USING THE HLA STANDARD IN THE CONTEXT OF AN INTERNATIONAL SIMULATION PROJECT: THE EXPERIENCE OF THE "SMASHTEAM"

[(a)]Alberto Falcone, [(b)]Alfredo Garro

[(a),(b)]Department of Informatics, Modeling, Electronics, and Systems Engineering (DIMES), University of Calabria, via P. Bucci 41C, 87036, Rende (CS), Italy

[(a)]alberto.falcone@dimes.unical.it, [(b)]alfredo.garro@dimes.unical.it

ABSTRACT
To understand, predict and optimize the behavior of systems, several M&S techniques are emerging. A well-known and mature standard is the IEEE 1516-2010 - High Level Architecture (HLA) that is a general-purpose architecture to support the reuse and interoperability of distributed simulation components. In this context, the paper presents a concrete experience in using the HLA standard in the context of an international project, the Simulation Exploration Experience (SEE), organized by the Simulation Interoperability Standards Organization (SISO), in collaboration with NASA and other industrial partners, and involving several university teams in the distributed simulation of a Moon base.

Keywords: Modeling and Simulation, High Level Architecture, Distributed Simulation, Model-based Design; 3D visualization

1. INTRODUCTION
Modeling and Simulation (M&S) represents a pillar supporting the acquisition of knowledge so as to understand, predict and optimize the behavior of systems on which to perform experiments and theoretical analyses (Banks 1998; Falcone, Garro, Tundis, 2014; Fortino, Garro, Mascillaro, Russo 2007; Fortino, Garro, Russo 2004; Fujimoto 2000; Kuhl, Weatherly and Dahmann 1999).
The increasing complexity of modern systems makes their design, development and operation extremely challenging and therefore new M&S methods, techniques and tools are emerging, also to benefit from distributed simulation environments. In this context, one of the most well-known and mature standard is the IEEE 1516-2010 - High Level Architecture (HLA) (IEEE Standard for Modeling and Simulation - High Level Architecture) that attempts to handle this complexity by providing a specification of a distributed infrastructure in which simulation units (called *Federates*) can run on standalone computers and communicate with one another in a common simulation scenario (called a *Federation*) through a Run Time Infrastructure (RTI) that is a sort of service bus that provides information, synchronization, and coordination services to the connected *Federates* (IEEE Standard for Modeling and Simulation - High Level Architecture).
To promote the adoption of distributed simulation practices, standards, and technologies, since 2011, the Simulation Interoperability Standards Organization (SISO), in collaboration with NASA and other research and industrial partners, has been organizing an annual event, named Simulation Exploration Experience (SEE) (The Simulation Exploration Experience project). The main objective of SEE is to provide undergraduate and postgraduate students with practical experience of participation in international projects related to M&S and, especially, to gain expertise in developing distributed simulations based on the HLA standard and compliant tools, by involving them in the simulation of a space exploration scenario.
This paper presents the experience of the "SMASHTeam" of the University of Calabria in the 2016 edition of the SEE project with the aim to highlight the exploited software engineering solutions as well as lessons learned and future research directions. In particular, three HLA Federates developed by the "SMASHTeam" through the use of the *SEE HLA Starter Kit* software framework (The SEE HLA Starter Kit project) will be presented: a *"Command and Control center"*, and two astronauts, one operating on the Moon surface and the other engaged in a EVA (Extra Vehicular Activity), *ASTROUNI*, and *OrbitASTROUNI* respectively.
The rest of the paper is organized as follows. An overview of the IEEE 1516-2010 - High Level Architecture (HLA) standard and a brief description of the SEE HLA Starter Kit software framework are presented in Section 2. In Section 3, the SEE project is described with particular focus on its mission and main objectives. Sections 4, 5 and 6 describe, with emphasis on their objectives, behaviors and 3D models, the *Command and Control center (C2C), ASTROUNI* and *OrbitASTROUNI* Federates, respectively. Finally, conclusions are drawn and future research directions are presented.

