

The Modelica language and the FMI standard for modeling and simulation of Smart Grids

Olivier Chilard¹ Jérémy Boes² Alexandre Perles² Guy Camilleri² Marie-Pierre Gleizes²

Jean-Philippe Tavella¹ Dominique Croteau¹

¹EDF Research and development, 1 avenue du général de Gaulle, 92140 Clamart France
{olivier.chilard, jean-philippe.tavella, dominique.croteau}@edf.fr

²Institut de Recherche en Informatique de Toulouse (IRIT), SMAC, Toulouse University, Université Paul Sabatier, France,
{Jeremy.Boes, Alexandre.Perles, Guy.Camilleri, Marie-Pierre.Gleizes}@irit.fr

Abstract

The smart power grids will extensively rely on network control to increase efficiency, reliability, and safety. In this context, the simulation of such complex systems is becoming an essential tool to support the development of Smart Grids.

This paper presents an overview of the EDF R&D Modelica library **GridSysPro** (GSP), which provides electrical components adapted to Smart Grid simulation; and a multi-agent approach for supporting the co-initialization process of complex network of FMUs.

Keywords: Smart Grid, Co-Simulation, Modelica.

Introduction

The smart power grids will extensively rely on network control to increase efficiency, reliability, and safety; to enable plug-and-play asset integration, such as in the case of distributed generation and alternative energy sources; to support market dynamics as well as reduce peak prices and stabilize costs when supply is limited. In turn, network control requires an advanced communication infrastructure with support for safety and real-time communication (Figure 1).

Simulating such complex systems is required for the development of Smart Grids. Several simulation tools are available on the market but these tools have two major drawbacks:

- They are generally not designed to import models developed for other tools.
- They are not adapted to large scale complex system of systems or cyber-physical systems as smart grids which require time-consuming calculation.

One solution to bypass these drawbacks is to use a co-simulation platform which can connect together several simulators and FMUs (Functional Mock-up unit).

EDF R&D is funding the development of its own co-simulation platform dedicated to the Smart Grids in partnership with LORIA-INRIA. A first release of this tool named MECSYCO is available under the Affero

GPL license v3 (<http://mecsyclo.loria.fr/>). The next published version (at the end of 2015) will upgrade MECSYCO with the coupling of different types of discrete-time or continuous-time simulators (including the FMUs) divided in three domains:

- The physics domain (process) : FMUs exported according to the FMI 2.0 standard from Dymola with models built from the EDF Modelica library **GridSysPro** or historical tools widely used at EDF (e.g. EMTP-RV) now compatible with the FMI standard;
- The telecommunication domain: NS-3, OMNeT++ or OPNeT ;
- The Information System domain with models designed with UML/SysML oriented tools.

MECSYCO is based on the Multi-Agent concept (one agent per simulator to describe a heterogeneous multi-model) and on the DEVS formalism (to conceive a decentralized execution algorithm respecting the causality constraints).

This paper provides first an overview of the EDF R&D Modelica library **GridSysPro** (GSP) composed of electrical components mapped on the zone related to the process of a Smart Grid (Figure 1). Besides that, to comply with the modeling of large scale electrical networks, a solution to co-initialize several interconnected FMUs exported from Dymola is described.

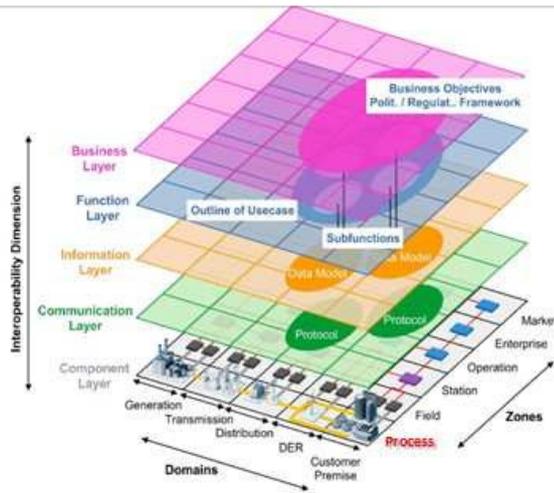


Figure 1 : The Smart Grid Architecture Model (SGAM)

1 GridSysPro Library

Modeling of electrical networks has always been a major scientific challenge for analysis and design.

