The use of the IEEE HLA standard to tackle interoperability issues between heterogeneous components

Ahmad Almaksour, Hadi Gerges, Simon Gorecki, Grégory Zacharewicz, Jalal Possik

To cite this version:

Ahmad Almaksour, Hadi Gerges, Simon Gorecki, Grégory Zacharewicz, Jalal Possik. The use of the IEEE HLA standard to tackle interoperability issues between heterogeneous components. DS-RT 2022 - IEEE/ACM 26th International Symposium on Distributed Simulation and Real Time Applications, Sep 2022, Alès, France. pp.175-178, 10.1109/DS-RT55542.2022.9932042 . hal-03857880

HAL Id: hal-03857880
https://hal.mines-ales.fr/hal-03857880

Submitted on 17 Nov 2022

The use of the IEEE HLA standard to tackle interoperability issues between heterogeneous components

Ahmad Almaksour[a], Hadi Gerges[a], Simon Gorecki[b], Gregory Zacharewicz[c], Jalal Possik[a, *]
[a]*FGES, Université Catholique de Lille, F-59000 Lille, France*
[b]*Univ. Bordeaux, CNRS, IMS, UMR 5218, Talence, France*
[c]*Laboratory for the Science of Risks, IMT Mines Ales, Ales, France*
[*] *Author to whom correspondence should be addressed.*
{ahmad.almosaalmaksour, hadi.gerges@lacatholille.fr}@lacatholille.fr, simon.gorecki@u-bordeaux.fr, gregory.zacharewicz@mines-ales.fr,
jalal.possik@univ-catholille.fr

***Abstract*—Classical simulation methods become not flexible and performant enough in complex models, necessitating the use of a distributed simulation technique to split the load and heterogeneity into separate sub-components and manage the simulation time between them. In this type of simulation, interoperability and reusability issues arise and should be addressed. The IEEE High-Level Architecture (HLA) standard for distributed simulation emphasizes federates interoperability and reusability, as well as time management and advanced data distribution techniques. This paper presents the methodologies and techniques used to develop the HLA federates, as part of the Simulation Exploration Experience (SEE) project, to virtually recreate a mission on the moon. This project is organized by the National Aeronautics and Space Administration (NASA) and the Simulation Interoperability Standards Organization (SISO). For each SEE component, an HLA interface was developed to make it compliant with other SEE federates and reusable during the simulation run. Based on HLA mechanisms, heterogeneous components with an HLA interface were able to interexchange objects/attributes and interactions/parameters.**

***Keywords*—Modeling and simulation, HLA, distributed simulation**

I. INTRODUCTION

Modeling and simulation (M&S) are rapidly evolving approaches that aids in the virtual evaluation of real-world systems. Using computer-based M&S to simulate various systems and operational situations is more practical, and less time/cost-consuming than executing them in real-world settings. M&S is widely used to identify weaknesses and issues of real-world systems, as well as to analyze the repercussions of any changes made in settings. As a technology that supports learning and training abilities, as well as decision support and system analysis and evaluation, the M&S area is gaining traction.

Nevertheless, in complex models, resource-intensive processing is required, and while scaling up servers and supercomputers can in part solve the challenge of handling complicated computations and massive volumes of data, all performances are still bound by server cores and memory. As a result of the growth of more advanced models, distributed technologies are regarded as one of the finest options, aiming to partition the simulation into sub-components and run them on linked workstations while taking time management and synchronization into account. Interoperability concerns develop at several levels in distributed simulation (DS). First, at the operating system level, where each component can run on a different operating system. At the application level, components can be written in several programming languages. Then, at the middleware level, where several communication protocols can be used. Finally, at the data level, with heterogeneities between variables. In addition, time management is required when many components run concurrently. When several components represent a sub-simulation of the entire system, they must be time-synchronized for the DS system to work properly.

In this paper, we use the High-Level Architecture (HLA) standard for distributed simulation created by the Department of Defense (DoD) in the United States, later adopted by IEEE as an international standard for DS. As part of the SEE project organized by the National Aeronautics and Space Administration (NASA) and the Simulation Interoperability Standards Organization (SISO) communities, this project aims at virtually reproducing a mission on the Moon using HLA standard and simulators to interact with NASA federates to provide a spacecraft mission in lunar site [3]. This paper provides a detailed explanation of the methods used to create HLA interactive federates using Java as a main programming language. The rest of this article is structured as follows. A summary of recent research and relevant prior material are provided in Section 2. The methodologies and distributed simulation architecture are discussed in Section 3. The results and analysis are discussed in Section 4 while a summary and conclusion are provided in Section 5.

