



HAL
open science

BPMN Modeling for hla based simulation and visualization

Simon Gorecki, Youssef Bouanan, Grégory Zacharewicz, Nicolas Perry

► **To cite this version:**

Simon Gorecki, Youssef Bouanan, Grégory Zacharewicz, Nicolas Perry. BPMN Modeling for hla based simulation and visualization. Society for Modeling and Simulation - SpringSim-Mod4Sim 2018, Apr 2018, Baltimore, United States. pp.1-12. hal-02136421

HAL Id: hal-02136421

<https://hal.science/hal-02136421v1>

Submitted on 5 Jun 2019

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

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



Science Arts & Métiers (SAM)

is an open access repository that collects the work of Arts et Métiers ParisTech researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: <https://sam.ensam.eu>
Handle ID: <http://hdl.handle.net/10985/14997>

To cite this version :

Simon GORECKI, Youssef BOUANAN, Gregory ZACHAREWICZ, Nicolas PERRY - BPMN Modeling for hla based simulation and visualization - In: Society for Modeling and Simulation - SpringSim-Mod4Sim 2018, Etats-Unis, 2018-04-01 - SPRINGSIM 2018 Mod4SI - 2018

Any correspondence concerning this service should be sent to the repository
Administrator : archiveouverte@ensam.eu



BPMN MODELING FOR HLA BASED SIMULATION AND VISUALIZATION

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

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

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, France
perry.nicolas@ensam.eu

ABSTRACT

Modeling and Simulation (M&S) is attempting to tackle more and more complex systems, which makes its design highly challenging. Complex systems' M&S requires the consideration of several simultaneous points of view and involve skills from different scientific and technical fields. Distributed Simulation domain answers the question of coupling and running together heterogeneous components, e.g. IEEE 1516-2010 - High Level Architecture is one of the most used standard. However, none provides any official or at least recognized user-friendly (e.g. graphical) language to specify distributed simulation desired behavior. Due to its capacity to represent behavior of processes, Business Process Model and Notation (BPMN) standard could be an interesting solution for defining HLA execution scenario within a Model Driven Architecture approach. This paper aims to support the modeling phase of the HLA execution process in order to explicitly design the desired steps of orchestration between distributed HLA federates through the interpretation of a business process diagram.

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 into modular simpler sub problems from complex 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.

In our case, we will use various modeling and simulation 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 relate to each other, 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.

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 modeling and simulation 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 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).

- HLA is an IEEE standard (IEEE Computer Society 2010a) for distributed computer simulation systems. 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 among 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 among Federates.
- FMI (Functional Mock-Up Interface) [Functional mockup interface 2.0: The standard for tool independent exchange of simulation models] establishes itself as a standard for model exchange and co-simulation of equational models. 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).

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 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 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.

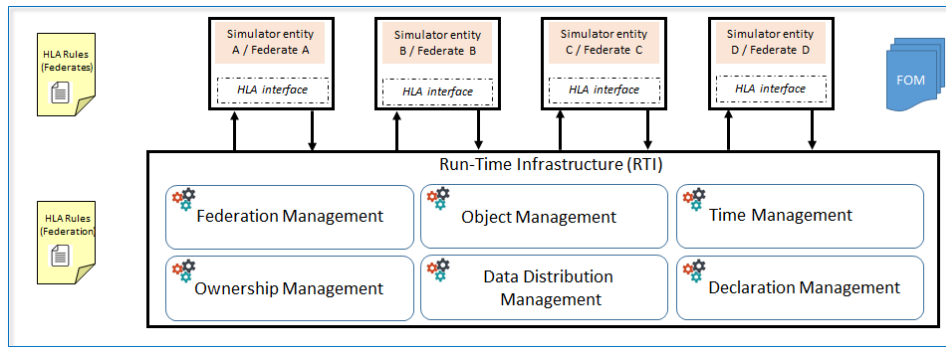


Figure 1: HLA Architecture diagram.

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.