Models are often used for studies of stability and control, for the analysis and optimization of power flow or for harmonic analysis and their distortion.

The common approach in electrical network simulation is based on classification of the phenomena according to their time scales (Figure 2). For each class of phenomena, particular mathematical models are developed (Figure 3).

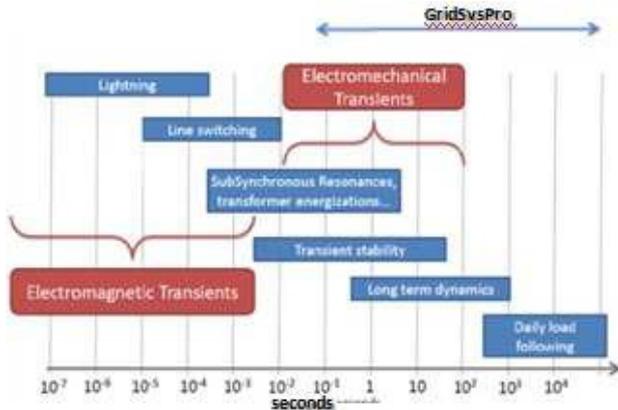


Figure 2 : Power system dynamics

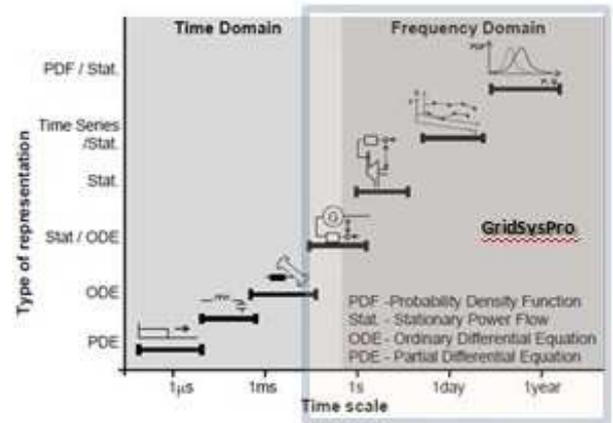


Figure 3 : Model representations for different time scales

1.1 Objectives of GSP

GridSysPro (GSP) is a Modelica library which allows stationary power load flow calculation, short circuit analyses and transient stability simulations.

The goal of stationary power load flow analysis is to find all branch currents and all nodal voltages amplitude and their angles according to electrical constraints applied at each injection node. It can help to calculate the use of power system resources and the power quality with respect to the voltage bandwidth constraints. In the real world, such analysis may be done for anticipating the effects of future operation decisions. In the simulated Smart Grid, the power flow analysis is a vital function to get the line currents and node voltages in the real power system. With this information, compliance to operating limitations such as those stipulated by voltage ranges and maximum loads, can be examined. In this way, the location of congestions and power outage situations can be identified. Moreover, the stationary power flow analysis is required to help the self-healing function, after the isolation step of the faulted section, to re-establish service to as many customers as possible from alternative sources/feeders in accordance with the operating limitations. Due to the ability to determine losses and reactive-power allocation, load-flow calculation also supports the planning engineer in the investigation of the most economical operation mode of the network.

Short circuit analysis recalculates the power flow after the occurrence of a fault in a power network. The faults may be a three-phase short circuit, a one-phase grounded, a two-phase short circuit, a two-phase grounded, a one-phase break, a two-phase break or a more complex fault.

The goal of transient stability simulation of power systems is to analyze the stability of a power system in a time window of a few seconds to several tens of seconds. Stability in this aspect is the ability of the system to quickly return to a stable operating condition after being exposed to a disturbance such as for example a tree falling over an overhead line resulting in the automatic disconnection of that line by its protection

systems. In engineering terms, a power system is deemed stable if the rotational speeds of motors and generators, and substation voltage levels can return to their normal values in a quick and stable manner.

1.2 Overview of the GridSysPro library

According to the objective retained, GSP allows the modeling of both transmission (HV) and distribution (MV/LV) electrical networks. The first version of GSP provides the following components:

- lines,
- transformer with or without load tap changer and different winding coupling,
- generators,
- adapted blocks in order to build different types of controllers like voltage and speed regulators,
- generic load which can represent different types of consumption according to sensitive factor as parameters related to voltage and frequency,
- electrical faults, analysers and breakers.

