USING HIGH LEVEL ARCHITECTURE TO COMBINE SIMULATIONS IN A COMPANY CONTEXT: MOBILE FACTOR

Simon Gorecki, Gregory Zacharewicz, Nicolas Perry

USING HIGH LEVEL ARCHITECTURE TO COMBINE SIMULATIONS IN A COMPANY CONTEXT: MOBILE FACTOR

[a]Simon GORECKI, [b]Gregory ZACHAREWICZ, [c]Nicolas PERRY

[a], [b]Integration from Material to Systems lab, University of Bordeaux, France
[c]Arts et métiers, ParisTech, ENSAM of Bordeaux, France

[a]simon.gorecki@u-bordeaux.fr, [b]gregory.zacharewicz@u-bordeaux.fr
[c]nicolas.perry@ensam.eu

ABSTRACT
Modeling and simulating play an important role in the industry, and even more in the innovative domain, for testing scenarios by anticipating different eventualities. Those simulations are often very complex and difficult to modify, that is why they could be built separately first and then run together. One of the most popular standards for distributed simulations is the IEEE 1516-2010-Evolved of the High Level Architecture (HLA) that supports implementation of distributed simulations.
In our context, a company has launched a project of mobile factory to set up solar panel field in several countries. That implies various study domains and expertise (foundation definition, study of solar transmitters structure, risk analysis, project management, factory modeling). The goal of this paper is to demonstrate the interconnections proposed between all these entities. It will be implemented in the frame of a global simulation managed by HLA.

Keywords: High Level Architecture; Mobile Factory

1. INTRODUCTION

In industrial design domain, the modeling and simulation tools are very important. They allow the representation of any complex system, to study their behavior and their interactions with the environment. Once the modeling phase is completed, the simulation of these systems will allow us to virtually design our subject to anticipate and avoid problems during the building phase.
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 for setting up a solar power plant. This project deals with different domains including risks. University of Bordeaux supports research in M&S and specifically in DS (Distributed Simulation). Most of these research works have created specific domain simulators. Each of these autonomous simulations is capable of representing a fragment of the global project. One of the last phases of this project is to assemble all these simulations to obtain a global simulation of the problem. However, all these simulations use different technologies and manipulate heterogeneous data, which complicates the assembly. To solve these problems, we will use the HLA standard, and especially the Pitch Technology. (Möller, 2012)

2. DISTRIBUTED SIMULATION & HIGH LEVEL ARCHITECTURE

In the computer simulation domain, distributed simulations are one of the most useful and powerful applications. Indeed, they consist of several components (often associated with one or more functions) that can be processed by different processes. All these components are part of a single execution which can be relocated to a different computer / server, hence the term "distributed". This concept of functions relocation makes the loads distribution possible on different machines, and thus increase the efficiency of a program.
One of the advantages of distributed simulation is to solve some interoperability problems. Interoperability is the interactions ability between systems. This problematic appears when several highly dissimilar systems (by their internal structure, exchanged data format, or semantic data) have to communicate. The interoperability problematic must be considered if interactions are at data level, service level or process level (Zacharewicz, Labarthe, Chen, & Vallespir, 2009). Those problematics involve at least two entities which try to communicate. Consequently, establishing interoperability consist in relating two systems together and remove incompatibilities. Incompatibility is the fundamental concept of interoperability. The concept of 'incompatibility' has a broad sense and is not only limited to 'technical' aspect - as usually considered in software engineering, - but also 'information' and 'organisation' (Zacharewicz, Chen, & Vallespir, 2009).
Indeed, in distributed simulations, the components are modular. They can have a heterogeneous architecture and exchange different structured messages. This enables the solution of interoperability problems.
In our application case, this notion of distributed simulation will be used with the High Level Architecture standard (HLA) (IEEE Computer Society, 2010). It is a specification of software architecture. It 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 simulations. This distributed simulations 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 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 technology is the synchronization. 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. In order to do this, various systems for synchronization of processes and time management are proposed by HLA.
According to the HLA standards, each simulation participating to the application is called "federate". A federate interacts with other federates. There are forming a group named HLA federation. All of these entities can communicate with each other through a Run-Time Infrastructure (RTI). It is the RTI that will manage the federation, authorize federates to communicate or not, and provide various services such as time management, file or data exchange, etc.