2. THE IEEE HIGH LEVEL ARCHITECTURE AND THE SEE HLA DEVELOPMENT KIT

HLA is an IEEE standard (IEEE 1516) for distributed computer simulation systems (IEEE Standard for Modeling and Simulation - High Level Architecture). It was developed by the U.S. Modeling and Simulation Coordination Office (M&S CO) (Modeling and simulation coordination office) in 1995, as a general-purpose simulation systems architecture so as to facilitate the reuse, interoperability and integration of distributed simulation components within a common simulation environment. Although it was initially developed for purely military applications, it has been widely used in non-military industries for its many advantages related to the interoperability and reusability of distributed and heterogeneous distributed simulation modules.

In the HLA standard a distributed simulation is called *Federation*. A *Federation* is composed of several HLA simulation entities, each called a *Federate,* that share a common specification of data communication which is defined in Federation Object Model (FOM) (IEEE Standard for Modeling and Simulation - High Level Architecture). A *Federate* is defined as an application that is conforms to the HLA standard. *Federates* consume and support the interfaces that are specified in the HLA Federate Interface Specification, thus can participate in a distributed simulation execution interacting one another by using the Run-Time Infrastructure (RTI).

Figure 1 shows a generic HLA Federation with the services provide by the RTI.

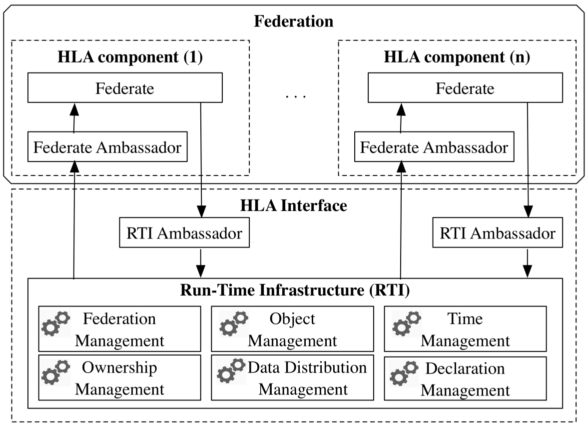

Figure 1: A generic HLA Federation.

The HLA standards defines a set of specifications that can be organized in three major core components:

- *Interface specification*, which defines an interface specification between the Federate and the RTI, which provides communication and coordination services to the Federates;
- *Object model template (OMT)*, which specifies the format of simulation data in terms of a hierarchy of object classes, attributes, and interactions among Federates running in a Federation execution;
- *HLA Framework and Rules*, which outlines the rules that a Federate must follow in order to be compliant to the standard (IEEE Standard for Modeling and Simulation - High Level Architecture).

The RTI represents a backbone of a Federation execution that provides a set of services to manage the inter-Federates communication and data exchange. They interact with RTI through the standard services and interfaces to participate in the distributed simulation execution. It supports the HLA rules and provides a set of services over the interfaces specified in the *Interface Specification*. More in details, the *Federation Management* keeps track of the execution state of both Federation and the joined Federates that are running on an RTI; the *Object Management* defines and handles how the running Federates can register, modify and delete object instances and to send and receive interactions once they have ownership of them; the *Time Management* service manages the logical time of a Federation including delivery of time stamped data and advancing of the Federates time; the *Ownership Management* regulates acquiring and divesting ownership of registered objects; the *Data Distribution Management* manages the distribution of state updates and interaction information in a simulation executions. It limits and controls the volume of data exchanged during the simulation among Federates so as to relay data only to those requiring them; finally, the *Declaration Management* manages data exchange among Federates according to a Federation Object Model (FOM) and by using a publish/subscribe scheme. For each Federate, this service keeps track of types of objects, attributes, or interactions that it has published/subscribed. In this way, the RTI is able to deliver data from publishers to subscribers.