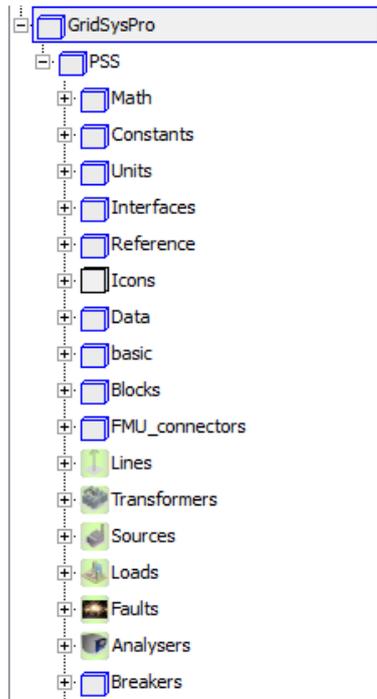


Figure 4 : Packages of GSP

Data and an Icon, which correspond respectively to external parameters and a graphical representation, are inherited by each main electrical component.

Component models are stored in hierarchically structured packages. The blue ones provide all elementary functions and models required to describe the main components (green) needed for network modeling.

1.3 Principles retained for the development of GSP

Because of electromagnetic transients are not considered in the GSP development (Figure 2 and Figure 3), power systems are described in a form using system of algebraic-differential equations. Thus the behavior of each passive component of the grid is defined by algebraic equations (complex number formulations) while the one related to electrical and mechanical parts of machines are determined by a system of differential equations.

In order to simulate large-scale three phases balanced and unbalanced networks, passive components have been defined by three single phase Quadruples $Y(QY)$.

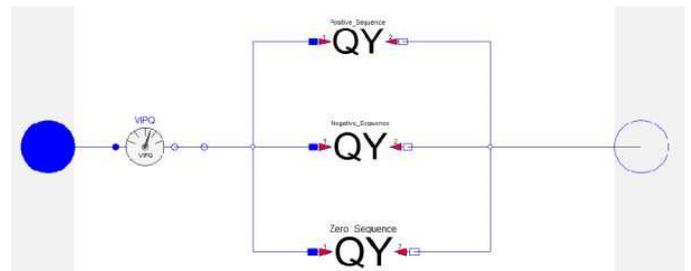


Figure 5 : model of a passive component

The algebraic equation of a QY is defined according to the equation (1) where only the variables of the positive and negative pins (p, n) are considered.

$$\begin{bmatrix} p.i \\ n.i \end{bmatrix} = [y] \cdot \begin{bmatrix} p.v \\ n.v \end{bmatrix} \quad (1)$$

Thus GSP passive components are described by three generic QY objects connected to two composite connectors containing three pins. The latter are related respectively to the positive, negative and zero sequence circuits. More precisely, the three phases a, b and c of each passive component are broken down into three sets of balanced single-phase phasors 1, 2 and 0 according to the transformation of Fortescue (2).

$$\begin{bmatrix} I_{abc}^p \\ I_{abc}^n \end{bmatrix} = \begin{bmatrix} F & 0_{3 \times 3} \\ 0_{3 \times 3} & F \end{bmatrix} \cdot P^{-1} \cdot \begin{bmatrix} Y_1 & 0_{2 \times 2} & 0_{2 \times 2} \\ 0_{2 \times 2} & Y_2 & 0_{2 \times 2} \\ 0_{2 \times 2} & 0_{2 \times 2} & Y_0 \end{bmatrix} \cdot P \cdot \begin{bmatrix} F & 0_{3 \times 3} \\ 0_{3 \times 3} & F \end{bmatrix}^{-1} \begin{bmatrix} V_{abc}^p \\ V_{abc}^n \end{bmatrix} \quad (2)$$

with:

$$\begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix} = [F] \cdot \begin{bmatrix} I_1 \\ I_2 \\ I_0 \end{bmatrix} \quad \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = [F] \cdot \begin{bmatrix} V_1 \\ V_2 \\ V_0 \end{bmatrix}$$

$$\begin{bmatrix} I_1^{pn} \\ I_2^{pn} \\ I_0^{pn} \end{bmatrix} = P \cdot \begin{bmatrix} I_{120}^p \\ I_{120}^n \end{bmatrix}$$

Therefore, the components of a grid, as lines, cables and transformers are represented by three decoupling circuits (Figure 5). Thus, the behavior of each object differs only by the definition of the y-parameter matrix $(\underline{Y}_1, \underline{Y}_2, \underline{Y}_0)$ of each QY model. Besides that an analyzer VIPQ can be used in order to provide voltage, current and power flow per phase (a, b, c).