3. RUN-TIME INFRASTRUCTURE (RTI)

A federation is composed of a set of federates and a Run-Time Infrastructure (Falcone et al., 2015). This RTI provides to federates all functionalities which are described by the specification. Federates can only interact through the RTI. They can "Publish" to inform the RTI and the other federates about an intention to send information. They also can "Subscribe" to reflect some information created and updated by other federates. This is the basic communication mechanic. The data flow exchanged between all the federates is represented in the same form of classical object-oriented programming. There are two kind of objects which are exchanged in HLA standard: Object Class and Interaction Class. (Möller, 2012)
Object Class are time persistent during the simulation. They have attributes that can be updated. For instance, in the case of a simulation where the Object Class would be a car, his attributes would be a position, a speed, a name. Interaction are not persistent over time and can have parameters. For example, in this case, "Start car", "Stop car", "Accident" would be possible Interaction.
This two kind of objects are described by a XML file named Federation Object Model (FOM) attached to a federate. That is an important point: the FOM describes all the information that will be exchanged between federates. It is the only thing which will be shared during the simulations. This has an impact on the safety of this technology. The federate are totally autonomous, the only exchange between them is described by the XML file.
Each federate are single. Originally, HLA was created for reusability, all the components can be executed separately, or with others through a federation.

4. BUSINESS CONTEXT

The company behind this work designs solar power plants. This project consists in installing solar panels fields in several countries in order to provide electricity in world areas which are not powered so far. However, the transport of solar panels fields is extremely expensive. To reduce these blows, we aim to design a mobile factory which would manufacture the solar panels on site, and thus reduce the transport cost. Rather than transporting finished products, only the mobile plant and raw materials will be carried out. This project's main challenges are the miniaturization of this factory in order to get it in the least transport containers (around 20), and the factory designing taking into account risks and low knowledge on this project.
In terms of implementation and difficulties caused by the project complexity, this innovation gave rise to several works which subjects deal with the following problems:

- Optimization and decision helping for defining the structure foundations of the solar panels field, depending on the ground structure. Simulation executed with Mathlab and Excel. (Piegay & Breysse, 2015)
- Study of the concept maturity integration in decision making process: applied for designing the solar transmiter supporting structure. Simulation executed with Mathlab and Excel. (El Amine, 2016)
- Study and dimensioning of the mobile factory dimensions, cost, etc, according to the demand. Simulation executed with Excel. (Benama, 2014)
- Study of project management method integrating risks. Calculating risks probabilities into project management Simulation executed with Excel. (Rodney, 2014)
- Tool to concatenate all the mentionned works to make them run under Papyrus engine: BPMN for modeling and simulate systems. (Posse, 2015)

Most of these works have led to a simulator creation which solve a specific domain problem submitted to it. It is now necessary to allow them to communicate for obtaining an overall simulation representing the whole solar panel field.

5. PROPOSITON

In order to overcome the interoperability problems, we will link the different components represented in the next scheme thanks to the HLA specification.
This allows the orchestration of our simulations set communicating within "global simulation". With HLA, allowing us to create and manage independent simulated

components, the global solar panels project will be set as a federation, composed of each simulator that we just describe (as federate).

In our federation, the Run-Time Infrastructure (RTI) provides different mechanisms to manage federate and federation. It will administer data exchange between federates, allow connection authorization to federation, time management and simulations synchronization.