2.1. The SEE HLA Starter Kit

Building complex and large distributed simulations systems, based on the IEEE 1516-2010 standard, is generally a challenging task and requires a considerable period of training to ensure a better understanding of its basic principles and function; as a consequence, the development of HLA *Federates* is complex, error-prone and time consuming because developers cannot focus on the specific aspects of their Federates but have to deal with the underlying HLA functionalities, such as the management of the simulation time, the connection on the RTI, and the management of common RTI exceptions. As a result, they cannot fully focus on the specific aspects of their own simulations.

In order to overcome the above mentioned issues, and helps teams in developing their HLA simulation components in the context of the SEE project, the the SMASH-Lab (System Modeling And Simulation Hub - Laboratory) of the University of Calabria (Italy) working in cooperation with the NASA JSC (Johnson Space Center), Houston (TX, USA), developed a software framework called *SEE HLA Starter Kit*.

The *SEE HLA Starter Kit* aims at easing the development of SEE Federates by providing: (i) a Java-based *software framework* (*SEE-SKF*) for the development in Java of SEE Federates; (ii) a *technical documentation* that describes the *SEE-SKF*; (iii) *user guide* to support developers in the use of the framework; and (iv) some *video-tutorials* that show how to create both the structure and the behavior of a SEE Federate (The SEE HLA Starter Kit project).

The *SEE HLA Starter Kit*, released under the LGPL licence, has been successfully experimented in the SEE project since the 2015 edition. In the 2016 edition, the Universities of Calabria (Italy), Bordeaux (France), Brunel (London, UK) and the Faculdade de Engenharia de Sorocaba, FACENS (Brazil) developed their SEE Federates by using the Kit (Anagnostou, Chaudhry, Falcone, Garro, Salah and Taylor. ACM/IEEE DS-RT 2015; Anagnostou, Chaudhry, Falcone, Garro, Salah and Taylor. ACM SIGSIM PADS 2015; Falcone, Garro 2016); the provided features improved the reliability of their SEE Federates and reduced the problems arising during the final integration and testing phases of the SEE.

3. THE SIMULATION EXPLORATION EXPERIENCE (SEE) PROJECT

To promote the adoption of distributed simulation practices, standards, and technologies, since 2011, the Simulation Interoperability Standards Organization (SISO), in collaboration with NASA and other research and industrial partners, has been organizing an annual event, named *Smackdown* and, now renamed *Simulation Exploration Experience* (SEE) (The Simulation Exploration Experience project).

The main objective of the SEE project is to offer a practical experience of participation in international projects related to M&S and, especially, to the HLA standard and compliant tools to undergraduate and postgraduate students so as to increase their skills in M&S techniques.

The 2016 edition took place during the 2016 Spring Simulation Multi-Conference (*SpringSim'16*) in Pasadena (CA, USA) from 3rd to 6th April 2016 where nine universities took part: University of Liverpool, University of Nebraska, University of Bordeaux, University of Munich, University of Brunel, University of Genoa, University of Alberta, University of Calabria and the Faculdade de Engenharia de Sorocaba FACENS (Brazil).

The team of the university of Liverpool contributed with three HLA Federates: (i) an area of the moon that is designated as a surface mining zone (*Regolith Mining Area*); (ii) a flight vehicle with six degrees of freedom (*Lunar Buggy*) that operates above the moon base; and (iii) a surface explorer that is able to inspect the lunar surface searching for caves and deep craters (*Sub-surface Feature Explorer*).

The University of Nebraska team created a vehicle that can transfer both cargo between facilities and fuel to other vehicles on the lunar surface (*Rover*).

The team of the Bordeaux University created a facility to store the supplies (*Supply Depot*).

The team of the Munich University developed an outpost located at Libration Point Two (*L2-Outpost*).

The University of Brunel team developed an agent based simulation of lunar regolith mining based on REPAST (*Excavator*) (Taylor, Revagar, Chambers, Yero, Anagnostou, Nouman and Chaudhry 2014).

The University of Genoa team produced: (i) an asteroid defense system (*IPHITOS*), which include an asteroid detection system and a missile base, and (ii) a medical facility (*SISMA*) that constantly checks the health status of astronauts.