During the initialization which is equivalent to a load flow calculation:

- generators are represented as either PQ or PV node as slack node depending on the attribute: LoadFlow_type. For each dynamic state variable on which derivative is applied one equation is given in the initial equation section,
- loads are defined as constraint of consumption.

1.4 Illustrations and validations

Some validations of GSP have been done where load flow and dynamic behaviors have been tested and compared respectively with OpenDSS and ObjectStab.

The OpenDSS is an electrical power system simulation tool developed by EPRI (USA Electric Power Research Institute) primarily for electric utility power distribution systems. It supports nearly all frequency domain (sinusoidal steady-state) analyses commonly performed on electric utility power distribution systems.

The ObjectStab package is a free Modelica Library for power systems voltage and transient stability simulations limited to single phase description of Network and dedicated to students. For GSP validation the use case of ObjectStab validated with EUROSTAG (common tool used by utilities for transient stability simulations) has been retained.

All these tests have been successful and the one related to the load flow simulation is presented here after.

The considered MV Network is a typical outgoing feeder of EDF energized by its MV primary substation (Figure 6). In order to simplify the description of networks a **Network Management Tool** developed by EDF R&D under MATLAB has been used. This NMT allows an automatic generation of the Modelica model of a network from the CIM XML file. More precisely the IEC 61970/61968 (CIM) provides a Common Information Model to support the information exchange between different EMS (Energy Management System). Its large data model provides the possibility to model physical (like cables, switches) and abstract objects (like documents, schedules, and consumer data) in the energy domain. The databases of EDF's electrical networks have been built according to the CIM standard. Therefore NMT and Dymola/GridSysPro allow an automatic Modelica implementation of EDF grids.

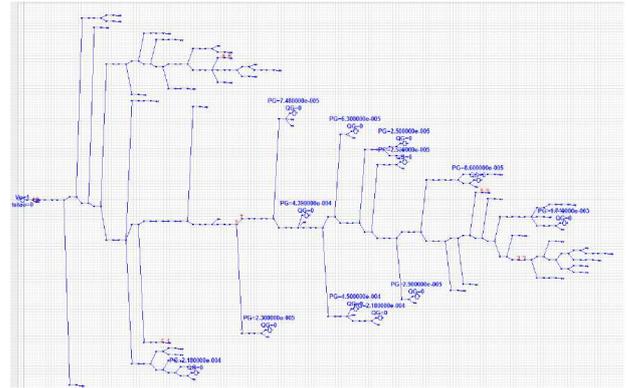


Figure 6 : the outgoing MV feeder retained for the GSP test

The Load flow simulations results are provided in Figure 7. These latter correspond to the voltage amplitude profile along the considered MV network from the primary substation to the end points of the grid.

The results obtained respectively by OpenDSS and GridSysPro are identical.

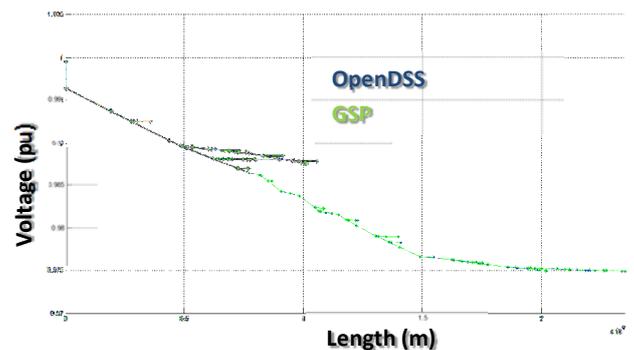


Figure 7 : The voltage amplitude profile obtained with OpenDSS and GSP

2 Co-initialization with FMUs exported from GSP

One solution to comply with the simulation of very large scale electrical networks described from GSP is to export the latter as several FMUs to be interconnected and simulated inside a co-simulation platform like MECASYCO. The segmentation of electrical networks into a set of FMUs is a design choice that depends on models and solvers properties (their execution cost...).