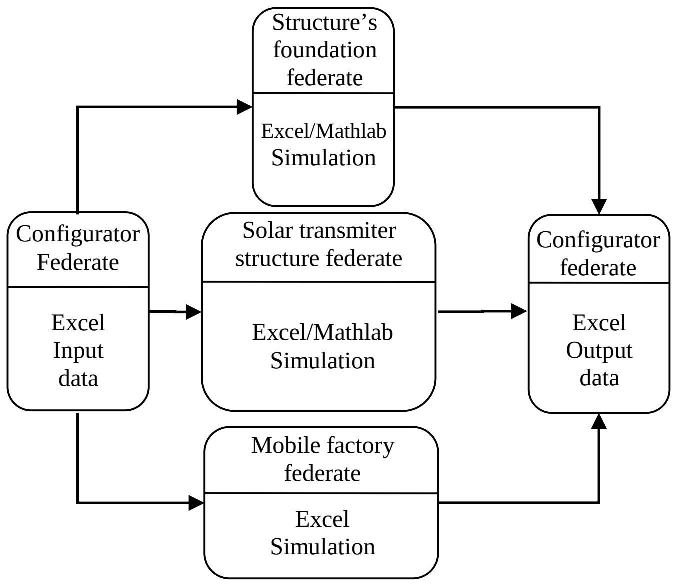

Figure 1: Business Process Model – Choreographies Diagram Mobile factory

We can see on the diagram above all the simulators organized in functional blocks connected according to the flow of the necessary flows. For the project management simulator with risk taking, it must be linked to each function in order to be able to generate untimely events that correspond to technical problems during the simulation.

In this context, the interoperability issues are strong on the one hand according to the data handled by each of the simulators, and on the other hand according to the various technologies used (Excel and Mathlab).

6. CONFIGURATOR

The starting point of all this simulations is the 'configurator'. It is used as a central point where all process will draw data, perform calculations, and store back their results. It acts as a data base module where each simulation will read and write values. No calculation are made in the part. There are 3 data modes managed by the configurator:

- Input data: will be filled by the user who defines the simulation's parameters.
- Intermediate variables: will be filled and used by several simulations in order to communicate between them.
- Output data: will contains all simulation's results.

Among the "input data" parameters, there are three different data types:

- Design variables: are parameters which must be optimized. This means we can act on those values in order to research the best compromise between all the goals. Each design variables are associated to a range of values which will vary until we found the right value.
- Project variables: are imposed by the project specifications. They depend on the system definition and strategic decision taken by the decision maker. Contrary to the design variables, we cannot act on these values. During the preliminary conception phase, the decision maker can be imprecise on some strategic choices, so, all the project variables are not every time constant, they can evolve.
- Environmental variables: describe natural properties (climatic, air, water, ground…) of the system environment, and also the social, economic and political environment.

The "intermediate variables" part has the role of link between simulations. It allows them to communicate. Any simulation can write in this field a value, a name and a unit which will be available for others programs.

Finally, the "output data" part is the global simulation final step. The user gave information in "input data", and after the run, he takes the results back in "output data". There are several output data types allocated, one for each simulator:

- Mobile factory
- Structure's foundation
- Solar transmitter

The figure below represents the data flow exchanged between the configurator and the three other simulations according to the (Yourdon & DeMarco, 1989) notation.

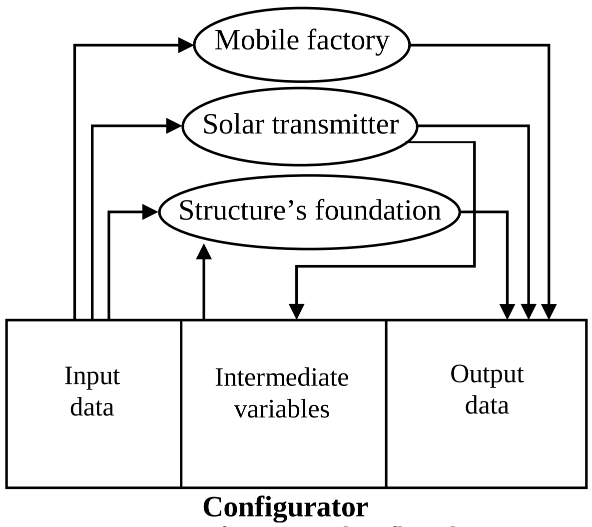

Figure 2. Configurator's data flow diagram

We can see that the “structure’s foundation” simulation need in input data from, “input data” and “intermediate variables”. This means that data from “intermediate variables” must be completed before the “structure’s foundation” runs. In order to fill this “intermediate variable” database, that we can see on the figure 2, we must run the “Solar transmitter” simulation first. This execution will complete the missing field. After that, The Structure’s foundation simulation followed by the mobile factory simulation can be executed. Those three simulations will complete the “output data” part and give the user the simulation’s result.
The figure 3 describes the simulation launch scenario that we have just described.

