HWP meaning Homopolar Wattmetric Protection) is described in Figure 3.

Messages (e.g. "residual current", "residual voltage" or "breaker to open") are sent over a communication network including the protections and a Supervisory Control And Data Acquisition. From the perspective of the grid power modeling, the two protections in Figure 3 are represented by the same model. They both detect the default but only one is supposed to clear it. The template corresponding to this use case is shown in Figure 4a.

The communication network models are executed by OMNeT++ (m-agent $\mathcal{A}_{O++}$). Since a standard HLA ambassador was available for OMNeT++ [26], we exploited this possibility for integrating it in MECSYCO . We needed to develop a special model behaving as an HLA Federate. In this way, OMNeT++ is connected to the MECSYCO multi-simulation through a standard HLA RTI. The other models in this example are contained in FMUs (m-agents $\mathcal{A}_{HWP_1}$ & $\mathcal{A}_{HWP_2}$ corresponding to protections and $\mathcal{A}_{GRID}$ corresponding to the power grid). Simulation results for both protections are shown in Figures 4b and 4c.

This example was the opportunity to run a complete example in a decentralized manner. This example is multi-formalisms, with equation-based models for the protections and the grid, and event-based models for OMNeT++.

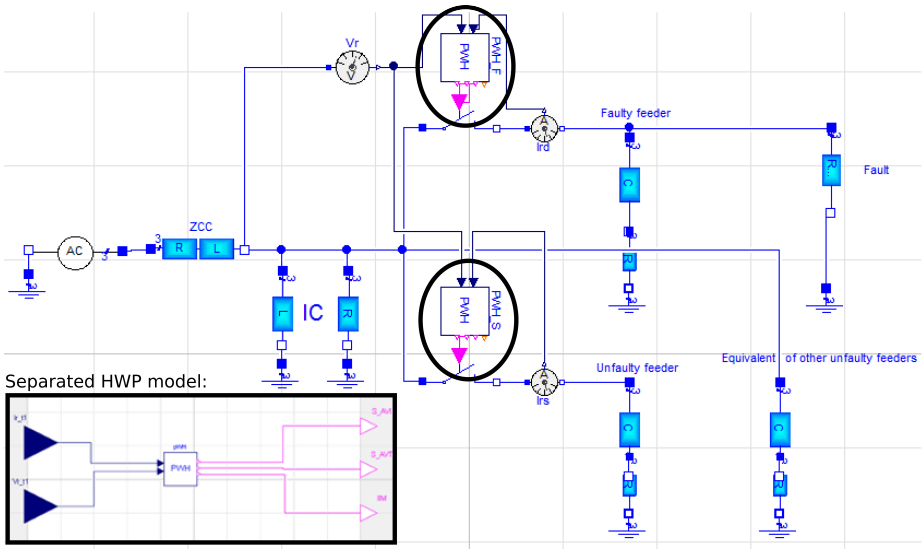

Figure 3: The HWP use case to simulate. The HWP protections (highlighted in the diagram) are separately modeled as two FMUs connected to the MV power grid FMU.

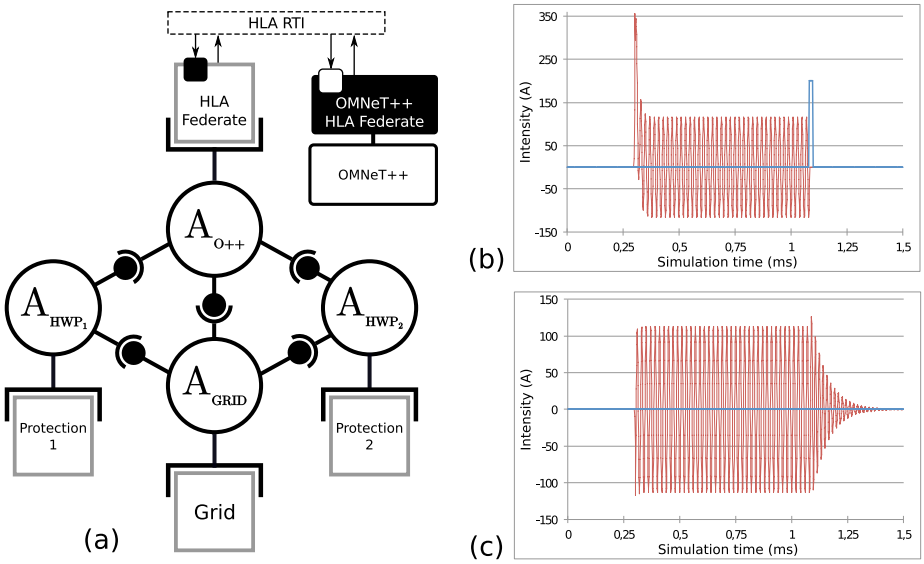

Figure 4: (a) Template corresponding to the HWP use case. (b) Simulation results for the m-agent $\mathcal{A}_{HWP_1}$, corresponding to the protection detecting a ground phase default, with the fault signal in blue. (c) Simulation results for the m-agent $\mathcal{A}_{HWP_2}$.

