# Integrating HLA-Based Distributed Simulation for Management Science and BPMN

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

# Integrating HLA-Based Distributed Simulation for Management Science and BPMN

**Simon Gorecki*, Youssef Bouanan*, Gregory Zacharewicz*, Judicael Ribault*, Nicolas Perry****

**IMS lab., University of Bordeaux, France*
*firstName.lastName@u-bordeaux.fr*
***Arts et métiers, ParisTech, ENSAM of Bordeaux, France*
*perry.nicolas@ensam.eu*

Abstract: Modeling and Simulation are becoming more and more complex, making their design very challenging. Modeling and Simulation of complex systems requires simultaneous consideration of several points of view and the study of these systems needs skills belonging to different scientific fields. In Distributed Simulation domain, IEEE 1516-2010 - High Level Architecture (HLA) is one of the most used standard. However, it does not provide any official graphical language for defining distributed simulation behaviors. Business Process Model and Notation standard could be an interesting solution for defining HLA execution scenario. This paper present application experiment applied to solar power plant. Our proposition consists in restrict the HLA execution process in order to use one federate as Master, controlling the others as Slaves. This allows us to generate a component responsible for the simulation execution process with parsing a Business Process Model and Notation diagram. In this paper, we present an application of this concept.

*Keywords: Modeling and Simulation; Distributed Simulation; High Level Architecture, Business Process Model and Notation, Interoperability.*

## 1. INTRODUCTION

The Modeling and 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 simulations (or models) from a complex system. Each autonomous simulation is executed on a different computer possibly geographically distributed from others. From a general point of view, this solution divides into modular simpler sub problems from complex problems, but also rises interoperability issues.

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 domains, and to integrate each domain models and tools (or subsystems) within frameworks of the M&S process.

In our case, we will use various M&S technologies in a semi-academic, semi-professional context: a French company has launched an innovative project to set up a solar power plant. This project deals with different domains including risks management. Most of these research works have created specific domain simulators. Each of these autonomous simulations can represent a fragment of the global project. One of the last phases of this project consists in coupling all these simulations to obtain a global simulation of the problem to treat. However, all of these simulations use different technologies and manipulate heterogeneous data. It complicates the coupling. As a response to these difficulties, we propose to use the High-Level Architecture (HLA) standard. Moreover, there is a special need of reusability in our context. Indeed, the company needs to reuse and run different components without any HLA knowledge. It is the issue we try to solve in this paper.

HLA is one of the most used standard specification software architecture for Distributed Simulation. However, the HLA standard does not provide any graphical language for defining the simulation scenario. This point could be interesting for the reusability problematic. In our case, the goal is to model and simulate business process (BP). A BP is a set of interrelated activities that are executed by one or more organizations working together to achieve a common business purpose (K L Ko 2009). Several modeling languages were introduced for defining industrials workflows, but Business Process Model and Notation (BPMN) is most widely adopted by users. The Business Process Model and Notation (BPMN) standard is a graphical notation for specifying business processes. It could be a viable solution to represent how Federates interact through a Federation. Using this standard as an HLA execution scenario will helps users to build simulation and insure execution traceability. This approach follows Model Driven Architecture (MDA) principles since it allows to divide conceptual modeling and execution in order to simplify work for people with low skills in IT and HLA.

However, the HLA standard doesn't provide any graphical language for defining the simulation scenario. This point could be interesting for the reusability problematic. The Business Process Model and Notation (BPMN) standard is a graphical notation for drawing business processes. It could be a viable solution for representing how Federates relate to each other, throw a Federation. The use of this standard will serve as an HLA execution scenario and help users to build simulation and assure execution traceability.

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 integration of BPMN standard with HLA standard. Section 4 discusses the detailed steps of the proposed framework and the reflection on how this approach can make the development of the HLA-based DS easier for Management Sciences. Finally, Section 5 is a conclusion of the paper.

## 2. BACKGROUND

This section introduces the different concepts used in the proposed framework for DS in management science.

### *2.1. Modeling and Simulation Interoperability*

From a M&S process perspective, distributed simulation implies dealing with different subsystems forming a coupled problem which are modeled and simulated in a distributed way. Indeed, the different domains of expertise may have different M&S tool, modeled and implemented in different languages. Besides, some of these tools must be available only on some specific hardware. Interoperability processes are then required to synchronize these heterogeneous tools and manage exchanges of data amongst them.

