INCLUDING IN HLA FEDERATION FUNCTIONAL MOCKUP UNITS FOR SUPPORTING INTEROPERABILITY AND REUSABILITY IN DISTRIBUTED SIMULATION

Youssef Bouanan, Simon Gorecki, Judicaël Ribault, Gregory Zacharewicz, Nicolas Perry

INCLUDING IN HLA FEDERATION FUNCTIONAL MOCKUP UNITS FOR SUPPORTING INTEROPERABILITY AND REUSABILITY IN DISTRIBUTED SIMULATION

Youssef Bouanan
IMS lab.
University of Bordeaux
351 Cours de la Libération
33400 Talence, France
youssef.bouanan@u-bordeaux.fr

Simon Gorecki
IMS lab.
University of Bordeaux
351 Cours de la Libération
33400 Talence, France
simon.gorecki@u-bordeaux.fr

Judicael Ribault
IMS lab.
University of Bordeaux
351 Cours de la Libération
33400 Talence, France
judicael.ribault@gmail.com

Gregory Zacharewicz
IMS lab.
University of Bordeaux
351 Cours de la Libération
33400 Talence, France
gregory.zacharewicz@u-bordeaux.fr

Nicolas Perry
Arts et Métiers ParisTech
ENSAM of Bordeaux
Esplanade des Arts et Métiers
33400 Talence
perry.nicolas@ensam.eu

ABSTRACT

Modeling and Simulation is attempting to tackle more and more complex systems, which makes their design highly challenging. Complex systems' Modeling and Simulation (M&S) require the consideration of several simultaneous points of view and involve skills from different scientific and technical fields. Distributed Simulation domain answer the question of coupling and running together heterogeneous components, e.g. IEEE 1516-2010 - High Level Architecture is one of the most used standard. Also, Functional Mockup Interface provide standard designed for the coupling of simulation tools (simulator coupling, tool coupling), and coupling with subsystem models, which have been exported by their simulators together with its solvers as runnable code. In this paper, we aim to provide bridge between HLA and FMI standard in order to couple those technologies.

Keywords: Model Driven Architecture; Distributed Simulation; High Level Architecture, Business Process Model and Notation.

1 INTRODUCTION

The Modeling & Simulation (M&S) concept is now a required step in any design of complex systems. It allows to early represent its behavior and interaction. The modeling phase describes a process and allows the development of an executable simulation which virtually designs our subject and anticipates its study. As technologies are growing, the systems complexity increases, and makes the system more difficult to simulate. Here comes the role of Distributed Simulation (DS): one simulation is divided into multiple sub functions (or models) from a large system. Each function is executed on a different computer possibly geographically distributed from others. From a general point of view, this solution divides complex problems into simpler modular sub problems, but also rises interoperability issues.

Modeling and Simulation (M&S) of complex systems requires the simultaneous consideration of several points of view. The system behavior has to be considered at different levels and scales. In addition, the study of these systems involves skills from different scientific, business and technical fields. The challenge is then to reconcile these heterogeneous points of view, and to integrate each domain models and tools (or subsystems) within frameworks of the M&S process. Different solutions and architectures have been proposed to simulate models in a distributed environment (e.g. DEVS/SOA, DEVSML, DEVS/REST, etc.) (Mittal, Risco-Martín, and Zeigler 2009, 2007; Al-Zoubi and Wainer 2009)

Two of the most popular efforts going in these directions are FMI (Functional Mock-up Interface) and HLA (High Level Architecture):

HLA is an IEEE standard (Association 2010) for distributed computer simulation systems (Association 2010). In the HLA standard, a distributed simulation is called Federation (see Figure 1). A Federation is composed of several HLA simulation entities, called Federate, which can interact with them by using the Run-Time Infrastructure (RTI). The RTI represents a Federation execution backbone and provides a set of services to manage the communication and data exchange between Federates.

FMI (Functional Mock-Up Interface) (Blochwitz et al. 2012) establishes itself as a standard for model exchange and co-simulation of equational models. The FMI functions are used (called) by a simulation environment to create one or more instances of the FMU (Functional Mockup Unit) and to simulate them, typically together with other models. An FMU may either have its own solvers (FMI for Co-Simulation) or require the simulation environment to perform numerical integration (FMI for Model Exchange). It enforces some generic rules and a software interface to manipulate equational models and their numerical solver using a combination of XML-files and compiled C-code. On that interface, any equational component can be embedded into an FMU (Functional Mock-up Unit) helping to solve the interoperability problem for the co-simulation of equational models. Then, the numerical resolution of a system can be performed by defining a set of communication points between the FMUs according to a trade-off between the accuracy of the simulation results and the performances of the co-simulation process (Camus et al. 2016). The FMI standard defines two interfaces: FMI for Model Exchange and FMI for Co-Simulation (Blochwitz et al. 2012).

