Model Driven Design of IT Systems for Smart Grids

David Oudart*†
*EDF R&D, Palaiseau, France
†SAMOVAR, CNRS, Université Paris-Saclay, Télécom SudParis, Évry, France

Abstract—**Smart Grids are complex systems interfacing power grids with information and communication technologies in order to automate decision making and balancing between production and consumption. In order to facilitate the IT engineering of these systems, we aim to provide a framework relying on executable modeling and simulation. Our main challenge is to interconnect models of the different business aspects of Smart Grids: physical models of the Energy Management System (EMS), models of the telecommunication infrastructure and behaviour models of the producers and consumers. Our solution complies with Model Driven Engineering (MDE) principles. Indeed, we rely on standards, like fUML for executable modeling and FMI for cosimulation, and model transformations to improve reusability, scalability and consistency. Our framework will be tested on the case study of the EMS of an island power grid mixing battery storage, renewable and fossil energy production.**

I. INTRODUCTION

A common way to see a Smart Grid is as an electric grid which is augmented by information and communication technologies, allowing prevention, better reactivity and improved response to events such as electrical failures. As it involves many technical domains, a Smart Grid is a complex system to design and simulation is therefore valuable to evaluate various behavioral assumptions. MDE principles are well-suited to address design and development issues of complex industrial systems by reasoning from executable models all along the life cycle of the system [3]. However there lacks a general approach to interconnect models from different technical domains for complex systems. These models are designed using different tools (such as State machines, Activity Diagrams, Modelica, Discrete Events, statistical models), and the accuracy of the predictive value of the simulation depends on the proper combination of their meaning and on the correct synchronization of their execution. It is therefore mandatory to ensure the macroscopic alignment of the models with the business processes, and to maintain the consistency of the global model while refining the individual models toward an implementation.

This paper proposes a framework of methods and tools based on MDE principles and focusing on computational and applicative development. IT models are integrated in simulations that execute all the models of the Smart Grid in order to perform predictive analysis. These models are refined towards their real implementation by successive model transformations. To achieve these goals, it is necessary to ensure that our models have consistent interfaces: they must share a common meaning of the exchanged values. Moreover the business processes have to be integrated to the simulations, to synchronize the execution of the models and validate their alignment to global and strategical concerns.

II. RELATED WORK

In the electrical community, the challenge of simulating Smart Grids is not new. [9] sets up an environment to cosimulate Smart Grids with distributed control. Their approach uses software and hardware-in-the-loop simulations, with real controllers controlling a Matlab simulation of the plant through UDP and TCP communications. They particularly addressed the issue of the adaptation between event-driven and continuous components.

[5] explains how a power grid simulator should be designed in order "to accommodate the requirement for interoperability". He developed his own power grid simulator based on numerical algorithms instead of only equations to combine discrete and continuous simulations in one engine. Sensors, controllers and electro-mechanical components are all modeled and simulated with that engine and a component library written in C++. The simulator also implements an interface for time management and the injection of discrete data at runtime. To integrate a communication simulator, the power grid models are wrapped in components for the OMNET++ or NS2 simulators. This approach is efficient but provides a very specific solution for the cosimulation of smart grids and therefore lacks extensibility. New behaviors and control equipments are defined by writing C++ code, so there is little support for managing the refinement of the components during design. Also, using code as models prevents the use of MDE techniques for generating the wrappers used to interconnect the different simulators and to check the consistency of the whole simulation.

[4] developed a framework called VPNet for Smart Grid simulation. The VPNet framework provides a cosimulation coordinator implemented in C# to interface the OPNET communication simulator with VTB, a simulator for power grids with automated control. The two simulators are also both extended with interface modules to allow exchanges with the coordinator. This approach takes into account only the communication and the power aspects of a grid. The control part is integrated in the electrical simulation and there is no support for the IT aspect, which goes beyond "classical" control in smart grids. Moreover, this approach is tied to two specific simulators while our goal is to build a framework

allowing (with specific development effort) the integration of any simulator. This is necessary because companies have developed specific simulators for handling technical aspects such as transients, harmonics and unbalanced networks. These legacy tools capture business knowledge that can be used during the design and analysis of a smart grid.