Distributed simulation technologies are a paradigm to model dynamic, heterogeneous, and spatial distributed systems. They not only aim at speeding up simulations, but also serve as strategic technologies for linking simulation components of various types (Chen et al. 2008). There are several approaches in the field of M&S offering interesting solutions to the challenges of the interoperability of simulation models and their execution on distributed computing environment. Two of the most popular efforts going in these directions are Functional Mock-up Interface (FMI) and. HLA is an IEEE standard (IEEE 1516) for distributed computer simulation systems (IEEE Computer Society 2010a). In the HLA standard a distributed simulation is called Federation (see Fig. 1). A Federation is composed of several HLA simulation entities, called Federate, which can interact among them by using the Run-Time Infrastructure (RTI). The RTI represents a backbone of a Federation execution and provides a set of services to manage the communications and data exchange among Federates. In this work, we adopted the HLA standard to simulate our system in a distributed environment.

### *2.2. High Level architecture (HLA)*

In the computer simulation domain, distributed simulation is one of the most useful approach 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 Simulatio**n (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 sensor. 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.

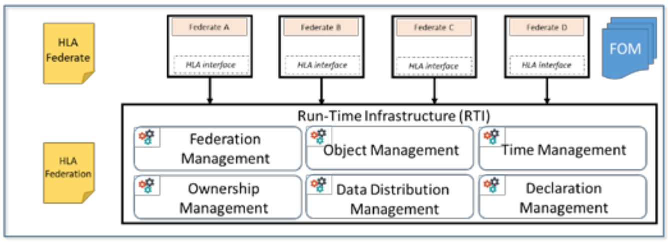


Fig. 1. HLA Architecture diagram.

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 HLA (IEEE Computer Society 2010b). It will support the specification of our software architecture.

### *2.3. Business Process Model and Notation (BPMN)*

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 2003) 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. Nowadays, BPMN becomes widely used by different organizations, as it is easy to learn, and powerful enough to depict the potential complexities of business processes (Sbayou et al. 2017).

## 3. STATE OF ART

As recalled previously, HLA is a well-known standard. It is described in different specification books, and it is used since 1996. One of them (HLA Object Model Template (OMT) (IEEE Computer Society 2010b) describes how interactions must be done between federates through mechanism of Subscribe (read) and Publish (write). Those communications are described by SOM and FOM XML files. However, there is no simple way to describe federate behavior inside the federation: each federate is autonomous. They can publish information, subscribe information, at any time. The only way to control them is to (re)develop them.

However, HLA components are still reusable according to the HLA Federation Development and Execution Process (FEDEP) (IEEE Computer Society 2010c) which has been recently integrated into the more general DSEEP (Distributed Simulation Engineering and Execution Process) standard). DSEEP uses seven-step process to guide the development of the simulation system as described:

- Defining Simulation Environment Objectives
- Perform Conceptual Analysis
- Design Simulation Environment
- Develop Simulation Environment
- Integrate and Test Simulation Environment
- Execute Simulation
- Analyze data and evaluate Results

Federates are autonomous, but must be reworked in order to be reused. This can be an advantage, but sometimes, in a high-reusability environment, this can be an issue. Several papers have already noticed the need of using graphical standard notation like BPMN in HLA domain. Despite this finding, there are few studies in the academic literature which have specified DS to simulate complex systems by graphically modeling the different component and their interaction of the DS project.

In industrial domain, visual interactive modeling environments are widely used. Taylor et al. (2008) find that, in DS domain, HLA is mainly used and the absence of graphical language can be problematic. Indeed, it can be easy to lose track regarding exactly what is wanted to occur between interoperating models. In that paper, the authors offered a set of interoperability reference models which can be used as guidelines to simplify interoperability process (Taylor, Turner, and Strassburger 2008).

Bocciarelli and al., presented the automated transformation of BPMN process into Extended Queuing Network (EQN) models, that are executed as distributed simulations in e-commerce scenario. This research works convert BPMN into executable code (Bocciarelli et al. 2012).

Falcone et al. discussed about the need of providing a standard graphical representation (BPMN) for managing HLA Simulation, in two aspects. The first aspect is to represent the definition on an HLA Federate through BPMN. The second aspect is to use BPMN in order to define and execute HLA Simulation. This last diagram could represent each steps of HLA federation life cycle (Falcone et al. 2017).

Bazoun et al. presented an Eclipse RPC tool named “SLMTOOLBOX” which is a graphic modeler, model transformer, and simulation engine, able to convert a BPMN diagram into DEVS models to simulate a process behavior (Bazoun et al. 2016).

All those papers have different goals, but they all demonstrate the importance of using BPMN as standardized graphic user interface in order to simplify the HLA development.