Here, we present a hybrid distributed simulation based on HLA and FMI. The integration of an FMU in a HLA Federation has several benefits. This combination can be exploited to create a complete solution to enable reuse, interoperability and distributed execution of simulation models. In the last few years, much research effort has been devoted to support this integration perspectives (FMI for HLA and HLA for FMI). Bastian et al. proposed an integration mechanism based on a master algorithm for Co-Simulation using FMI. This master algorithm can be applied depending on the properties of the involved slave simulators (Bastian et al. 2011). Awais et al. discussed the use of the HLA RTI as a master for the FMI simulation components to make FMI-based simulation components usable as plug and play components, on variety of distributed environments including grids and clouds. Others are based on the definition of wrappers for

integrating and (re)using FMUs (Functional Mock-up Units) in HLA-based simulations (Awais et al. 2013) and of some possible extensions to FMI to include HLA features.

In this study, we propose to reuse existing FMUs as components of a federate (Hybrid federate) in an HLA-compliant distributed simulation, i.e. federation. By this way, FMI will also serve as a model interface for distributed simulation entities in the concept of design phase.

The rest of the paper is organized as follows. Section 2 describes the different concepts and technological backgrounds of this work. Then, section 3 presents a brief study and discussion about the work related to the combination of FMI and HLA standards. It discusses also the detailed steps of the proposed framework and the reflection on how to include in HLA Federation entities that are available as FMUs. Finally, Section 4 details a simple demonstration of FMUs simulated in a distributed environment.

2 BACKGROUND

From a M&S process perspective, distributed simulation involves dealing with different subsystems forming a coupled problem that is modeled and simulated in a distributed manner. Indeed, the different domains of expertise may have different modeling and simulation tools, and can be modeled and implemented within different languages. Moreover, some of these tools might be available only on some specific hardware. Interoperability processes are required to synchronize these heterogeneous tools and manage exchanges of data amongst them.

Distributed simulation is a paradigm to model dynamic, heterogeneous, and spatial distributed systems. Its aim is not only to speed up the simulations, but also to serve as strategic technologies to link various types of simulation components (Chen et al. 2008). There are several approaches in the field of M&S offering interesting solutions to the challenges of the simulation models interoperability and their execution on distributed computing environment. Two of the most popular efforts going in these directions are FMI (Functional Mock-up Interface) and HLA (High Level Architecture).

2.1 High Level Architecture (HLA)

In the computer simulation domain, distributed simulation is one of the most useful approaches to reuse and run together different applications. Indeed, it consists of several co-running components (often associated with one or more functions) which can be processed by different processors. All of these components are part of a single execution which can be located on different computers / servers, hence the term "distributed". This concept of functions relocation makes the loads distribution possible on different machines, increasing the efficiency of a program.

One of the advantages of distributed simulation is to solve interoperability problems. Interoperability is the interactions ability between systems. This issue appears when several highly dissimilar systems (by their internal structure, exchanged data format, or semantic data) must communicate. The interoperability issue must be considered if interactions are at data level, service level or process level (Zacharewicz et al. 2009).

Indeed, in distributed simulations, the components are modular. They can have a heterogeneous architecture and exchange different kind of structured messages. This enables the solving of interoperability problems.

HLA defines a framework which allows the creation of global execution. This framework defines how to create a "global" simulation, which is made of several distributed simulation participants. Distributed simulation participants are called federates, they can communicate with one another. It was originally created by the Office of Defense Modeling and Simulation (DMSO) of US Department of Defense (DoD) to facilitate the assembly of stand-alone simulations with a different architecture. The original goal was the reuse and the interoperability of military applications, simulations and sensors. This standard is designed to resolve interoperability and reusability issues between software components. Another interesting aspect of this specification is the synchronization aspect. It allows to dynamically manage interoperability issues

with simulations exchange messages: it must be ensured that messages are sent at the right time, in the right order, and that they do not violate causal constraints. To do this, various systems for synchronization of processes and time management are proposed by HLA. An example of a federation with four federates can be seen in Figure 1.

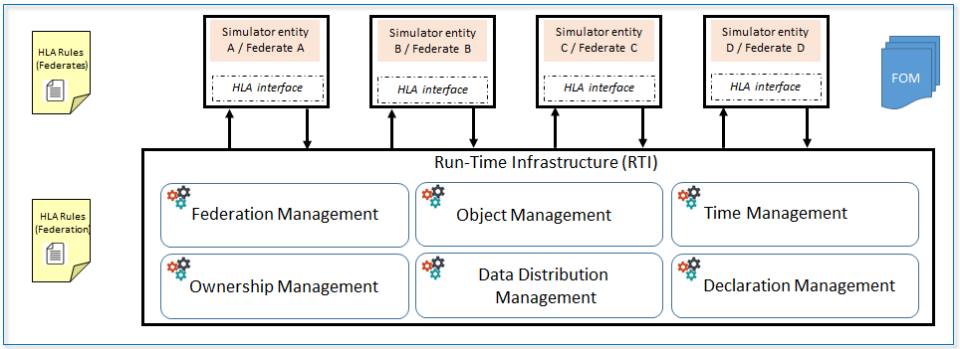

Figure 1: HLA Architecture diagram.