II. RELATED WORK

Complex industrial processes necessitate methods and tools for interconnecting a set of heterogeneous components running on many platforms and operating systems [2], [9], [12]. Manufacturing modeling and simulation systems are made up of lots of processes, a set of exchanged data, the interconnection of services, and supply chain collaboration that can face interoperability barriers. According to IDEAS consortium in [8], interoperability is considered if there is a need of interactions to involve several entities and extends beyond the bounds of any particular system. As a result, achieving interoperability involves merging systems and removing incompatibilities. Our purpose is to address interoperability issues by identifying barriers (incompatibilities) that prohibit interoperability from occurring during a system's dynamic execution. As a result, modeling languages are required to capture all the system's components. Yet, to date, M&S for resource collaboration in complex systems still face interoperability barriers. It is stated in the proposed Model Driven Service Engineering Architecture (MDSEA) [7], that the different categories of resources should be identified at the design phase to enable better interoperability.

Then, at the time of execution, data interchange synchronization is required. Interesting contributions to enterprise modeling, interoperability, and distributed-simulation standards are detailed in the literature. Distributed simulation (DS) is a growing research area. One of its advantages is the distribution of tasks among many computers, which boosts program productivity. Another advantage of distributed simulation is the possibility to combine the strengths of many systems into a potent system while addressing interoperability concerns [4]. Interoperability is the ability of two or more entities to interact and collaborate despite differences in implementation language, data structure and format, execution environment, or model abstraction [5]. The synchronization of time is another well-known issue in distributed simulation. It is crucial to ensure that simulations deliver data at the appropriate times, in the correct orders, and without violating any causal constraints [6].

Falcone et al. developed a unified approach to combining the Functional Mockup Interface (FMI) and the IEEE 1516—High Level Architecture standard in [13]. To combine the standards, the authors offer two techniques: The first method involves constructing a hybrid federate using adapters. A Functional Mock-up Unit (FMU) is included in the federate, along with an adapter that manages all interactions between HLA-RTI and the FMU component. The second solution uses a mediator layer to coordinate the behavior of the entire system, which consists of a series of FMUs, an HLA federate that can use FMU to simulate a specific component, and a mediator layer.

Taylor et al. [14] proposed adopting distributed-simulation standards to improve a commercial simulation tool (commercial-off-the-shelf simulation packages (CSPs)). The authors demonstrate how a network of HLA federates allows existing business process models (modeled in BPMN) to communicate with each other.

In [2], [15], authors propose to use a Business Process Modeling and Notation modeling and simulation tool to orchestrate and solve interoperability issues in a complex system composed of two different distributed simulation standards (HLA and FMI).

In line with the literature, we can observe that most of the interoperability challenges across complex heterogeneous systems are solved using a distributed simulation standard such as the IEEE HLA standard. We will discuss these research topics in more detail in the section that follows.

III. MATERIAL AND METHODS

As part of this project, we have developed a Robot Technician (RT), a rover tasked with looking after other robots on the virtual moon. In order to provide other robots, the assistance and support they require, it is connected to them via the HLA interface. RT receives faults or warnings from linked robots, analyzes them, and then responds in accordance with scheduling and priority algorithms. For example, when two faults are received simultaneously from two different robots, RT determines the urgency and danger of both mistakes while taking into consideration the distance to reach these robots.

The tools used in this project are Pitch Run Time Infrastructure (pRTI) software for HLA development, Eclipse IDE for Java programming, and Unity, DON, and Blender for 3D modeling and development.

The 3D model of the RT was designed using Blender (Fig. 1). This model has its own objects/attributes and interactions/parameters to publish/subscribe to the HLA RTI in order to move on the lunar base and send/receive the appropriate data during the simulation run. Those settings are specified and defined in the next paragraphs.