2.3 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 are 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 used a lot 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 HLA-based distributed simulation execution process

The proposed DS execution process formalization is derived by combining methods and concepts used as guidelines for conducting production engineering and DS projects. Efforts on providing guidelines for distributed simulation which use the HLA allow the development of the IEEE Recommended Practice for HLA-based Distributed Simulation Engineering and Execution Process (DSEEP) standard (IEEE Computer Society 2003). DSEEP is a revision of the Federation Development and Execution Process (FEDEP) developed by ISO and Standardized by the IEEE. It recommends general practice for developing DS using HLA. DSEEP is a generalized framework which can be adapted to specific organizational needs and individual applications. However, the process for DS (DSEEP) does not provide a well-defined graphical representation for specifying how the simulation component (Federate) are connected with each other, and how they react on user requests to subscribe/publish appropriate data and use time management mechanism if necessary.

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, making it semi-automatically. 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 .

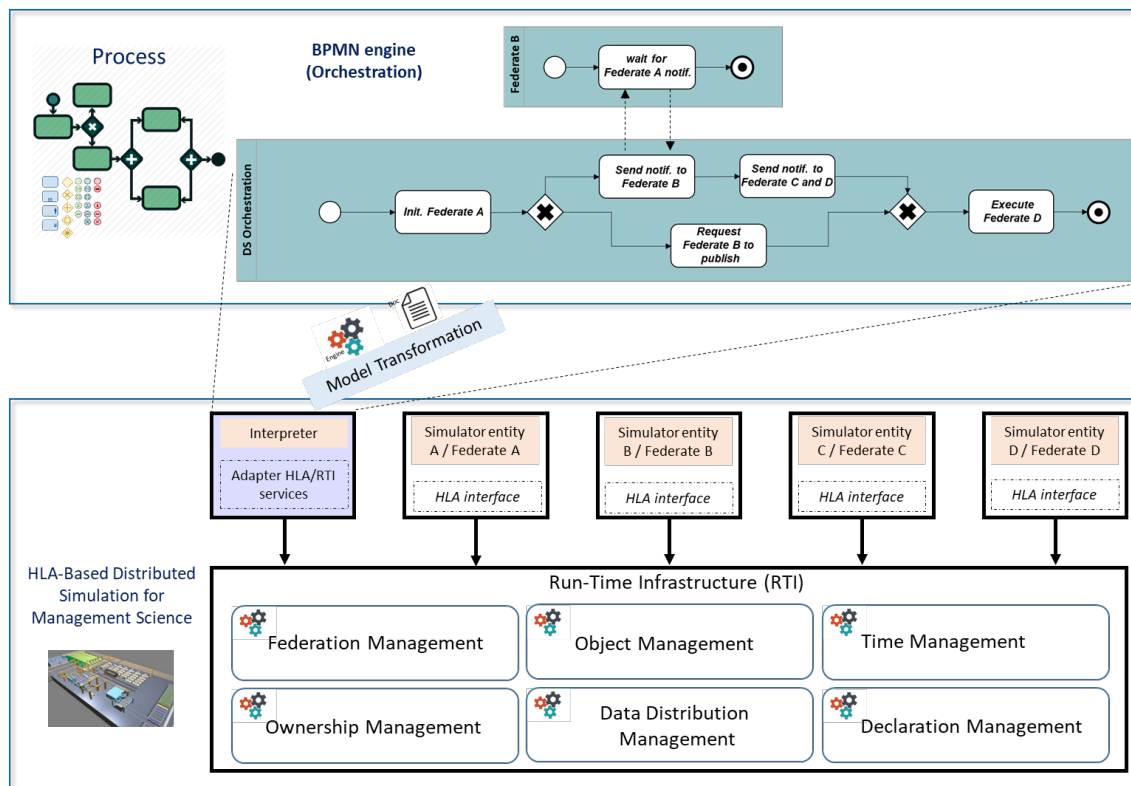


Figure 2: Combining HLA-based Distributed Simulation and BPMN.

4.2 Technical application

From a technical point of view, the implementation of this concept was done in a first time on a simple example in order to test our proposition. The Figure 3 describes technical architecture of the proposition according to 3 layers.