According to the HLA standard, each simulation participating to the application is called "federate". A classical HLA federate consists of a simulation model and local RTI component (LRC). The simulation model is a physical, mathematical, or logical representation of processes and systems. These entities can communicate with each other through a Run-Time Infrastructure (RTI). It is the RTI which manages the federation, authorizes federates to communicate or not, and provides various services such as time management, file or data exchange, etc. The FOM file is a XML file which describes interactions/communications between federates (see Figure 1).In our application case, this notion of distributed simulation will be tackled by the High Level Architecture standard (HLA) (IEEE Computer Society 2010b). It will support the specification of our software architecture.

2.2 Functional Mock-up Interface (FMI) for Co-Simulation

The Functional Mock-up Interface (FMI) for Co-Simulation interface is designed both for the coupling of simulation tools (simulator coupling, tool coupling), and coupling with subsystem models, which have been exported by their simulators together with its solvers as runnable code. It is an interface standard for the solution of time dependent coupled systems consisting of subsystems that are continuous in time or time-discrete (Bastian et al. 2011; Blochwitz 2014; Sievert 2016). It provides interfaces between master and slaves and addresses both data exchange and algorithmic issues. FMI for Co-Simulation consists of two parts (Figure 2):

- Co-Simulation Interface: a set of C functions for controlling the slaves and for data exchange of input and output values as well as status information.
- Co-Simulation Description Schema: defines the structure and content of an XML file. This slave specific XML file contains “static” information about the model (input and output variables, parameters, …) and the solver/simulator (capabilities, …). The capability flags in the XML file characterize the ability of the slave to support advanced master algorithms which use variable communication step sizes, higher order signal extrapolation etc.

A component implementing the FMI is called Functional Mock-up Unit (FMU). It consists of one zip file containing the XML description file and the implementation in source or binary form (dynamic library). A master can import an FMU by first reading the model description XML file contained in the zip file. Coupling simulators by FMI for Co-Simulation hides their implementation details and thus can protect intellectual property.

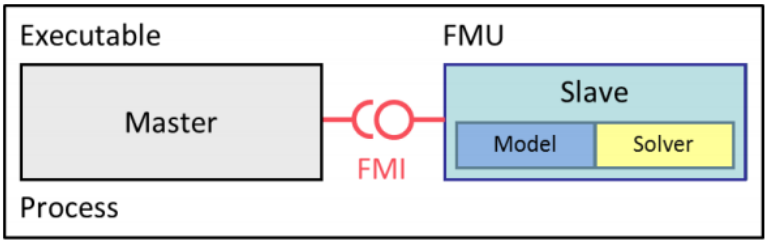

Figure 2: Co-simulation with generated code on a single computer.

Co-simulation exploits the modular structure of coupled problems in all stages of the simulation process beginning with the separate model setup and preprocessing for the individual subsystems in different simulation tools (which can be powerful simulators as well as simple C programs). During time integration, the simulation is again performed independently for all subsystems restricting the data exchange between subsystems to discrete communication points ti_c.

FMI for Co-Simulation provides an interface standard for the solution of time dependent coupled systems consisting of subsystems that are continuous in or time-discrete. In a block representation of the coupled system, the subsystems are represented by blocks with (internal) state variables x(t) that are connected to other subsystems (blocks) of the coupled problem by subsystem inputs u(t) and subsystem outputs y(t). In this framework, the physical connections between subsystems are represented by mathematical coupling conditions between the inputs u(t) and the outputs y(t) of all subsystems, (Kübler and Schiehlen 2000) (see Figure 3).

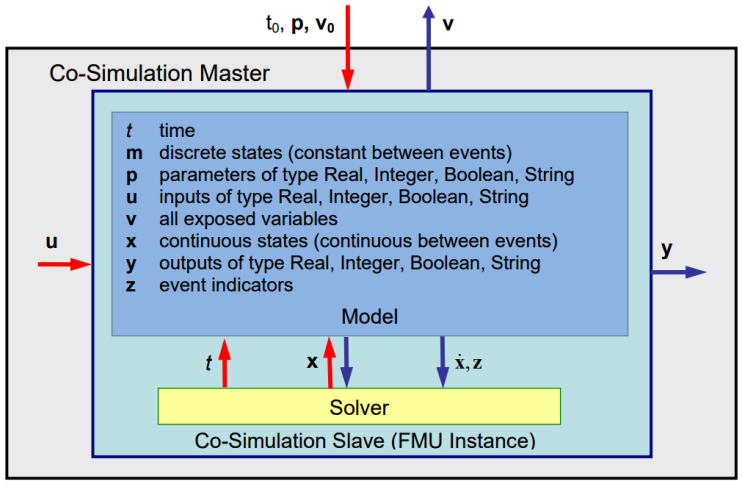

Figure 3: Data flow at communication points.