The aim of this paper is different than all the others because it proposes a new concept: the explicit specification of (and potentially restriction) of HLA execution process, in order to achieve that goal. The proposed approach uses one federate as Master, controlling the others as Slaves. The Master federate read a BPMN diagram given as in input, and orchestrate all federates like it is described by the BPMN diagram.

## 4. CONTRIBUTION

### *4.1. Towards a formalization of the execution process of the HLA-based distributed simulation*

The use of graphical modeling, in particular BPMN standard, to define and model part of distributed simulation process (Design Simulation Environment and execution scenario) 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, so that everyone in an organization can understand the objectives of a simulation; (3) execution traces of programs (Federation development) can be easily viewed and analyzed; (4) it creates a bridge which reduces the gap between the HLA code level and the corresponding BPMN processes (Falcone et al. 2017). Moreover, there is a need to improve the traceability execution process in HLA environment. Indeed, each federate is autonomous and it is not possible to anticipate (or plan) any execution scenario: each federate is evolving at its own pace and the final user cannot capture the overall behavior (or the HLA DS execution process). The use of BPMN standard to control execution process would be a solution for this issue. In addition, numerous BPMN tools can support the process of software development by offering a variety of editors and code generation components. This allows us to automatically generate the component responsible for the simulation process through a BPMN diagram.

The articulation of these two standards enables the automatic generation of executable code which can be packaged in global DS system. A second way would be to build the HLA federate component (master) which has the responsibility to orchestrate the steps of execution process through the interpretation of a BPMN diagram and controlling the data exchange between the Federation (Figure 2). This proposition enables an approach towards MDA, with the use of BPMN Model in the one hand, and the separated execution of it in the other hand.

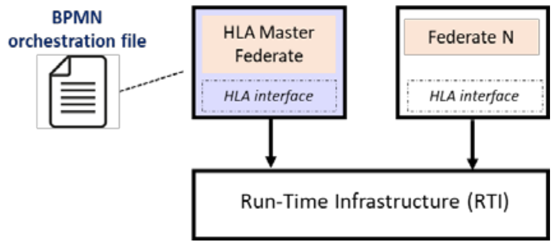


Fig. 2. Combining HLA-based Distributed Simulation and BPMN.

According to HLA standard, each federate can communicate with Object or Interaction class through Subscribe and Publish mechanism. Thanks to that, they can share simple and complex types of data. However, any document or method are provided by the standard to specify which Distributed Simulations are connected to which others. By using a BPMN diagram as DS orchestration, we can model federate execution scenario (Fig. 2) and interaction that they have with others entities (Fig. 3). In the Fig. 3, we propose to model HLA federate interactions with BPMN data associations flow. As we can see in the above figure, Federate A will publish an Object Class, which will be read by Federate B.

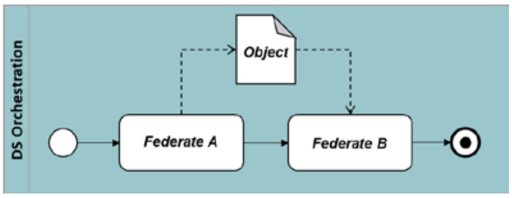


Fig. 3. Interaction between federate with BPMN data association flow diagram.

The author of the DS Orchestration document can also add BPMN comments in order to, precise which type of data are exchanged and link the FOM file.

### *4.2. Experiment*

From a technical point of view, after a few tests, we made an example for our proposition. Our study case is a solar power plant mobile factory (Gorecki, Zacharewicz, and Perry 2017). The goal of this simulation is to estimate production cost of a solar power plant in using several distributed simulations. As we can see on Fig. 4, we simulate the assembly of a power plant fragment. As represented in the BPMN DS Orchestration diagram, each assembly station is autonomous simulations, they are part of an HLA federation. These 5 assembling posts are using a single production line which is the solar panels chassis. Each simulation will add to the chassis a new part.