However, in this case the load flow calculation is distributed into each FMU and a master has to be developed in order to coordinate the calculation of each boundary variables related to input and output of each FMU.

2.1 Co-initialization with an Adaptive MultiAgent System (AMAS)

A graph of connected FMUs can be expressed in general way as following $(out) = \langle FMU \rangle (in)$. Each

FMU can be viewed as a function owning inputs and providing outputs. FMUs are connected through some of their inputs assigned to outputs of other FMUs, defining the calculation graph of the global problem. In this view, the vectors \overrightarrow{in} and \overrightarrow{out} represent all inputs and outputs of all FMU, while $\langle FMU \rangle$ is the aggregation of all FMU functions on which the input vector \overrightarrow{in} is applied. Trying to co-initialize multiple FMU is equivalent to verifying the following property: $\overrightarrow{out} = \langle FMU \rangle(\overrightarrow{in}) = \overrightarrow{in}$.

Thus, the co-initialization problem is in its general formulation equivalent to the search of fixed points in mathematics. Indeed, given a set E and an application $f: E \rightarrow E$, a point x is a fixed point if $f(x) = x$. E can be a metric space in n dimensions. When the FMUs graph owns many cycles, the problem of co-initialization corresponds to a complex fixed point search problem.

In this paper, we propose to explore the potentialities of a multi-agent approach for solving this type of fixed point search problem. For this, we choose to apply the AMAS (Adaptive Multi-Agent Systems) theory developed by SMAC team in (Georgé, Gleizes, & Camps, 2011). This theory has shown its suitability for solving complex and dynamic problems in many applications (Jorquera, Georgé, Gleizes, & Régis, 2013), (Brax, Andonoff, Gleizes, & Glize, 2013), (Capera, Gleizes, & Glize, Mechanism Type Synthesis based on Self-Assembling Agents, 2004).

In this section, we will very briefly present some important concepts of Multi-Agent Systems and the AMAS theory. The AMAS theory will then be used as a guide in the design of a multi-agent system able to solve the fixed point search problem presented above. Finally, we will present some results of the application of this multi-agent system on a GSP generated case study.

2.2 Multi-Agent Systems

A multi-agent system is a set of autonomous entities called agents, interacting in a common environment, acting to solve in coherent way a common task. This last point is important because it implies the unity of the MAS. Even if each agent has its own individual goal, in some situations their goal can possibly be in conflict with the others.

According to (Wooldridge & Jennings, 1995) and (Ferber, 1999), an agent is a physical or a software entity which:

- is autonomous,
- exists in an environment that it can perceive and on which it can act,
- has a partial representation of this environment,
- is able to communicate with other agents,
- has resources,

- has skills and can offer services.

The behavior of an agent results from its perceptions, its knowledge, its skills, and naturally its goals. It follows a life cycle in three stages repeated infinitely throughout its execution:

- the stage of *perception* during which the agent acquires new information on the environment,
- the stage of *decision* in the course of which the agent chooses the next actions to be made,
- the stage of *action* during which the agent performs the actions chosen in the previous stage.

An essential characteristic of agents is their autonomy: they decide themselves to act or not and the nature of their actions.

2.3 Adaptive Multi-Agent System Theory

The Adaptive Multi-Agent System (AMAS) theory appears suitable for the fixed point search problem (see (Capera, Georgé, Gleizes, & Glize, 2003) (Whitehead, 2008)).

Due to their distributed structure, AMAS are flexible and self-adaptable to several strategies of simulators control. The first aim of the AMAS theory is to design Multi-agent System having a coherent collective activity that achieves the right task. This property is named "functional adequacy" and the following theorem is proved: "For any functionally adequate system, there is at least a cooperative interior medium system which fulfills an equivalent function in the same environment". Therefore, it focuses on the design of cooperative interior medium systems in which agents are in cooperative interactions. The specificity of the theory: "the emergence" resides in the fact that the global function of the system is not coded within the agents. Agents have only a partial knowledge. The global function of this system emerges from the collective behavior of the different agents composing it. Each agent possesses the ability of self-organization i.e. the capacity to locally rearrange its interactions with others depending on the individual task it has to solve. Changing the interactions between agents can indeed lead to a change at the global level. This induces the modification of the global function. This capacity of self-organization enables to change the global function without coding this modification at the upper level of the system. Self-organization in AMAS is based on the capacity an agent possesses to be locally "cooperative".