Computing the solution of an FMI Co-Simulation model means to split the solution process in three phases (Figure 4):

1. Initialization Mode: This mode is used to compute at the start time $t0$ initial values for internal variables of the Co-Simulation slave, especially for continuous-time states and for the previous discrete-time states.

2. Step Mode: This mode is used to compute the values of all continuous-time and discrete-time variables at communication points by numerically solving ordinary differential, algebraic and discrete equations. FMU model is executed via calling *doStep()* method. Intuitively, before running a step, FMU input parameters are set by calling *FMUSetXXX(…)* and after the completion of this step the model output parameter are read by the master via calling *fmiGetXXX(…)*.

3. Termination phase: In this phase, the model components are unloaded and the memory is cleaned up.

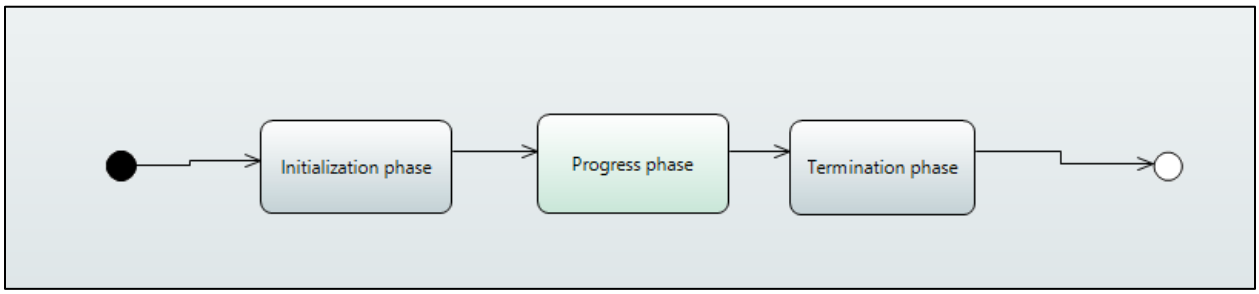

Figure 4: FMU Co-Simulation Model Computational Flow

2.3 Hybrid simulation

A simulation study consists of several phases, for example, phenomenon identification, conceptual modeling, input and output data analysis, model translation/implementation, verification, validation and experimentation. In Mustafee et al., authors distinguish between hybrid simulation and hybrid M&S study based on the techniques applied, and also the stage in which they are applied (Mustafee et al. 2015). According to (Powell and Mustafee 2014), on the one hand, the use of multiple M&S techniques in the model implementation stage is referred to hybrid simulation. On the other hand, hybrid M&S study refers to the application of methods and techniques from different disciplines such as operations research (other than M&S), systems engineering and computer science to one or more stages of a simulation study.

3 COMBINING HLA AND FMI

To include in HLA simulation entities that are available as FMUs, a HLA component has to act as a master for the FMUs in order to manage their lifecycle during the HLA simulation. The master serves as an interface, establishes connections and exchanges data between the FMUs which act as slaves. Slaves are assumed to communicate with the master only. In more details, the master has the responsibility to orchestrate the steps of Co-Simulation through the execution of two tasks: (i) track and control the data exchange between the Federation and the controlled FMUs; and, (ii) synchronize the simulation time between the HLA Federation and the FMUs (Garro and Falcone 2015). This integration has many advantages:

- Heterogeneous FMUs can be reused into a HLA simulation environment without making structural or behavioral changes on them;
- Greater interoperability among FMUs because they can interact with one another in a distributed computing environment through HLA;
- FMUs can be created and tested independently through well-established simulation packages compliant to the FMI standard;
  All model functionalities as well as the solver are included in the FMU making it possible to use such FMUs without software used to generate/export them (e.g. MATLAB/Simulink, SimulationX, etc.). FMUS exported from SimulationX either for model exchange or for co-simulation can be used without SimulationX and do not require any runtime license (“FMI” n.d.).

3.1 Including in HLA simulation/Federation entities that are available as FMUs

The key difference between FMI and HLA is that HLA provides specific mechanisms for data exchange (HLA follows the publish-subscribe messaging pattern) and time management (it used to handle time in the federation and ensure that the federates run in a specific order) that enable the integration in a distributed computing environment of heterogeneous simulation models created according to the HLA standard. Only

FMUs generated according to the FMI for Co-Simulation modality can be taken into consideration for the inclusion in a HLA simulation. In the FMI for Model Exchange modality the solver module is not part of the FMU so the integration such of this kind of FMU in a HLA simulation it is not practicable (Garro and Falcone 2015). In (Garro and Falcone 2015), authors presented two possible solutions (Adapter-based and Mediator-based) to integrate in HLA simulation/Federation entities that are available as FMUs.