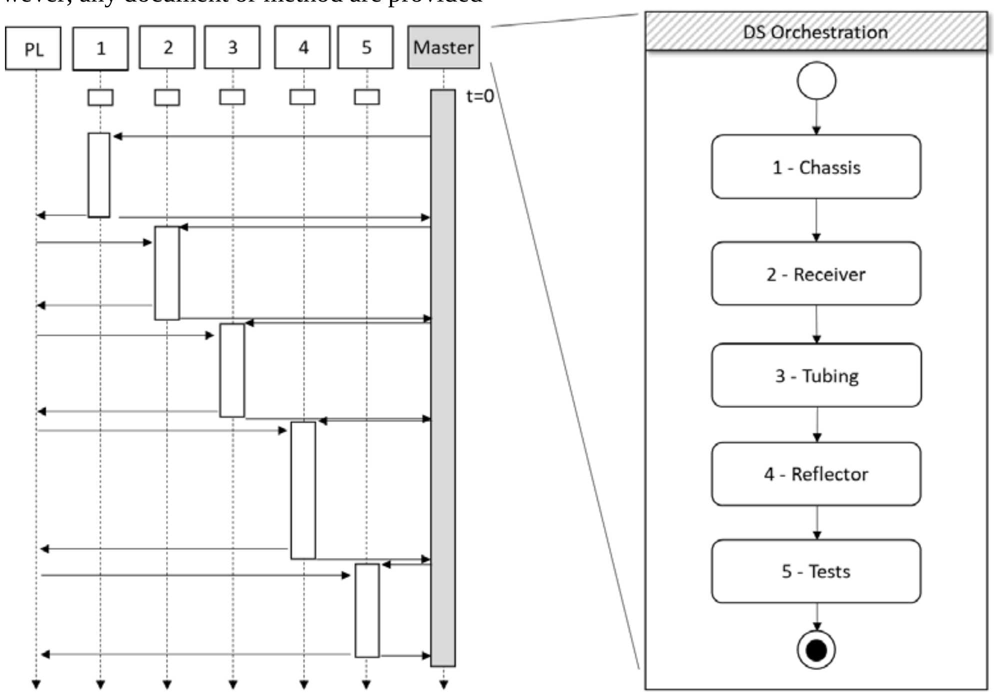


Fig. 4. BPMN and UML sequence flow diagram for solar panel production line.

The first simulation (1) bring to the production line a new solar panel chassis. It will be created depending on several parameters. This simulation also takes risks in account and can build a chassis with dimensions defect. Those conceptions errors are generated randomly. The second simulation is about the solar receiver (2), it place a solar reflector on the chassis in order to receive solar ray. The (3)is linked to the (2) it will consiste in place tube containing coolant liquid. The Reflector Simulation (4) concern assembling on the production line mirrors for reflecting solar ray in the solar receiver. Finally, the last simulation (5) check for all previous simulations errors. If one of them made a mistake, the car will be rejected. Otherwise, it will be expediate on the power plant.

During the simulation, all federates interact with Production Line (PL) which is represented by (Fig. 5) a MySQL Database.

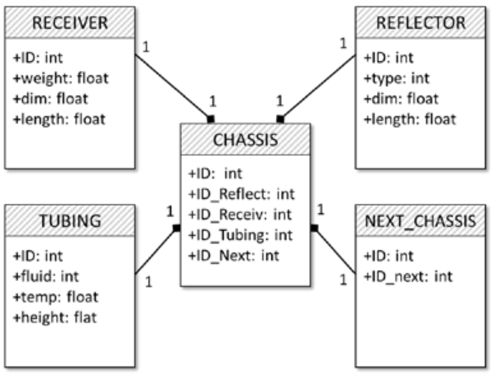


Fig. 5. Production Line (PL) UML class diagram.

Each Distributed Simulations read and write informations in that external entity representing the Production Line. During the execution course, each federate successively access to the database and write the result of their own simulations. In the final step, the last simulation will compare information written in database to the customer needs. If errors are detected, the car is rejected, otherwise, the product is expedited.

The graphic below (Fig. 6) describes technical architecture of the proposition according to 3 technical layers.

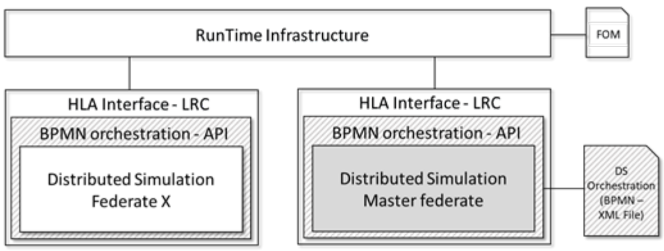


Fig. 6. Layering technical architecture.

The first layer (RTI, FOM, and HLA Interfaces) represent the HLA standard architecture, which is not modified. All HLA and RTI functionality are still available in this application (this allows us to say the HLA standard is still respected), but in a upper layer, we propose to reduce RTI functionality in order to simplify the scenario's definition step. All federates contain a first layer HLA Interface - LRC which implements Local RTI Component enabling communication with RTI, and we consider that all communication messages are described in the Federation Object Model (FOM) file.