Therefore AMAS agents locally cooperate in order to satisfy their own goals as well as they try to help other agents to achieve their goals. This notion of local goals is crucial for reaching a global solution, and is represented by a measure of criticality. This measure denotes the agent difficulty to reach its goals. It is used in a local way by agents in order to result in a system where the satisfaction of all agents is balanced.

Moreover, an agent will modify its behavior if it thinks that its actions are useless or detrimental to its environment. Such situations are called Non-Cooperative Situations (NCS). Some behavioral rules, specific to NCS's, help agents to solve or avoid these situations. By solving NCS's, in regard to their own local goals, cooperative agents collectively find a solution to the global problem. Therefore one can consider the behavior of an AMAS as emergent.

2.4 AMAS for co-initialization (Fixed Point Search Problem)

In this part, we present the use of the AMAS theory as a guide for the design of a multi-agent system dedicated to the co-initialization (solving the fixed point search problem). Following the AMAS theory, we start by identifying agents, their neighborhood, their criticality and their perceptions and actions. After, we will very roughly describe the behavior of agents.

2.4.1 Agents

The objective of the co-initialization is to reach a state of the FMUs graph in which every input of every FMU is equal to the output of other FMU connected in input. We chose to represent in the form of agent the connections between FMUs. In other words, if an agent represents one link between only two FMUs then the extremities of the link must be equal. More exactly, we place an agent at every output of every FMU, as in Figure 8

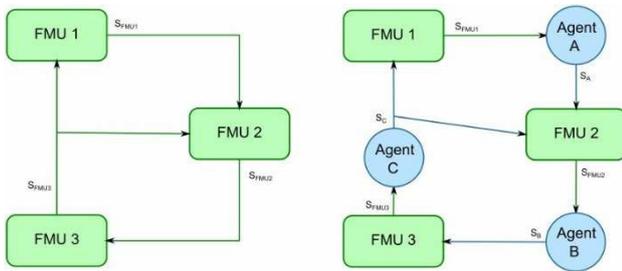


Figure 8 To the left, an example of FMU graph. To the right, the same graph with cooperative agents.

In this figure, the following equalities have to be satisfied: $S_{FMU1} = S_A$, $S_{FMU2} = S_B$ and $S_{FMU3} = S_C$. Therefore, the proposed system is only composed of one agent type corresponding to links between FMUs.

2.4.2 Neighborhood

The neighborhood of an agent is defined as the set of all agents being directly influenced by it. Therefore, the neighborhood of an agent α corresponds to all agents in output of the FMU to which α is connected in input, as well as the agent α itself. For example (Figure 6), the neighborhood of the agent A is composed of agent A and B whereas the neighborhood of the agent C consists of three agents A, B and C.

2.4.3 Criticality Measure

The fixed point search problem is solved if, after having acted, every agent observes on its inputs a value equal to the one that it had assigned on its outputs in the previous step. In other words, the problem is solved if, for every agent: $|input_t - output_{t-1}| = 0$ which constitutes the own objective of all agents of this system.

The criticality represents the difficulty that an agent has to satisfy its own objective. In this case, the criticality measure is obvious: $criticality = |input_t - output_{t-1}|$. Following AMAS theory, all agents will try to decrease their criticality to 0 what will solve the fixed point search problem.

2.4.4 Perceptions and Actions

An agent A corresponding to the output S of a FMU, perceives on its input the value of S. A also perceives the values of criticality of all agents of its neighborhood. Finally, A perceives the value of partial derivatives (Jacobian matrix) of all FMU functions to which it is connected.

The agent A can modify its own output, which is assigned a value to all FMU inputs to which S is connected. The agent action can thus be of three types: increase, decrease, or not change its value of output.

2.4.5 Agent Behavior

Our system is homogeneous, meaning that all agents possess the same behavior algorithm. The objective of each agent is to decrease the level of criticality of its neighborhood (including itself). The action of an agent can have a beneficial effect (which imply a decrease of the criticality), harmful (which causes an increase of the criticality), or indifferent (which does not provoke a variation of criticality) on each agent of its neighborhood.

Thanks to the observation of the sign of Jacobian matrix of all FMU connected to its output, each agent can, to a certain extent, know the effect of its action.