In the following, the two approaches based in the proposal presented in (Garro and Falcone 2015) to realize the integration FMI for HLA-based simulation. The two approaches (Adapter-based and Mediator-based) the concept of Hybrid Federate to manage the lifecycle of an FMU.
In the *Adapter-based approach* the *Hybrid Federate* is composed by two elements (Figure 5):

- An FMU containing the FMI for Co-Simulation API; the behavior of the component to simulate and its solver.
- An Adapter managing all the interactions between the RTI infrastructure and the FMU (e.g. publish/subscribe of the attributes that are produced/used by the FMU, Object discovery, Datatypes mapping), as well as the lifecycle of the FMU.

FMI Co-Simulation scalar variables can only map to HLA basic data types because FMI Co-Simulation only supports the following primitive types: *real*, *integer*, *string, Boolean* and *Enumeration* and HLA attributes can represent any data type structure, from basic data types to the complex data type structures (Yilmaz et al. 2014). In this context, a simulation environment using complex data types cannot be directly supported by FMI Co-Simulation (the hybrid federate cannot use or produce complex data types).

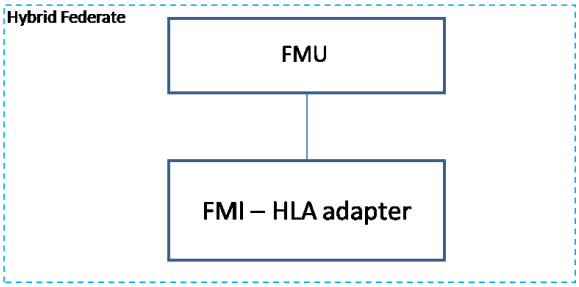

Figure 5: The Adapter-based integration approach.

In the *Mediator-based approach*, the structure of the Hybrid Federate is composed of three elements (see Figure 6):

- A set of FMUs, each of which contains the behavior of the component to simulate and its solver;
- A HLA Federate containing its own simulation logic and uses the FMUs to simulate specific components.
- A mediator layer to coordinate/orchestrate the behavior of the whole Hybrid Federate. It allows the communication between FMUs and HLA Federate.

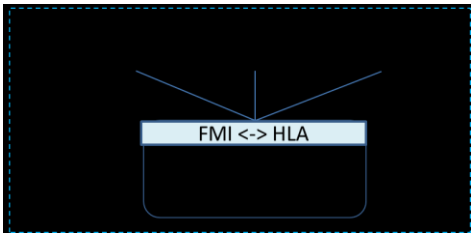

Figure 6: The Mediator-based integration approach.

3.2 Hybrid federate specification Using BPMN

The use of graphical modeling to define and model part of distributed simulation process has several benefits: (1) it shows R&D projects and requirements in context rather than in isolation; (2) it allows specialists to define in a unified and standardized way the execution process of the simulation. Several modeling languages were introduced for defining industrials workflows, but Business Process Model and Notation (BPMN) is most widely adopted by users. BPMN is a graphical notation for drawing business processes. BPMN has been proposed by the Business Process Modelling Initiative (BPMI) and is currently maintained by the Object Management Group (OMG, 2011.) which provides this standard for IT and business actors. It is frequently supported by a computer program which enables a quite easy graphical description of complex processes. It provides a standard notation which is easily understandable by all stakeholders; also bridges the communication gap frequently occurring between business process design and implementation.

In the last few years, much research effort has been devoted to support the specification of HLA-based simulation using BPMN as standardized graphic user interface; some of them aim at providing a workflow simulation system that
combines the functionalities provided by the BPMN and HLA standards. More in detail, the system allows domain experts to capture and specify the business work in the form of BPMN models (Lee 2010). Others are based on using BPMN for the HLA core concepts (*federate and federation*) representation (Falcone et al. 2017). In this section, a graphical representation of the hybrid Federate based on the mediator approach is proposed. The HLA federate specification is based on the proposal presented in (Garro and Falcone 2015). A BPMN model that describes the main activities performed by a generic Mediator-based approach Hybrid Federate is shown in Figure 7.

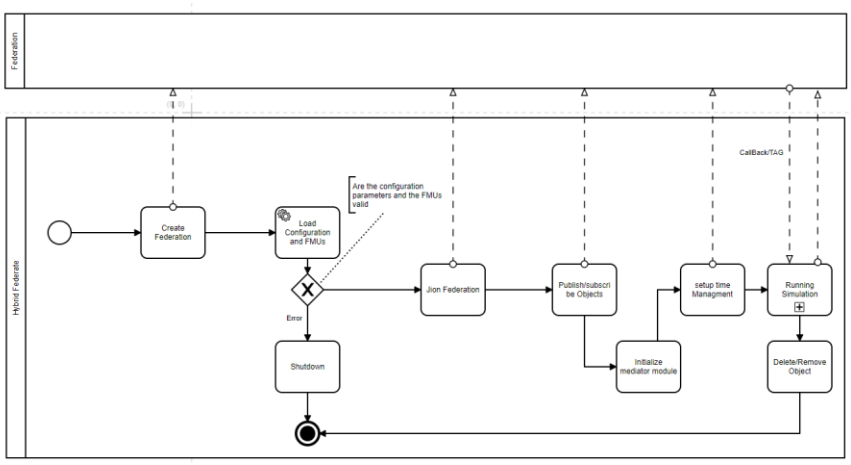

Figure 7: Lifecycle of the HLA Hybrid Federate.