Fig. 1. The 3D model of the RT was designed using Blender

HLA enables the integration of various simulation components i.e., “federates” that are connected via the Run Time Infrastructure (RTI), a service bus, which provides communication by data exchange between federates (Fig. 2.). The global simulation that brings together all federates by the RTI to work together is a “federation”. The federation uses a Federation Object Model (FOM), which is an XML file that describes objects/attributes and interactions/parameters to be exchanged between federates (Fig. 2). A federation must first be created for federates to join before synchronization and data exchange are possible. Through HLA's publish/subscribe and time management mechanisms, each federate must first join the federation to establish connections between them.

The SpaceFOM used in this project was developed by SISO in 2015 to provide a practical method for system integration and facilitate collaboration between space system simulations [16].

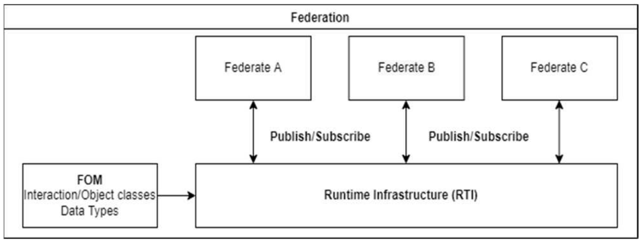
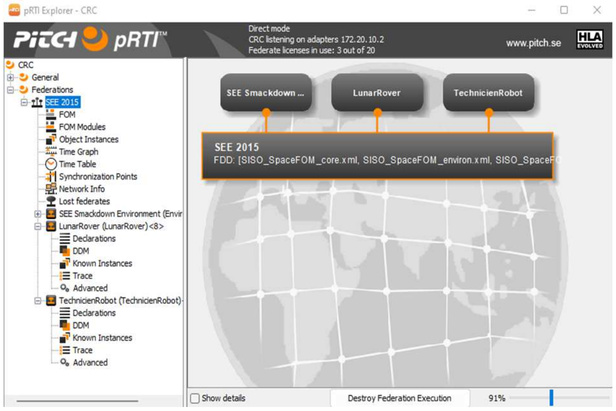


Fig. 1. HLA architecture

Among the various capabilities developed in the RT federate are the ability to recharge the batteries of other robots and change their tires. An object class and an interaction class has been added to the FOM file to define the information that should be sent/received to/from the RTI.

A master platform has been developed to build the federation. The starter kit framework (SKF) and pitch RTI (pRTI) java libraries have been used fo the HLA federates and establish the connection to the RTI central unit (Fig. 3).



Direct mode
CRC listening on adapters 172.20.10.2
Federate licenses in use: 3 out of 20

SEE 2015
FDD: [SISO_SpaceFOM_core.xml, SISO_SpaceFOM_environ.xml, SISO_SpaceF

Fig. 3. HLA Federation using pitch prti

An object class has been developed that represents our federate (Technician Robot), and we created the Federation Object Model (FOM), which contains the description of the model. This object has (1) *'name'*, (2) *'parent_name'*, (3) *'position'*, and (4) *'available'* as attributes.

(1) The *'name'* attribute refers to the name of the object.
(2) *'parent_name'* defines the reference frame. In other words, it specifies where the object is located.
(3) *'position'* represents the position of the object with three coordinates x, y, and z.
(4) *'available'* represents the availability of the object.

In order to exchange necessary information, the object contains several methods such as getAvailability() and getPosition(). Before joining the federation, a federate should be connected to the RTI using the method *connectToRTI(settings_designator*). Some parameters have to be filled out to connect to the RTI, such as the host address and port number. Following this, we use the method *joinFederationExecution()* to join the space master federation (Fig. 4). In addition, the object should be subscribed to the Federation using the method *subscribeSubject('object class')*. To run the experiment, another federate 'Lunar Rover' has been developed to interact with the RT object federate.

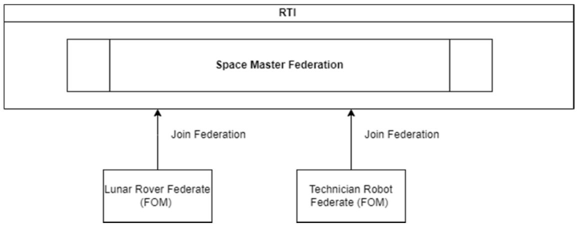


Fig. 2. Lunar Rover and Technician Robot federates joining the space master federation

A *'communication'* interaction class has been developed, which has a string-typed attribute, getter and setter methods, used to interact with other federates and declared in the FOM. The RT object updates parameter values for the interaction class using the method *updateInteraction("interaction object class")* and publishes the interaction class parameters using the SKF library's *publishInteraction("interaction class")* function. For instance, if the RT receives a request for assistance from another federate, the RT publishes a message *"I am on the way"* or *"the repair is complete"*. It is important to remember that the robots must subscribe using the method *subscribeInteraction("interaction class")* in order to obtain the interaction class parameters. Robots will not be able to get the message otherwise. Following subscription, the federates get all data transmitted by the "communication" interaction class.