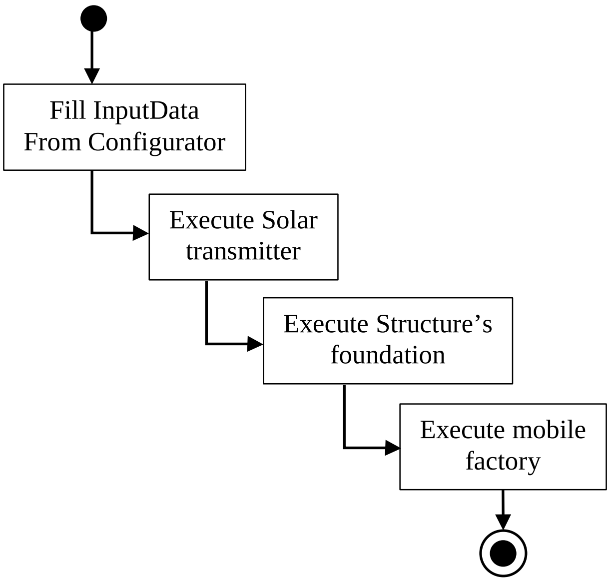

Figure 3. Simulation launch scenario

From an HLA point of view, the configurator could be a federate that:

- Contains input, intermediate, and output data we have just described.
- Distributes this data to each federate when they need them.
- Verifies data values.
- Manages the simulation launch scenario.

7. SOLAR TRANSMITTER

This simulation has for objective to determinate which parameters are the most efficient for the solar transmitter system. The presence of immature concepts makes decision-making difficult in this development phase. Furthermore decisions taken during this conceptual design phases have a critical impact on the product life cycle cost. To deal with issue, a simulation has been developed in order to anticipate conceptual problems (El Amine, 2016). As pictured in the figure 4, this simulation is composed of several Mathlab functions that use configurator’s data in order to define the best solar transmitter’s parameters considering the constraints imposed by specifications. This simulation also determinates intermediate variables that are required for the structure’s foundation simulation. That is why this simulation must be executed first.

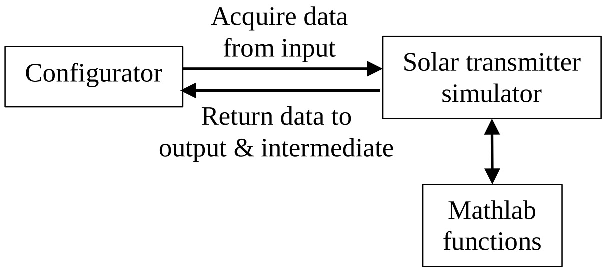

Figure 4. solar transmitter simulation architecture

8. STRUCTURE FOUNDATIONS SIMULATION

This simulation is about defining structure foundation of the solar field, and defining steel frame according to the specifications (Piegay & Breysse, 2015). The development of those structures is complex because it must respect contradictory instructions from the decision-maker. Thus, there is a need to find the best compromise between safety requirements, and economic stakes. Variability of environmental parameters and uncertain acceptance threshold are incorporated in a decision process simulation. The responses of the system can then be characterized by a performance measurement and by a dispersion measurement, making possible the judgment of the solution robustness while respecting the probabilistic character of the input data.
The effects of the interaction between ground and structure are also considered for achieving an overall structure optimization. When the simulation is over, we will have several parameters in output that give information about the most optimized structure foundation and steel frame.
For this simulation, we deal with two data sources. The first is the configurator described before. It will draw data from “input data” and "Intermediate variables” (which has just been filled by the Solar transmitter simulation) to store them in its own Excel file. The second data source is another Excel file that contains technical information about different sorts of metal beams (it is named “profile”). Those two databases will be used by two calculators: one in Excel file, it calculates structure’s foundations parameters, stock and share all simulation results to the configurator. The second calculator is a set of Mathlab functions which use the two databases for calculating the solar field steel frame. At the end of the simulation, data are brought back in the first calculator as you can see on the figure 5.