[6] notes that current Smart Grid simulation environments generally focus on one domain of the system, and expresses the need for a fully integrated environment handling multi-agent control, and interactions between these agents and the power system components. The paper presents a list of requirements for appropriate smart grid simulation tools, such as using time-stepped simulations, or allowing different paradigms in the models that are integrated. The authors conclude that there is a general lack of interoperability, and particularly of “a standardized and lightweight simulator API to enable syntactic interoperability (how is data exchanged), a standardized semantic description to achieve semantic interoperability (what is the meaning of exchanged data), and a well-defined model-independent scenario description language (what components are to be simulated and how are they interconnected)”.

The existing solutions for the simulation of smart grids show what is technically possible and the kind of results we can expect. However, we would like to go further and to provide support for an evolving simulation of smart grids, from abstract models with coarse physical behaviors, up to very detailed models of the power grid, the telecommunication network, and the “smart” control algorithms. The requirements in [6] are also part of our goal.

III. PROPOSED SOLUTION

Our primary concern is to address the diversity of modeling tools and languages. We choose to consider the FMI standard [1] to perform cosimulations, instead of implementing our own ad-hoc connectors between simulation engines, in order to benefit from the reliability of a standard and to improve the reusability of our models and the modularity of our cosimulations. Moreover, the interfaces rely on primitives types which are basic enough to be implemented in a large variety of domain specific languages (DSL), and the particular format of FMU protects the intellectual property, valuable in the industrial context of Smart Grid development.

Previous work in our team proposed a framework to model and execute business processes in [7]. This framework relies on the fUML standard, which is an executable subset of UML. Class diagrams are used for data entities, and activity diagrams for dynamic behaviors. They developed specific extensions for the execution engine Moka allowing to integrate software in the loop, driven by the business process seen as an orchestrator, and then ensure Business and IT synchronization.

In our solution, the business process represents the whole dynamic behavior of the IT and its interactions with physical models. We iterate on previous work by using FMI for the interactions between models, instead of specific connectors. Business processes are written in fUML and exported to FMUs through Papyrus and its executable engine Moka, as described in [2]. Although fUML theoretically supports the description of a dynamic behavior, in practice from our experience the language is not adapted to complex behaviors manipulating a large set of variables, as is the case in smart grid applications. Model transformations would enable the refinement of a behavior from a basic one, in the early design phases, to a more precise and relevant one by switching to a more suitable paradigm and a different language. Hence we looked for a solution to delegate an operation from a fUML model encapsulated in an FMU to a process in another FMU (figure 1), by synchronizing the execution and the results. Using the “plug & play” property of an FMU, we should be able to execute the delegated behavior along its successive transformations and refinements, and eventually test the final code of an IT application.

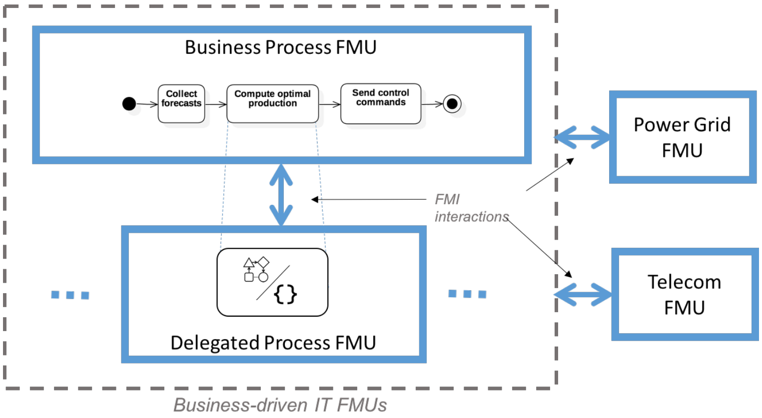

Figure 1. Business process controlling a smart grid simulation

The validation of IT behaviors would be ensured by the global cosimulation. IT, communications and power grid simulators are wrapped with FMI, and a Master Algorithm of cosimulation is in charge of managing interactions between FMU and synchronizing the time-stepped simulations. [9] showed that the size of the step interval can have a significant impact on the accuracy of the results in case one or more models follow an event-driven paradigm. [8] proposes some additions to the FMI standard to build a Master Algorithm with an event detection strategy in order to adapt the time step at best. The authors implemented this functionality in the DACCOSIM multi-simulation open source software, so we chose to use it as our cosimulation environment.