3.3 Business context

In our context, a company designing solar power plants have special needs. This project consists in installing solar panels fields to provide electricity in a large areas which are not powered so far. However, the transport of solar panels fields is extremely expensive. To reduce this cost/blow, they are designing a mobile factory which manufactures the solar panels on site. Rather than transporting finished products, only the mobile plant and raw materials would be carried out. The main challenges of this project are : the factory miniaturization to fit in the least transport containers (around 20), risks management caused by low knowledge, and designing resistant structure depending on the environment of the power plant.

To solve each one of the issues, the company gave rise to several works which each subject deals with one problem:

- Optimization and decision helping for defining the structure foundations of the solar panels field, depending on the ground structure.(Piegay and Breysse 2015)
- Study of the concept maturity integration in decision making process: applied for designing the solar transmitter supporting structure.
- Study and dimensioning of the mobile factory dimensions, cost, etc., according to the demand. (Benama 2014)
- Study of project management method integrating risks. Calculating risks probabilities into project management (Rodney 2014)
- Tool to model company workflows (based on BPMN) in order to represent the future solar power plant. Its main goal is to control each simulation (Posse 2015)

Most of these works have tackled M&S to solve widely existing issues related to various domains. Our objective consists in proposing an interoperable architecture which handles all previous works using HLA and FMI standards to create a general middleware in order to simulate a whole system in a distributed environment.

4 DEMONSTRATION

In this section, the idea is to use the features of HLA and FMI to design a set of federates and a federation to show viability of HLA for distributing simulation of FMUs. This part provides a base proof of concept of FMUs simulated in a distributed environment. The HLA federation was based on the Bouncing Ball model. The Bouncing Ball model is a commonly used model in the modeling domain. It is a simple simulation of a ball being dropped and bouncing on the ground, it simulates the height and the velocity of the ball. The Bouncing Ball FMU file has been generated using the FMU SDK by Qtronic (Pohlmann et al. 2012). In this simulation scenario, two federates will join the HLA federation Figure 10 They became as follows :

- Federate A: federate responsible for displaying the ball simulation in a 2D viewport.
- Hybrid federate B : A federate responsible for simulation of the Hight and the velocity of the ball. This FMU run at a statically hardcoded speed.

We ca see on the figure below the hybrid federation technical architecture. The federate A receive data from HLA communication standard according to an interaction subscribed on the hybrid federate B. This interaction is described in the Federation Object Model (FOM) (Figure **9**) file which contain coordinates of the simulated ball. At the reception of these informations, federate A trace a point on a graphic with java.awt package. All along the simulation, data are calculated by the FMU, they are sent through HLA interface with interaction, and drawn by federate A.

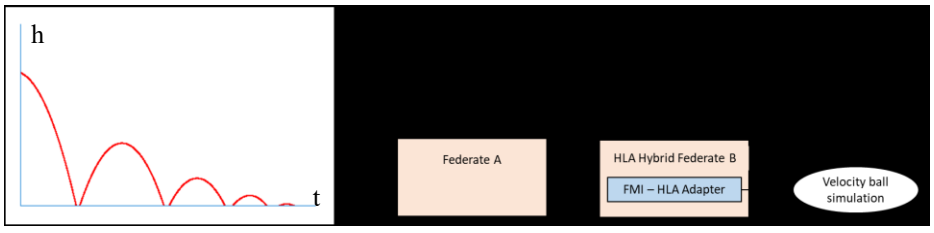

Figure 8: HLA Hybrid federation

From the Hybrid federate B point of view, the objective is to be a master and control the Functional Mockup Units thanks to javaFMI wrapper. Indeed, this library allow to import FMU with the FMI standard. Data result of this FMU are published and sent to Federate A through HLA RTI. This information flow must be declared on the FOM file according to Figure 9. In our context, data types used are basic,so they can be easily converted to HLA standard. From the time management point of view, the Hybrid federate B will request RTI to advance in time after each running step time of the FMU executed.

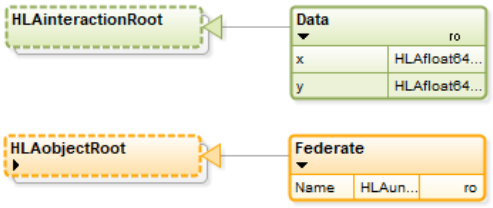

Figure 9: Federation Object Model

Informations describes on the upper figure are contained into the XML FOM file of the Figure **8**.