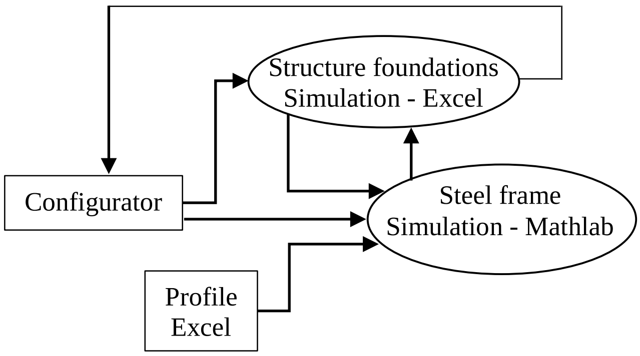

Figure 5. Structure foundations + steel frame simulation architecture – dataflow diagram

In the solar panel field system, the "Profile" database that we have just mentionned is not used by any other simulation. Therefore, it can be added to the structure foundations block. This set of two simulations plus one database could form an HLA federate. This federate has the role of simulate and defines the solar field structure foundation and defines the steed frame specification.

9. MOBILE FACTORY SIMULATION

This mobile factory simulation concerns strategic decision making between making or buying materials. In France, industrial production is coming to grief: the country exports few manufactured products. In addition, some companies have undertaken the relocation of their production and getting closer from the final customers in order to increase their margin. When a customer needs a production in a limited time, the mobile production system (MPS) can be the solution. The concept of MPS consists in mobilizing the same production system to successively satisfy several orders from geographically dispersed customers, directly on the final customer's site. (Benama, 2014)

This simulation is executed on a single Excel file. It is a system which takes in input data from the configurator, and have also its own data input. Those other data can be divided in two categories:

- Workforce variables: employees numbers available on site, teams numbers, managers numbers, break time of teams, etc…
- Technical variables: mirrors numbers per palette, palettes number per container, field surface, reflector number, etc…

As for the previous simulation, this one has its own private database which contains information about each technical operation of the solar field factory. For every construction's step of the field, we have got, on this database, information about costs in time, human, and material resources. This Excel sheet will be used as reference for mobile factory simulation.

With all this data, the simulation will execute step by step several calculations, as represented in the figure 6. The first step consists in calculating some of the mobile factory costs. Then, we also calculate the quantities and some of the raw materials buying costs. After this, comes the transport phase (comparison of the different costs depending on the type of transport carried out, storage, etc…). Later, we appraise the raw materials supply. This step permits the organization in time of the raw materials supply, according to the duration of the production cycles. And finally, the last stepis about the calculation of the plant assembly and dismantling phases, which will be done to define how much time should be expected, if qualified staff would be requiered and how much, and the cost of this project.

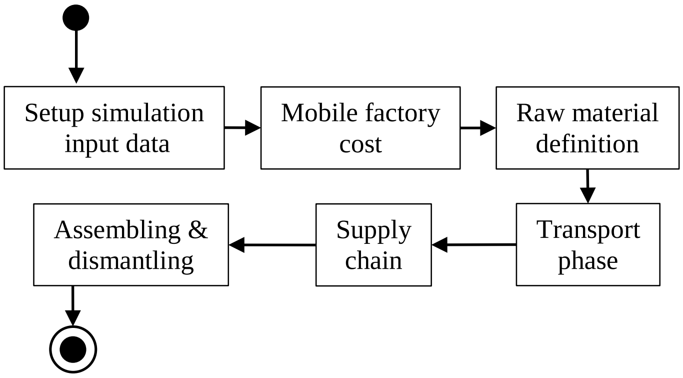

Figure 6. Mobile factory simulation scenario

In output of this simulation we will have data about solar field fabrication cost (in €/m²), information about the factory production rate, solar field installation time, the operators number required to install the plant, and the operators number required to operate the plant.

From an HLA point of view, this simulation and its associated database will be in a federate. The only communication will be for storing data input and output with the configurator.

10. ASSEMBLING SIMULATION WITH HLA

As we said earlier, each of those simulations will be defined as a federate in HLA architecture (Figure 7). The configurator will also be a federate that contains data necessary for all simulations. It will manage simulation scenario (order of execution), manage permission access to databases for each federate, and verify data values.