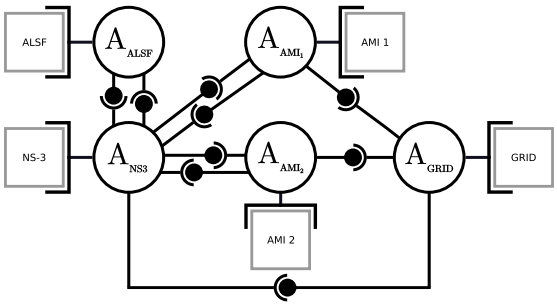

Figure 5: The cascado-cyclic load shedder use case with only two AMI.

5.4 Power Grid, Communication and Information System Domains

The final example combines electrical models and communication network models with an information (and decisional) system model (*I.S.*). These three kinds of model correspond to the three main fields of smart grid. The template of this use case is described in Figure 5. It corresponds to a load shedder with a cascado-cyclic algorithm and two consuming houses. Each house is equipped with a heat pump and an AMI (Advanced Metering Infrastructure) smart meter (m-agents $\mathcal{A}_{AMI_1}$ and $\mathcal{A}_{AMI_2}$). The heat pumps are always enabled, except in a time slot determined by the ALSF (Advanced Load Shedding Function) decisional system (m-agent $\mathcal{A}_{ALSF}$). This last sends a message to meters for indicating their time slot (specific to each house), and the meters are responsible for enabling or disabling the heat pump (thanks to the m-agent $\mathcal{A}_{GRID}$) during the simulation according to this information. Each meter has to confirm the consideration of the message received, by responding a boolean to the ALSF.

Each meter, the Power grid and the decisional system are modeled with FMUs and the communication network models are directly implemented with the NS-3 IP network simulator software. On the MECSYCO side, we used the generic model artifact for NS-3 (see Section 5.1). Each message exchanged between the decisional system and the grid or the meters is simulated by NS-3, through the m-agent $\mathcal{A}_{NS3}$.

This use case was tested with a real life demonstrator and five houses. We measured the voltage consumed by the group of houses, supposed to be reduced by the load shedding. We can then compare these measurements to the simulation results.

This example is multi-formalisms (FMUs are equation-based while NS-3 models are event-based), multi-languages (m-agents for FMUs are written in Java while the NS-3 m-agent is in C++) and multi-platforms (m-agents for FMUs are executed on Windows while the NS-3 m-agent is on GNU/Linux). Moreover, the time-scale used for the FMUs is the minute while the time-scale used for NS-3 is the nanosecond. This issue was solved thanks to an operation

Use Case	Model	Domain	Simulator	Platform	Language	Formalism	Time Scale
HWP	Protection 1	Electricity	FMU	Windows	Java	Equational	ms
	Protection 2						
	Grid						
	O++	Telecom	OMNeT++	GNU/Linux		Event	
Cascado-cyclic	AMI 1-5	Electricity	FMU	Windows	Java	Equational	s
	ALSF	I.S.	Ad-hoc			Automaton	
	NS-3	Telecom	NS-3	GNU/Linux	C++	Event	ns

Table 1: Overview of the heterogeneity and interoperability challenges managed in the two multi-domain examples.

at the coupling artifact level.

6 Discussion

The examples show the ability of MECSYCO to build multi-simulations that involve different domains of expertise, using different formalisms and different tools. Furthermore, they show its ability to integrate existing models developed on different tools and existing norms or standards. A summary of the challenges managed in the two previous examples is given in Tab. 1.

We also wish to emphasize several aspects. The coordination of simulators and the formalisms integration is ensured by the DEVS formalization of MECSYCO . To design a new application, efforts are put on simulators integration into MECSYCO framework and on information representation and translation. There is no need to redesign a new coordination algorithm. A model artefact has to be built once for all (e.g. for FMUs) for a given kind of coupling.

Using a decentralized multi-agent paradigm and implementing its concepts with a decentralized coordination algorithm ease the deployment on several machines (with possibly different operating systems) and with different programming language. The step comes after the multi-model design: domain experts can focus on the multi-model design while computer scientists will focus on implementation stage.

7 Conclusion

The paper presented how the multi-agent paradigm through the MECSYCO approach can be successfully applied for the smart grids multi-model simulation. It demonstrated through examples developed in a joint project between LORIA-INRIA and EDF R&D how multi-agent can answer the requirements of the smart grids domain.

Three detailed examples showed a real use case (La Réunion island); a multi-domain, multi-formalism example; and finally described a complete use case under development. All these examples were developed in order to provide prac-

tical (multi-agent) solutions to EDF in the perspective of performing modeling and simulation based studies instead of experiments with real demonstrators.