The second layer (which correspond on Fig. 6 by cross hatch rectangles) represent an API which implements the master/slave mechanism described in the previous section. This layer will, from the Master federate point of view (gray rectangle), use publish and subscribe RTI functionalities to send: run, stop, and freeze orders to the federates (white rectangle), according to the DS Orchestration diagram. From the other federates point of view, this layer will freeze the local simulations at t = 0 (as described on Fig. 4) and put them in stand up status. Still from the slave federate's point of view, the API will be in charge to receive, run, stop and freeze orders from the master federate. Those orders will be applied on the local runtime simulation. The master federate will read the xml file that describes the BPMN diagram and address to each federate run, stop, freeze orders through the BPMN orchestration API layer.

This experiment allows us to demonstrate the reusability increase of our proposition. With the BPMN orchestration API, we can simply modify a simulation execution process with a graphical standard. In our example, the engineer can modeling federates behaviors through BPMN standard, and directly execute it without modifying code.

## 5. CONCLUSION

In distributed Simulation, IEEE HLA is a widely used standard because of its capacity to solve heterogeneous issues and/or its reusability potential. Federation enables to define a set of entities (federates) representing different solutions for a large problem. Literature reported that each federate is autonomous, have its own architecture but the design of federates' orchestration before simulation time is not easy an easy task. This papers is a contribution to facilitate the specification of federates communication making more explicit the publish and subscribe steps. In other words, it attempts to give a more clear and explicit support to describe the desired sequence of execution between the different federates. This contribution specifically finds its interest, with the complexity of the system to model, where designing a complete distributed simulation is a challenge. An example has been introduced to illustrate the contribution, a distributed simulation can be orchestrated with a BPMN diagram. We consider that the adoption of Business Process Modeling and Notation standard as graphical notation for defining HLA execution scenario will help non-specialists of M&S to define in a unified and standardized way the execution process of the simulation and also will facilitate DS execution process analysis after simulation run.

Future research efforts will be devoted to the definition of how communication (publish/subscribe) protocol are defined in our proposition. The BPMN Orchestration will use message flow to indicate publish and subscribe exchanges but must be developed to precise the category of data to be exchanged for instance.

## 6. REFERENCES

Bazoun, Hassan, Judicael Ribault, Gregory Zacharewicz, Yves Ducq, and Hadrien Boyé. 2016. "SLMTOOLBOX: Enterprise Service Process

Modeling And Simulation By Coupling DEVS And Services Workflow." *International Journal of Simulation and Process Modelling, Inderscience,* May.

Bocciarelli, Paolo, Daniele Gianni, Alessandra Pieroni, and Andrea D'Ambrogio. 2012. "A Model-Driven Method for Building Distributed Simulation Systems from Business Process Models." *Simulation Conference (WSC), Proceedings of Winter, Berlin, Germany*, December, 1–12.

Chen, David, J. Stephen Turner, Wentong Cai, and Muzhou Xiong. 2008. "A Decoupled Federate Architecture for High Level Architecture-Based Distributed Simulation." *Journal of Parallel and Distributed Computing* 68 (November): 1487–1503.

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

Gorecki, Simon, Gregory Zacharewicz, and Nicolas Perry. 2017. "Using High Level Architecture to Combine Simulations in Company Context Mobile Factory." *I3M, Barcelona*, September, 281–90.

IEEE Computer Society. 2010a. *1516.2-2010 - IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-- Object Model Template (OMT) Specification*.

IEEE Computer Society. 2010b. *1516-2010 - IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-- Framework and Rules*.

IEEE Computer Society. 2010c. *IEEE Standard 1516-2010 for M&S - HLA - Framework and Rules*. http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5553440.

K L Ko, Ryan. 2009. "A Computer Scientist's Introductory Guide to Business Process Management (BPM)." *Crossroads*, June 2009.

Sbayou, Mariem, Youssef Bouanan, Gregory Zacharewicz, and Juline Francois. 2017. "Agent Based Modeling Architecture with BPMN and DEVS Network." *I3M, Barcelona*, September, 391–98.

Taylor, Simon J.E, J. Stephen Turner, and Steffen Strassburger. 2008. "Guidelines for Commercial Off-the-Shelf Simulation Package Interoperability." *Winter Simulation Conference, Miami, USA*, December, 193–204.

Zacharewicz, Gregory, Olivier Labarthe, David Chen, and Bruno Vallespir. 2009. "Multi Agent HLA Entreprise Interoperability (Short-Lived Ontology Based)." *The International Workshop on Modelling & Applied Simulation, Puerto de La Cruz, Spain*, September, 187–96.