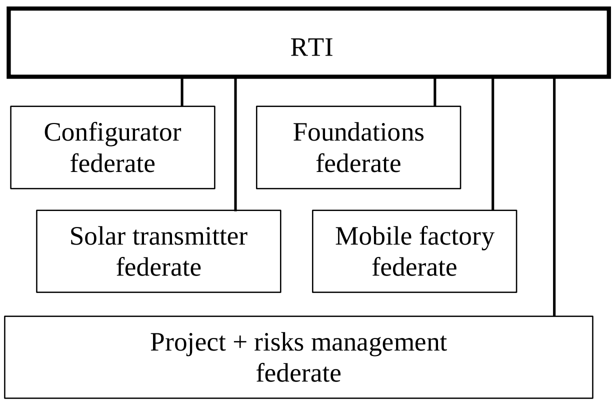

Figure 7. Global simulation as HLA view

Another federate which will be added to the federation is the project management simulation integrating risks (Rodney, 2014). Its function has the role of generating random events in the system to simulate risks, unexpected events, and defaults. It will act in the database system to generate understaffed situation, supply chain breaks, and so on. But it will also act in every simulation for generating problems, downturns, etc.

This federate has a special role because it will not only interact with the configurator, but also with all federates present in the federation.

In HLA standard, to guarantee a great communication, each federate must declare a Federation Object Model (FOM) that will describe the data exchange. In our case, each simulation previously described must declare in a file (the FOM) data that they must exchange with any other federate. For instance, the solar transmitter federate will communicate with the configurator federate, and the Project + risks management federate. So, its FOM will describe, data stored in input and output of the configurator as Object Class, and it will also describe as Interaction Class the communication with the risk generator. According to the configurator description, it will manage federate execution order, and will be described as an Interaction Class.

11. CONCLUSION

The industrial problem has been previously tackled domain by domain independently. Here the state of the art revealed that distributed simulation and the HLA standard can give an interesting answer to couple these heterogeneous works. One of the advantages of using this technology is the concept of interoperability and adaptability. The federation has permitted to define a set of entities representing different functions of the enterprise. Each function is (or must be) autonomous and have its own architecture. This allows the possibilities simply adding or removing federates to the simulation federation in order to revoke (or invoke) a functionality of the global simulation. The future step will consist in defining, according to HLA Federation Development Process (FEDEP), the behavior of each federate to preserve a global synchronization of information exchange over time.

REFERENCES

Benama, Y. (2014). Supporting make or buy decision for reconfigurable manufacturing system, in multi-site context. Ajaccio, France, 150–158.

El Amine, M. (2016). Integration of concept maturity in decision-making for engineering design: An application to a solar collector development. Springer-Verlag, London, pp 235-250.

Falcone, A., Garro, A., Anagnostou, A., Chaudhry, N. R., Salah, O.-A., & Taylor, S. J. . (2015). Easing the Development of HLA Federates The HLA Development Kit and its exploitation in the SEE Project. 2015 IEEE/ACM 19th International Symposium on Distributed Simulation and Real Time Applications (DS-RT), Chengdu, China.

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

Möller, B. (2012). The HLA Tutorial. Pitch Technologies, Sweden. Retrieved from http://www.pitchtechnologies.com

Piegay, N., & Breysse, D. (2015). Multi-Objective Optimization and Decision Aid for Spread footing design in Uncertain environment. Geotechnical Safety and Risk 5, Rotterdam, the Netherlands, pp 419-424.

Rodney, E. (2014). Integrating Risks in Project Management. 16th International Dependency and Structure Modelling, Paris, pp 419-424.

Yourdon, E., & DeMarco, T. (1989). Mordern Structured Analysis. Englewood Cliffs, NJ, Yourdon Press.

Zacharewicz, G., Chen, D., & Vallespir, B. (2009). Short-Lived Ontology Approach for Agent/HLA Federated Entreprise Interoperability. International Conference on Interoperability for Enterprise Software and Applications, Beijing, China.

Zacharewicz, G., Labarthe, O., Chen, D., & Vallespir, B. (2009). MULTI AGENT HLA ENTERPRISE INTEROPERABILITY (SHORT-LIVED ONTOLOGY BASED). The International Workshop on Modelling & Applied Simulation, Puerto de La Cruz, Spain, pp 187-196.

AUTHORS BIOGRAPHY

Simon GORECKI: Ph.D Student at IMS lab (Integration from Material to Systems laboratory), University of Bordeaux. Domain research is about simulating process with distributed simulations and HLA (High Level Architecture).

Gregory ZACHAREWICZ: Associate professor HDR at IMS lab, University of Bordeaux. Domain research is about M&S and enterprise engineering.

Nicolas PERRY: Full Professor at ParisTech ENSAM of Bordeaux. Domain research is about system engineering, product process integration and green manufacturing.