HLA standard provide two types of data exchange :

- "Interaction Class" are not persistent over the time and can have parameters. They are most of the time used to describe ephemeral entities during the simulation. Interactions are used here for coordinate "x" and "y" of the ball which are generated by the FMU, and published over the RTI by the hybrid federate.

- "Object Class" are persistent during the simulation. They can have attributes which can be updated by federate. In our application example, "name" will be used to identify "Federate A" and "Hybrid Federate B" during the simulation.

As we can see on Figure 10, we are using the free RTI Pitch version which allow us to create and manage a federation that can contains two federates. After execution of the distributed simulation, we can observe the result generated by the federate A in the left of Figure 8. In future, we will use the free Portico opensource RTI which allow HLA standard 1516e.

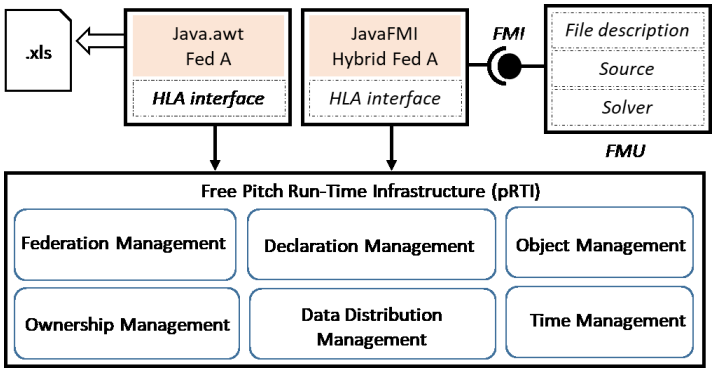

Figure 10: The HLA-FMI Federation.

5 CONCLUSION AND FUTURE WORKS

In Modeling and Simulation domain, many efforts are done to increase reusability of each technologies. Out of these technologies, Hight Level Architecture and Functional Mockup Interface are mainly used and try to go in this direction. In this paper we show that combining HLA based standard and Functional FMI is possible and can offer increase of interoperability. However, many aspects of hybrid federates can be improved. Considering a FMU needed in an HLA based DS, we will need access to its input and output over the federation. From that, at least one or more hybrid federate will be required to adapt a FMI to HLA context, and FOM files to must describes its interactions. According to this, we could automatically generate FOM files depending on the output and input of each FMU involved in the federation. Also, it could be possible to automatically generate a hybrid federate for each FMU needed in the DS.

REFERENCES

Al-Zoubi, Khaldoon, and Gabriel Wainer. 2009. "Performing Distributed Simulation with RESTful Web-Services." In *Winter Simulation Conference*, 1323–1334. Winter Simulation Conference.

Association, IEEE Standards. 2010. *1516–2010-IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)*.

Awais, Muhammad Usman, Peter Palensky, Atiyah Elsheikh, Edmund Widl, and Stifter Matthias. 2013. "The High Level Architecture RTI as a Master to the Functional Mock-up Interface Components." In *Computing, Networking and Communications (ICNC), 2013 International Conference On*, 315–320. IEEE.

Bastian, Jens, Christoph Clauß, Susann Wolf, and Peter Schneider. 2011. "Master for Co-Simulation Using FMI." In , 115–20. https://doi.org/10.3384/ecp11063115.

Benama, Youssef. 2014. "Supporting Make or Buy Decision for Reconfigurable Manufacturing System, in Multi-Site Context." *Ajaccio, France*, September, 150–58.

Blochwitz, Torsten. 2014. "Functional Mock-up Interface for Model Exchange and Co-Simulation." *July [Online] Https://Www. Fmi-Standard. Org/Downloads (Accessed January 2016)*.

Blochwitz, Torsten, Martin Otter, Johan Akesson, Martin Arnold, Christoph Clauss, Hilding Elmqvist, Markus Friedrich, Andreas Junghanns, Jakob Mauss, and Dietmar Neumerkel. 2012. "Functional Mockup Interface 2.0: The Standard for Tool Independent Exchange of Simulation Models." In *Proceedings of the 9th International MODELICA Conference; September 3-5; 2012; Munich; Germany*, 173–184. Linköping University Electronic Press.

Camus, Benjamin, Virginie Galtier, Mathieu Caujolle, Vincent Chevrier, Julien Vaubourg, Laurent Ciarletta, and Christine Bourjot. 2016. "Hybrid Co-Simulation of FMUs Using DEV&DESS in MECSYCO." In *Proceedings of the Symposium on Theory of Modeling & Simulation*, 8. Society for Computer Simulation International.