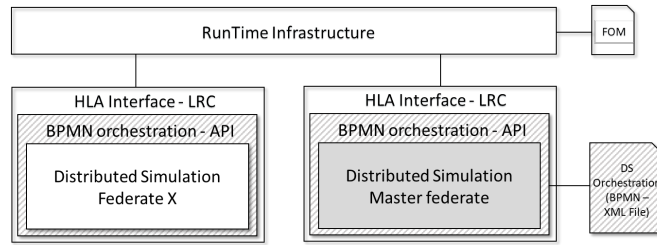


Figure 3: Layering technical architecture.

Figure 3 shows several technical layers. 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 figure 3 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 Figure 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.

We can compare this architecture to TCP/IP protocol (Fall and Richard 2011): to ensure effective communication between federates, each layer need and use next and previous layers. In our case, the highest layer which represent distributed simulations (the two Distributed Simulations on Figure 3) can be compared to *Application layer* in TCP/IP protocol. The BPMN orchestration layer (in cross hatch) can be associated to the *transport layer*, HLA interfaces are like the *Internet layer*, and the HLA functionalities provided by RTI can be associated to *Network Access Layer*.

After implementation, we can observe on the left side of Figure 4, the UML sequence flow diagram representing the federation life cycle. This diagram respects what is described in the DS Orchestration BPMN Diagram on the right side of Figure 4.

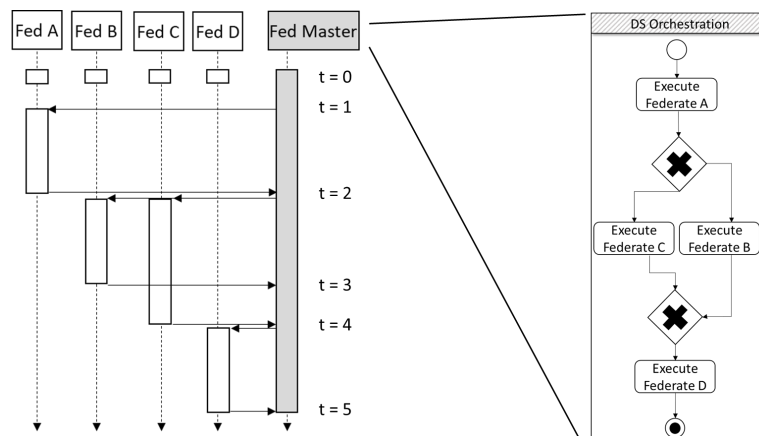


Figure 4: BPMN and UML sequence flow diagram of test federation.

In our example, each simple federate makes simple actions: write a message, sleep a few seconds, and write a second message after its end. At the beginning of the program (time step $t = 0$), each basic federate publishes and subscribes to the master variables through RTI Component and waits for master federate instructions. At the same time, the master federate reads the BPMN file in order to identify the name and numbers of federate during the global simulation. When all federates are mapped, the master will send a message to the first federate according to the DS Orchestration file ($t = 1$).

Federate A, is now triggered by this notification and execute its algorithm. When the Federate A done ($t = 2$), its BPMN orchestration layer send message to master federate to indicate the end of work. The master can now go to the next step and trigger federate C and B. At the BPMN merge point, the master federate know it had to wait federate C and B end process ($t = 3$ & 4) to execute last simulation, etc. At the end of the process ($t = 5$), the master federate detects an endpoint, this will cause the federation destruction and simulation end.

4.3 Using the BPMN simulation execution process diagram for 3D animation scenario

Another interesting point with High Level Architecture based Simulation is 3D visualization. This point has already been mentioned in different research works for representing federates evolution within a distributed simulation (Falcone, Garro, and Spadafora 2014). Researchers has developed a HLA 3D renderer engine federate able to represent federates evolving in federation without the need of programming each scenario steps directly in the engine. In our study case, a 3D visualization can be done by an analysis from the engine of an events chain, where events are described by the BPMN Distributed Simulation process representing the sequence of distributed simulations results. We are proposing a 3D engine federate which can read and interpret BPMN simulation execution process diagram. This federate also takes in input several needed data about graphic engine like 3D models, data about each models it has to represent, and parameters about the simulation base. The main objective of this federate is also to graphically transpose Distributed Simulation execution scenario in order to ensure the execution traceability. With this method, we can have a simulation scenario representation. To do so, the 3D engine federate communicates with the “interpreter” federate presented in the last chapter and exchange information about federation activity during the simulation.