IV. PLAN FOR VALIDATION

To validate our framework, we want to use it for the design and development of an EMS for the “Île de Sein” island power grid use case. Île de Sein is a french island which has a power grid independent from the mainland, with its own production equipments. A diesel power plant is the main energy producer, and is complemented by a photovoltaic farm and battery storage to balance the production with the industrial and residential consumptions. Besides distributed control, the smartness of the grid is handled by a central EMS which can make decisions based on the knowledge of the state of all the components of the grid. One of the main purposes of the software is to optimize the costs and production, by minimizing diesel energy over renewable energy. The Telecom network links all the grid control systems to the EMS. IT systems should be modeled starting from the business process, with as much automation as possible in order to ease the implementation of the applications, to ensure business and IT experts can work together. Then, the relevance of the models should be analyzed through cosimulations of business, IT, Telecom and electrical models. Finally, we want to validate the consistency between IT models and their implementation, by replacing in the cosimulation the EMS FMU by the final code of the EMS, itself wrapped in an FMU. In each cosimulation case, the global results of the cosimulation are our main criteria to analyze and validate the behavior of the business and IT components. It implies to consider that the physical models are correct and that their behavior is known and determined.

V. Expected Contributions

The expected contributions of our work are a methodology for designing and modeling smart grids in an executable way, and tooling to support this methodology. The methodological aspects aim to guide the designers in the construction of the models of the different parts of the grid and in the definition of the interfaces between these models. The tooling aspect handles the executability of the models and the proper orchestration of their executions. It also handles the consistency of several refinements of a model to allow the simulation of the whole grid at different levels of detail.

VI. Conclusion and Current Status

So far we have mostly worked on the tooling aspects and focused on a cosimulation involving only the power grid and the business/IT models. Electrical components and their automated control are modeled in Modelica, with a load-flow approach (balance between power production and consumption), and wrapped with FMI (OpenModelica and Dymola implement the model export to FMU). We managed to simulate a basic control of the grid FMU with a fUML process wrapped in another FMU, which amounts to interfacing an event-based process with a continuous physical system over FMI. This work is necessary to determine what kind of information must be available in the models, which has consequences on the methodology used for their design. However, we have also worked on some methodological aspects to take into account the separation of different business types of knowledge in the design of these models.

References

[1] T. Blochwitz, M. Otter, M. Arnold, C. Bausch, C. Clauss, H. Elmqvist, A. Junghanns, J. Mauss, M. Monteiro, T. Neidhold, D. Neumerkel, H. Olsson, J.-V. Peetz, and S. Wolf. The functional mockup interface for tool independent exchange of simulation models. In *Proceedings of the 8th International Modelica Conference; March 20th-22nd; Technical University; Dresden; Germany*, pages 105–114. Linköping University Electronic Press, 2011.

[2] S. Guermazi, S. Dhouib, A. Cuccuru, C. Letavernier, and S. Gérard. Integration of UML models in FMI-Based co-simulation. In *2016 Symposium on Theory of Modeling and Simulation (TMS-DEVS)*, pages 1–8, April 2016.

[3] J. Hutchinson, M. Rouncefield, and J. Whittle. Model-driven engineering practices in industry. In *2011 33rd International Conference on Software Engineering (ICSE)*, pages 633–642, May 2011.

[4] W. Li, A. Monti, M. Luo, and R. A. Dougal. VPNET: A co-simulation framework for analyzing communication channel effects on power systems. In *2011 IEEE Electric Ship Technologies Symposium*, pages 143–149, April 2011.

[5] J. Nutaro. Designing power system simulators for the smart grid: Combining controls, communications, and electro-mechanical dynamics. In *2011 IEEE Power and Energy Society General Meeting*, pages 1–5, July 2011.

[6] S. Rohjans, S. Lehnhoff, S. Schütte, F. Andrén, and T. Strasser. Requirements for Smart Grid simulation tools. In *2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE)*, pages 1730–1736, June 2014.

[7] Rachida Seghiri. *Modeling and simulation of an Entreprise Architecture-Application to Smart Grids (in French).* PhD thesis, Université Paris-Saclay, July 2016.

[8] J.-P. Tavella, M. Caujolle, S. Vialle, C. Dad, C. Tan, G. Plessis, M. Schumann, A. Cuccuru, and S. Revol. Toward an Accurate and Fast Hybrid Multi-Simulation with the FMI-CS Standard. In *21 st International Conference on: Emerging Technologies and Factory Automation (ETFA-2016)*, Berlin, Germany, September 2016.

[9] C.-H. Yang, G. Zhabelova, C.-W. Yang, and V. Vyatkin. Cosimulation Environment for Event-Driven Distributed Controls of Smart Grid. *IEEE Trans. Industrial Informatics*, 9(3):1423–1435, 2013.