Another interaction class *"HelpRequest"* has been created, this class sends a help request to the RT federate. RT receives this information and identifies the location of the robot that is asking for help or repair. An interaction *'Needs'* has also been developed to define the needs of the object federates. For example, the Lunar Rover may publish the type of damage, as an interaction, to the RTI so that the RT federate, which is subscribed to this interaction, receives this information and acts accordingly. Fig. 5 provides a quick overview of how the federates collaborate through the RTI and the federation to which they belong.

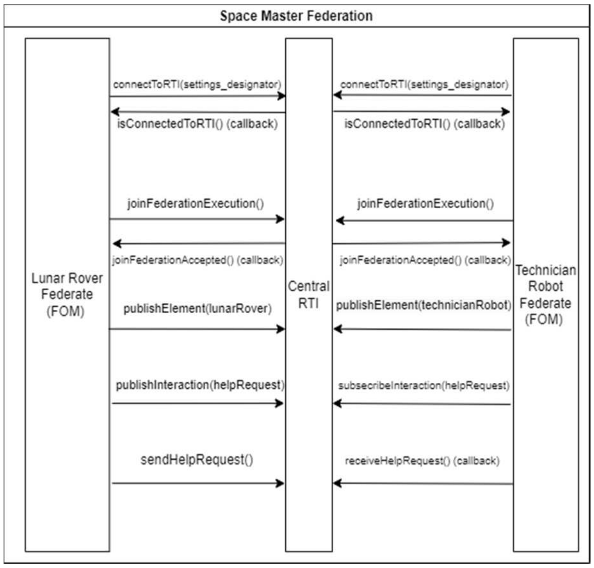


Fig. 3. HLA environment and methods used for communication

Blender is used to alter the predesigned 3D model, and the output file has then been uploaded to the DON directory so that the model may be seen in a 3D form. The main goal is to see how the RT 3D model interacts with any other model (federate) having an HLA interface that is linked to the same federation.

IV. RESULTS AND ANALYSIS

As a result, the contact between the RT and the Lunar Rover components was established by resolving interoperability issues at application, data, middleware, and OS levels using HLA. As well, RT and Lunar Rover federates smoothly operates within the space master federation created by the NASA team. In the SEE scenario, the Lunar Rover may ask the RT for assistance. In response, the RT approaches the Lunar Rover and acts accordingly. The arrival time to the Lunar Rover varies depending on where each object is located on the moon's surface. When there are several broken robots, RT responds based on priority. (e.g., Recharging a battery versus Fixing an inflated tire). The distance, processing time, and level of urgency are the factors that determine the priority ranking. The interest for the students is to learn how to adjust the parameters to be able to correct the errors as quickly as possible. Fig. 6 describes an example of the communication between RT and the Lunar Rover.

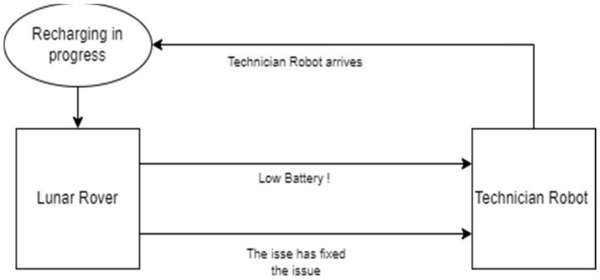


Fig. 4. Example of Communication Between the Federates

V. CONCLUSION AND FUTURE WORK

This paper is written as feedback from the SEE project organized by NASA and the SISO communities. The main objective of this project is to get students to face the challenges of interoperability between heterogeneous components using HLA as an international standard for distributed simulation. An explanation of the methods used to create HLA interactive federates is presented in this paper. We intend to draw more students into the world of distributed simulation as part of future development. In particular, work with the HLA time management mechanism that is not yet fully implemented. We believe that working in an international team to discuss the HLA mechanism and how to control the simulation's progression over time while receiving an advance grant from the RTI is a worthwhile experience for students.