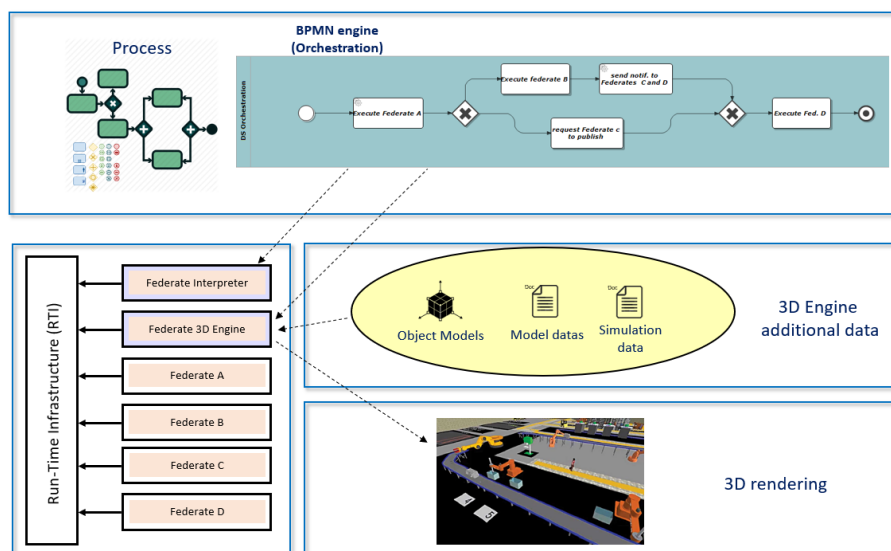


Figure 5: 3D Engine federate using BPMN simulation execution process diagram in federation.

4.4 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 (El Amine 2016).
- 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 standard to create a general middleware in order to simulate a whole system.

4.5 Concept applied on management science

Those simulations occur in highly heterogeneous environment. Each one of them uses several technologies and manipulates different types of data. They are autonomous and complementary, so they cannot be merged in only one application. Moreover, there is a reusability need where each simulation is used as stand-alone application. To solve these problems, HLA standard is appropriate because it defines each one of them as federate (Gorecki, Zacharewicz, and Perry 2017).

Each simulation is a component thanks to the HLA specification and enables the orchestration of a “global simulation” as Figure 4. With HLA allowing us for creating and managing independent simulated components, the global solar panels project is set as a federation, composed of each simulator that we have just described (as federate).

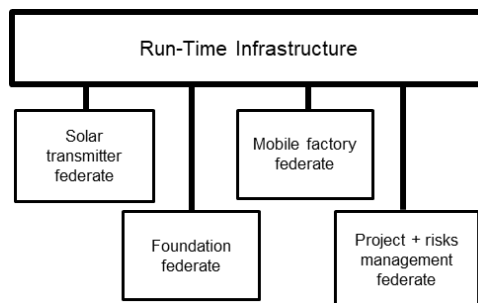


Figure 6: Standard HLA architecture.

Moreover, the company has a high need in reusability of this components. Indeed, each one of them would be used as standalone simulation, or as member of a custom federation. Execution scenario

customization has to be the easiest possible. Furthermore, this scenario customization must be easy to control because it is intended for non-engineer's people.

To do so, the company is building a graphic modeler tool which can easily generate Business Process Model and Notation under XML syntax.

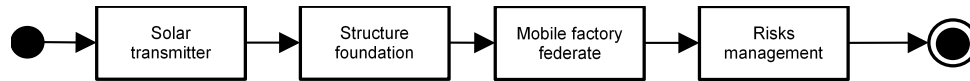


Figure 7: BPMN execution scenario for global simulation.