As short term perspectives, we envisage to integrate business tools as pre and post processing of information: generating automatically the physical domain from CIM (Common Information Model[3]) as input of the multi-simulation and connecting business tools to visualize simulations.

Acknowledgement

This work is partially funded by EDF R&D through the strategic project MS4SG.

References

[1] Fritzson, P., Engelson, V.: Modelica - a unified object-oriented language for system modeling and simulation. In Jul, E., ed.: ECOOP'98. LNCS. (1998)

[2] Dynamic Modeling Laboratory (Dymola): http://www.3ds.com/products-services/catia/capabilities/modelica-systems-simulation-info/dymola

[3] Mahseredjian, J., Dennetière, S., Dubé, L., Khodabakhchian, B., Gérin-Lajoie, L.: On a new approach for the simulation of transients in power systems. Electric Power Systems Research **77**(11) (2007)

[4] Henderson, T.R., Roy, S., Floyd, S., Riley, G.F.: NS-3 project goals. In: Proceeding of WNS2 '06, ACM (2006) 13

[5] Varga, A., Hornig, R.: An overview of the OMNeT++ simulation environment. In: Proceedings of ICST. (2008) 60

[6] Enterprise Architect: http://www.sparxsystems.com.au/products/ea/index.html

[7] Vangheluwe, H., De Lara, J., Mosterman, P.J.: An introduction to multi-paradigm modelling and simulation. In: Proc. AIS2002. Pp. (2002) 9–20

[8] 1516_WG - HLA Evolved Working Group: IEEE standard for modeling and simulation (M&S) high level architecture (HLA)– framework and rules. IEEE Std 1516-2010 (Revision of IEEE Std 1516-2000) (Aug 2010) 1–38

[9] Siebert, J., Ciarletta, L., Chevrier, V.: Agents and artefacts for multiple models co-evolution: building complex system simulation as a set of interacting models. In: Proceedings of AAMAS '10. (2010) 509–516

[3] An IEC (International Electrotechnical Commission) standard to model data in the power grids.

[10] Hopkinson, K., Wang, X., Giovanini, R., Thorp, J., Birman, K., Coury, D.: EPOCHS: a platform for agent-based electric power and communication simulation built from commercial off-the-shelf components. IEEE T Power Systems (2006)

[11] Nutaro, J., Kuruganti, P., Miller, L., Mullen, S., Shankar, M.: Integrated hybrid-simulation of electric power and communications systems. In: IEEE PES '07. (2007) 1–8

[12] Lin, H., Sambamoorthy, S., Shukla, S., Thorp, J., Mili, L.: Power system and communication network co-simulation for smart grid applications. In: (ISGT), 2011 IEEE PES. (2011) 1–6

[13] Galán, J.M., Izquierdo, L.R., Izquierdo, S.S., et al.: Errors and artefacts in agent-based modelling. JASSS **12**(1) (2009) 1

[14] Omicini, A., Ricci, A., Viroli, M.: Artifacts in the A&A meta-model for multi-agent systems. AAMAS **17**(3) (2008) 432–456

[15] Ricci, A., Viroli, M., Omicini, A.: Give agents their artifacts: the A&A approach for engineering working environments in MAS. In: Proc. AAMAS '07, ACM (2007)

[16] Zeigler, B., Praehofer, H., Kim, T.: Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems. Academic Press (2000)

[17] Vangheluwe, H.: DEVS as a common denominator for multi-formalism hybrid systems modelling. In: Proc. of CACSD 2000. (2000) 129–134

[18] Zeigler, B.P.: Embedding DEV&DESS in DEVS. In: Proc. of DEVS'07. (2006)

[19] Quesnel, G., Duboz, R., Versmisse, D., Ramat, É.: DEVS coupling of spatial and ordinary differential equations: VLE framework. In: Proc. of OICMS '05. (2005)

[20] Chandy, K.M., Misra, J.: Distributed simulation: A case study in design and verification of distributed programs. IEEE Trans. Software Engineering, (1979)

[21] Camus, B., Bourjot, C., Chevrier, V.: Combining DEVS with multi-agent concepts to design and simulate multi-models of complex systems (WIP). In: Proc. of TMS/DEVS '15, SCS (TBP)

[22] Vaubourg, J., Chevrier, V., Ciarletta, L.: Co-simulation of IP network models in the smart grids context, using a DEVS-based platform. Technical report, Inria, Université de Lorraine (2015)

[23] OMG: (DDS), v1.2 http://www.omg.org/spec/dds/1.2 (January 2007)

[24] Functional Mock-up Interface: https://fmi-standard.org/

[25] Blochwitz, T., Otter, M., Arnold, M., et al.: The functional mockup interface for tool independent exchange of simulation models. In: 8th International Modelica Conference. (2011) 20–22

[26] Galli, E., Cavarretta, G., Tucci, S.: HLA-OMNET++: An HLA compliant network simulator. In: 12th IEEE/ACM DS-RT '08. (2008)