An agent has to form an idea of the amplitude and the direction in which it will vary the value of its output in order to decrease the criticality of its neighborhood as fast as possible. This information is represented by two internal variables, managed dynamically:

- δ is a positive real value corresponding to the amplitude of the variation,
- σ is an integer in $\{-1;0;1\}$, it indicates the direction of the variation.

At each life cycle, an agent modifies its output value in the following way:

$$o_t = o_{t-1} + a_t$$

Where a_t is calculated from the amplitude σ and the direction δ . Indeed, we apply a variation in the direction σ with amplitude equal to δ . Thus :

$$a_t = \begin{cases} \sigma \cdot \delta & \text{if } \sigma \neq 0 \\ \text{random}(-\delta, \delta) & \text{else} \end{cases}$$

If $\delta=0$, the variation is randomly decided between $-\delta$ and δ , both values being equiprobable. The presence of random can be justified by the fact that all agents perceive, decide and act simultaneously by following the same behavior. If the latter was purely deterministic, the system would have to cope with the problem of the bar of El Farol (Whitehead, 2008), resulting in a non-desirable synchronization of actions, and thus an ineffective exploration of the search space and a convergence outside the solution.

According to their perceptions and following cooperative rules, agents adjust their δ and σ variables in order to decrease the criticality of themselves and their neighbors. Thus, the overall criticality tends to decrease over time, while the AMAS converges toward a solution. Due to the size of this paper, we will not describe the adjustment mechanism of agents.

2.4.6 A case study: FMU graph generated with GSP

In this case study, we consider three FMU: A, B and C having respectively two, four, and two outputs. Thus there are 8 agents in the AMAS system which will initialize the network. In the Figure 9, the neighborhood graph of agents is presented. Agents are assigned to outputs of the considered FMU network. Each FMU was exported according to the FMI 2.0 standard from Dymola with EDF Modelica library GSP.

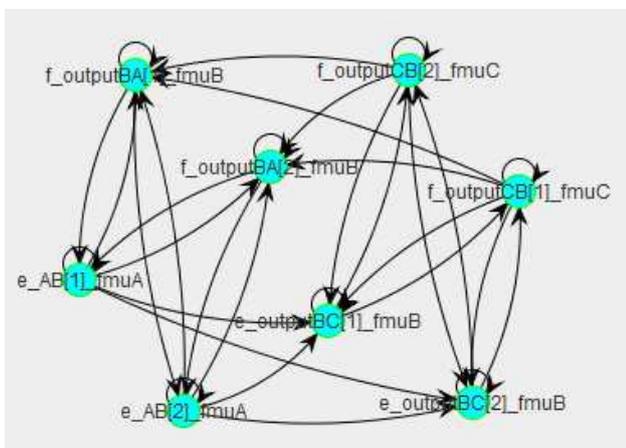


Figure 9: Neighborhood Graph

From a theoretical point of view, for solving the fixed point search problem, the criticality of all agents should reach the value 0. However, in practice, due to numerical aspects, it may be reasonable to reach a value close to 0. In this case study, the fixed point search was stopped if agent criticalities reach a value lower than 10^{-4} .

The Figure 10, shows the criticality curves of the eight agents. The total number of system cycles is indicated in abscissa and criticality values in ordinate. The best

solution is reached after around 49000 cycles, with a residue (error) of 3.74×10^{-5} .

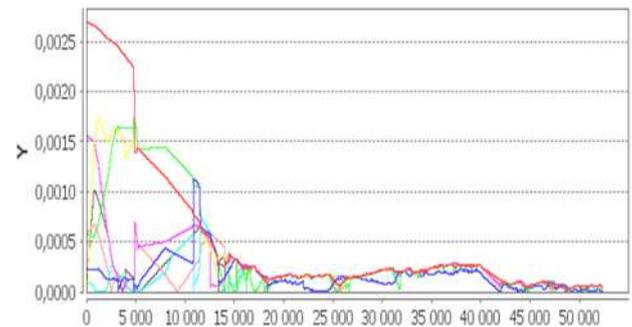


Figure 10: AMAS System Convergence

In this case study, the AMAS system converges with low criticality values (lower than 10^{-4}). Therefore, the presented system is able to co-initialize the FMU network with a reasonable precision.