Falcone, Alberto, Alfredo Garro, Andrea D'Ambrogio, and Andrea Giglio. 2017. "Engineering Systems by Combining BPMN and HLA-Based Distributed Simulation." In *Systems Engineering Symposium (ISSE), 2017 IEEE International*, 1–6. IEEE.

"FMI." n.d. Accessed April 5, 2018. https://www.simulationx.com/simulation-software/experts/fmi-simulation.html.

Garro, Alfredo, and Alberto Falcone. 2015. "On the Integration of HLA and FMI for Supporting Interoperability and Reusability in Distributed Simulation." In *Proceedings of the Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium*, 9–16. Society for Computer Simulation International.

Kübler, Ralf, and Werner Schiehlen. 2000. "Two Methods of Simulator Coupling." *Mathematical and Computer Modelling of Dynamical Systems* 6 (2): 93–113.

Lee, D. 2010. "Development of Mediator-Based Direct Wokrflow Simulation System and HLA/RTI-Based Collaborative BPMS Middleware." PhD Thesis, Ph. D. Thesis, KAIST, South Korea.

Mittal, Saurabh, José L. Risco-Martín, and Bernard P. Zeigler. 2009. "DEVS/SOA: A Cross-Platform Framework for Net-Centric Modeling and Simulation in DEVS Unified Process." *Simulation* 85 (7): 419–450.

Mittal, Saurabh, José Luis Risco-Martín, and Bernard P. Zeigler. 2007. "DEVSML: Automating DEVS Execution over SOA towards Transparent Simulators." In *Proceedings of the 2007 Spring Simulation Multiconference-Volume 2*, 287–295. Society for Computer Simulation International.

Mustafee, Navonil, M'Hammed Sahnoun, Andi Smart, Phil Godsiff, David Baudry, and Anne Louis. 2015. "Investigating Execution Strategies for Hybrid Models Developed Using Multiple M&S Methodologies." In *Proceedings of the 48th Annual Simulation Symposium*, 78–85.

OMG, Business Process Modelling Notation. n.d. *Version 2.0, 2011*.

Piegay, Nicolas, and Denys Breysse. 2015. "Multi-Objective Optimization and Decision Aid for Spread Footing Design in Uncertain Environment." *Geotechnical Safety and Risk 5, Rotterdam, the Netherlands*, October, pp 419-424.

Pohlmann, Uwe, Wilhelm Schäfer, Hendrik Reddehase, Jens Röckemann, and Robert Wagner. 2012. "Generating Functional Mockup Units from Software Specifications." In *Proceedings of the 9th International MODELICA Conference; September 3-5; 2012; Munich; Germany*, 765–774. Linköping University Electronic Press.

Posse, Ernesto. 2015. "PapyrusRT: Modelling and Code Generation." *OSS4MDE 2015, Ottawa, Canada*, pp 54-63.

Powell, John, and Navonil Mustafee. 2014. "Soft or Approaches in Problem Formulation Stage of a Hybrid M&S Study." In *Proceedings of the 2014 Winter Simulation Conference*, 1664–75.

Rodney, Elodie. 2014. "Integrating Risks in Project Management." *16th International Dependency and Structure Modelling, Paris*, June, pp 419-424.

Sievert, Nicke. 2016. *Modelica Models in a Distributed Environment Using FMI and HLA*.

Yilmaz, Faruk, Umut Durak, Koray Taylan, and Halit Oğuztüzün. 2014. "Adapting Functional Mockup Units for HLA-Compliant Distributed Simulation." In *Proceedings of the 10 Th International Modelica Conference; March 10-12; 2014; Lund; Sweden*, 247–257. Linköping University Electronic Press.

Zacharewicz, Gregory, Olivier Labarthe, David Chen, and Bruno Vallespir. 2009. "Multi Agent/HLA Enterprise Interoperability (Short-Lived Ontology Based)." In *The International Workshop on Modelling & Applied Simulation Part Of*, 187–196.

AUTHOR BIOGRAPHIES

YOUSSEF BOUANAN is Postdoctoral researcher at University of Bordeaux. His research interests include M&S Theory, agent-based model and workflow. His email address is youssef.bouanan@u-bordeaux.fr.

SIMON GORECKI is Ph.D. Student at University of Bordeaux in IMS Lab. Domain research is about simulating process with distributed simulations and HLA (High Level Architecture). His email address is simon.gorecki@u-bordeaux.fr

JUDICAEL RIBAULT is a Ph.D. freelance software architect and associate researcher at University of Bordeaux and IMS Lab. His email address is judicael.ribault@u-bordeaux.fr.

GREGORY ZACHAREWICZ is Associate Professor HDR at University of Bordeaux and IMS Lab with both competences in enterprise engineering and M&S. His email address is gregory.zacharewicz@u-bordeaux.fr

Nicolas PERRY: Full Professor at ParisTech ENSAM of Bordeaux. Domain research is about system engineering, product process integration and green manufacturing. His email address is perry.nicolas@ensam.eu