To get this behavior, we offer to setup a third federate with the role of “Master” federate, or “extended RTI agent”. It can control (through subscribe and publish mechanics) all the other “Slaves” federates. For this, we need to complete several points:

- Each slave federates of the federation must be totally autonomous.
- Each slave federates must have an overlay allowing them for being dependent from the master federate. They subscribe to triggers in the master federate, which allows them to be started remotely. During the execution time, each federate is able to communicate only with the master federate, so they must publish data into the master.

This overlay sets federates as slaves depending on the master federate. Then, the simulation scenario depends on orders sent by the master to each federates. The last step is to use BPMN diagram in order to control executions orders. For that, the graphic BPMN designer can generate UML code describing HLA scenario needed as Figure 4. The objective is to link the BPMN standard and HLA Master federate. This can be done in two different ways:

- The first way consists in implementing a UML interpreter into the master federate, able to read the UML file and convert it into HLA subscribe/publish mechanism to execute the simulation scenario described by BPMN graphic (full automatic solution).
- The second way consists in implementing into the graphic BPMN modeler an extension able to generate code, containing the master federate according to the BPMN description. This code should be packaged into HLA federation to be executed (semi-automatic solution).

A second need for the company is the necessity of getting a 3D rendering of the simulation. To do so, the second proposition of this paper is suitable. By defining the BPMN simulation execution process and 3D objects model, the 3D engine federate is able to create a 3D render of the factory installation.

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 first 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. 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 a graphical representation, potentially extended from with BPMN notations, for specifying the behavior of HLA-based simulation component (Federate) in terms of actions. It will also include the specification of communication (publish/subscribe) protocol to be defined. 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. In future works, the framework could use a library to interpret the XML file. Finally, we will investigate how to combine HLA standard and FMI standard.

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.
- Benama, Youssef. 2014. "Supporting Make or Buy Decision for Reconfigurable Manufacturing System, in Multi-Site Context." *Ajaccio, France*, September, 150–58.
- Blochwitz, Torsten, Martin Otter, Johan Åkesson, Martin Arnold, Christoph Clauss, Hilding Elmqvist, Markus Fredrich, et al. 2012. "Functional Mockup Interface 2.0: The Standard for Tool Independent Exchange of Simulation Models." *Proceedings of the 9th International Modelica Conference, Munich, Germany*, January, 173–84.
- 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.
- 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." *Symposium on Theory of Modeling & Simulation, Pasadena, USA*, January, 1–8.
- 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.
- El Amine, Mehdi. 2016. "Integration of Concept Maturity in Decision-Making for Engineering Design: An Application to a Solar Collector Development." *Springer-Verlag, London*, October, 235–50.
- 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.
- Falcone, Alberto, Alfredo Garro, and Francisco Spadafora. 2014. "Simulation Exploration Experience: A Communication System and a 3D Real Time Visualization for a Moon Base Simulated Scenario." *Distributed Simulation and Real Time Applications (DS-RT), Toulouse France*, November, 113–20.
- Fall, Kevin, and Stevens Richard. 2011. *TCP/IP Illustrated, Volume 1: The Protocols*. Addison-Wesley.
- 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. 2003. *IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process*.
- 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.
- 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, 419–24.
- Posse, Ernesto. 2015. "PapyrusRT: Modelling and Code Generation." *OSS4MDE 2015, Ottawa, Canada*, 54–63.
- Rodney, Elodie. 2014. "Integrating Risks in Project Management." *16th International Dependency and Structure Modelling, Paris*, June, 419–24.
- 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 Enterprise Interoperability (Short-Lived Ontology Based)." *The International Workshop on Modelling & Applied Simulation, Puerto de La Cruz, Spain*, September, 187–96.

AUTHOR BIOGRAPHIES

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

YOUSSEF BOUANAN is Postdoctoral researcher at University of Bordeaux. He received his Ph.D. degree in Production Engineering from University of Bordeaux, France. His research interests include M&S theory, agent-based model and workflow. His email address is youssef.bouanan@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 is Full Professor at ParisTech ENSAM of Bordeaux. Domain research is about system engineering, product process integration and green manufacturing. His email address is nicolas.perry@ensam.eu