Moreover, the AMAS algorithm was previously used in quite different domains with several thousands of agents (corresponding here to the number of parameters of the FMUs). The resolution principle is totally local and depends mainly on two characteristics of the application:

1. The number of agents influencing quasi-linearly the solving time,
2. the branching factor the agents (equivalent to the mean number of its neighbors). For a given class of problem (here co-initialization of FMUs), the number of cycles is stable.

Consequently the duration of the co-initialisation problem depends roughly linearly of the number of FMU.

3 Conclusions and perspectives

This paper presents an overview of the Modelica library GridSysPro (GSP) composed of electrical components mapped on the zone related to the process of a Smart Grid. Beside that to comply with the modeling of large scale electrical networks a solution to co-initialize several interconnected FMUs exported from GSP/ Dymola, is described. More precisely the interconnection of several FMUs requires the determination of initial values of all FMU inputs (co-initialization). This problem is complex and can be formulated as a fixed point search problem. We proposed the use of the AMAS (Adaptive Multi-Agent System) theory for designing a system able to solve this problem. We illustrate the suitability of the proposed system in a case study generated from GSP.

The previously presented version of GridSysPro includes several components allowing it to represent and simulate an electrical network. Nowadays, we are moving toward the concept of Smart Grid which is an evolution of the electrical network allowing notably bi-directional exchanges of energy and information through lines and an intelligent and autonomous control.

That is why an interesting perspective could be to integrate a set of advanced features to this library as part of the initiative on FMU, co-initialization and Modelica.

As an example, the first advanced feature that will be integrated is an autonomous voltage regulation system. This feature is expected, on the one hand, to be able to build a coordinated regulation between medium voltage and low voltage networks and, on the other hand, to deal with the massive integration of decentralized generators.

As previously seen, the Adaptive Multi-Agent System theory seems adapted to solve this kind of problem, notably by the amount of elements interacting in the system and by the need to support the topology changes.

The proposed approach splits the problem into two steps. Firstly, informed agents get the voltage and power values from sensors they are linked with and cooperate with others in order to help them find the missing values. And secondly, agents communicate in order to find the set of voltage set-points to guarantee the compliance with the contractual voltage range at consumption points. Such an approach of this problem allows building a voltage regulation regardless of the size of the network.

4 References

- Julien Vaubourg, Yannick Presse, Benjamin Camus, Christine Bourjot, Laurent Ciarletta, Vincent Chevrier, Jean-Philippe Tavella, Hugo Morais, Boris Deneuve, & Olivier Chillard (PAAMS 2015). SmartGrid Simulation with MECSYCO.
- Brax, N., Andonoff, E., Gleizes, M.-P., & Glize, P. (2013). Self-adaptive Aided Decision-making - Application to Maritime Surveillance. *International Conference on Agents and Artificial Intelligence (ICAART)*, (pp. 419-422). Barcelona: INSTICC - Institute for Systems and Technologies of Information, Control and Communication.
- Capera, D., Georgé, J.-P., Gleizes, M.-P., & Glize, P. (2003). The AMAS Theory for Complex Problem Solving Based on Self-organizing Cooperative Agents. *International Workshop on Theory And Practice of Open Computational Systems*, pp. 389-394.
- Capera, D., Gleizes, M.-P., & Glize, P. (2004). Mechanism Type Synthesis based on Self-Assembling Agents. *Journal of Applied Artificial Intelligence*, 921-936.
- Ferber, J. (1999). *Multi-agent systems: an introduction to distributed artificial intelligence*. Addison-Wesley Reading.
- Georgé, J.-P., Gleizes, M.-P., & Camps, V. (2011). Cooperation. (G. Di Marzo Serugendo, M.-P. Gleizes, & A. Karageorgos, Eds.) *Self-organising Software*, 193--226.
- Jorquera, T., Georgé, J.-P., Gleizes, M.-P., & Régis, C. (2013). A Natural Formalism and a MultiAgent Algorithm for Integrative Multidisciplinary Design Optimization. *IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT), Atlanta, USA, 17/11/2013-20/11/2013* (pp. 146 - 154). IEEE Computer Society.
- Whitehead, D. (2008). The el farol bar problem revisited : Reinforcement learning in a potential game. *ESE Discussion Papers*, 186.
- Wooldridge, M., & Jennings, N. R. (1995). Intelligent agents: Theory and practice. *Knowledge engineering review*, 10(2), 115-152.