ACKNOWLEDGMENT

We thank the Faculty of Management, Economics and Sciences (FGES) at Lille Catholic University for the financial support of our research.

REFERENCES

[1] J. Possik, A. D'Ambrogio, G. Zacharewicz, A. Amrani, and B. Vallespir, "A bpmn/hla-based methodology for collaborative distributed des," in *2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, Capri, Italy, 2019, pp. 118–123. doi: 10.1109/WETICE.2019.00033.

[2] S. Gorecki, J. Possik, G. Zacharewicz, Y. Ducq, and N. Perry, "Business Models for Distributed-Simulation Orchestration and Risk Management," *Information*, vol. 12, no. 2, p. 71, 2021.

[3] P. R. Elfrey, G. Zacharewicz, and M. Ni, "SMACKDOWN: adventures in simulation standards and interoperability," in *Proceedings of the 2011 winter simulation conference (wsc)*, 2011, pp. 3958–3962.

[4] J. Possik *et al.*, "A Distributed Simulation Approach to Integrate AnyLogic and Unity for Virtual Reality Applications: Case of COVID-19 Modelling and Training in a Dialysis Unit," in *2021 IEEE/ACM 25th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, Valencia, Spain, 2021, pp. 1–7. doi: 10.1109/DS-RT52167.2021.9576149.

[5] A. D'Ambrogio and D. Gianni, "Using CORBA to enhance HLA interoperability in distributed and web-based simulation," in *International Symposium on Computer and Information Sciences*, 2004, pp. 696–705.

[6] G. Zacharewicz, N. Giambiasi, and C. Frydman, "Gdevs/hla environment: A time management improvement," in *the 17th IMACS World Congress on Scientific Computation, Applied Mathematics and Simulation*, 2005, pp. T4-I.

[7] H. Bazoun, G. Zacharewicz, Y. Ducq, and H. Boyé, "SLMToolBox: an implementation of MDSEA for servitisation and enterprise interoperability," in *Enterprise Interoperability VI*, Springer, 2014, pp. 101–111.

[8] C. IDEAS, "Thematic Network, IDEAS Interoperability Development for Enterprise Application and Software Roadmaps," IDEAS Tech Report Annex, 2002.

[9] J. Possik, "Contribution to a methodology and a co-simulation framework assessing the impact of lean on manufacturing performance," PhD Thesis, Bordeaux, 2019.

[10] J. Possik, C. Yaacoub, S. Gorecki, G. Zacharewicz, and A. D'Ambrogio, "A Model Based Systems Engineering Approach to Automated Music Arrangement," in *2021 Annual Modeling and Simulation Conference (ANNSIM)*, Jul. 2021, pp. 1–12. doi: 10.23919/ANNSIM52504.2021.9552105.

[11] W. Guédria, Y. Naudet, and D. Chen, "Maturity model for enterprise interoperability," *Enterp. Inf. Syst.*, vol. 9, no. 1, pp. 1–28, 2015.

[12] D. Lee, "Development of Mediator-Based Direct Wokrflow Simulation System and HLA/RTI-Based Collaborative BPMS Middleware," PhD Thesis, Korea Advanced Institute of Science and Technology Daejeon, Korea, 2010.

[13] A. Falcone and A. Garro, "Distributed co-simulation of complex engineered systems by combining the high level architecture and functional mock-up interface," *Simul. Model. Pract. Theory*, vol. 97, p. 101967, 2019.

[14] S. J. Taylor, N. Mustafee, S. J. Turner, K. Pan, and S. Strassburger, "Commercial-off-the-shelf simulation package interoperability: Issues and futures," in *Proceedings of the 2009 Winter Simulation Conference (WSC)*, 2009, pp. 203–215.

[15] S. Gorecki, J. Possik, G. Zacharewicz, Y. Ducq, and N. Perry, "A multicomponent distributed framework for smart production system modeling and simulation," *Sustainability*, vol. 12, no. 17, 2020, doi: 10.3390/su12176969.

[16] E. Z. Crues, D. Dexter, A. Falcone, A. Garro, and B. Möller, "SpaceFOM - A Robust Standard for Enabling A-Priori Interoperability of HLA-Based Space Systems Simulations," *Journal of Simulation*, pp. 1–21, 2021.