The team of the university of Alberta developed a construction site on the lunar surface (*Modular Construction*) where some excavators are making a facility (*Modules 3D Printing*).

The *University of Calabria*, contributed with two teams that produced four Federates.

The *"SMASHTeam"* developed: (i) the Command and Control Center (*C2C*) that offers a flexible communication services to the other entities involved in the simulation scenario; (ii) an astronaut, called *ASTROUNI*, that operates in the oxygen factory placed on the moon surface and is constantly monitored by the Astronaut Health Monitoring System; (iii) an astronaut, called *OrbitASTROUNI*, that operates in orbit mainly in proximity of a "Space Exploration Vehicle" and, as well as *ASTROUNI,* is constantly monitored by the Astronaut Health Monitoring System (Falcone, Garro, Longo, Spadafora 2014).

The *"University of Calabria – Team 1"*, developed (iv) a drone called *DREDIS*. It proposes an innovative solution based on using drones inside the lunar base for reconnaissance and exploration missions, thus considering drones as a lifesaving resource that increases safety for a traditionally hazardous situation.

The team of FACENS developed the Lunar habitat (*LuHA*) that provides to the astronauts housing, shelter, and a place to fulfill their basic needs, as well as communicate with the mission control center placed on earth.

All the teams developed for their Federates a 3D model so as to interact with the *Distributed Observer Network* (*DON*) environment (The Simulation Exploration Experience project). *DON* is a real-time 3D visualization environment, which is based on Unity3D, (3D Game Engine, Unity3D) developed by the NASA team that tracks all the SEE Federates and displays their updates on the 3D environment during the simulation execution through the *DON Visualization Tool* (*DON-VT*).

The following sections describe in detail, the three Federates developed by the *"SMASHTeam"* for the 2016 edition of the project. The Federates have been built in Java by using the capabilities provided by the *SEE HLA Starter Kit* (see Section 2), whereas their 3D models were designed through the use of Autodesk 3DS MAX (Autodesk 3DS MAX).

4. THE COMMAND AND CONTROL CENTER FEDERATE

The Command and Control Center (*C2C*) provides flexible communication functionalities to the other entities participating in the simulation scenario. It works as a mediator and manage all the communication among entities. To provide its services, the *C2C* is equipped with: (i) an *antenna* mounted on a tower for managing surface-to-surface communications among the entities populating the moon base; and (ii) a *parabolic antenna* for managing surface-orbit communications (see Figure 5).
The *C2C* Federate define a compact and effective way to interact from/to entities. In particular, a standard packet format has been defined as showed in Figure 2.

Source Federate	Destination Federate
Payload	

Figure 2: The Command and Control Center (C2C) packet format.

The *Source Federate* field contains the identification of the sender of the packet.
The *Destination Federate* field contains the identification of the receiver of the packet.
The *Payload* field represents the data portion of the packet.
The *C2C* Federate defines two interaction classes created by using the *@InteractionClass* class provided by the SKF to declare a HLA InteractionClass (IEEE Standard for Modeling and Simulation - High Level Architecture; Möller 2013). More in details: (i) the *C2CMessageTx* interaction class manages the communication from the C2C to another entity; and (ii) *C2CMessageRx* handles the communications coming from an entity (*Source Federate*) to the C2C.
Upon the receipt of the data packet, the C2C inspects and forward it by using a "*C2CMessageTx – data*" message to the destination entity (*Destination Federate*).
The behavior and 3D model of the C2C is described in the following subsections.

4.1. Behavior

The SKF provides, through the *SKFAbstractFederate* class, a life cycle to a SKF-Based Federate.
In this way, developers can focus on the specific *proactive* and *reactive* behavioral aspects of the Federate by defining the action to perform (Falcone, Garro 2016; Falcone, Garro 2015; The SEE HLA Starter Kit project).
The *proactive* behavior of the C2C federate is shown in Figure 4 through a UML Activity Diagram.

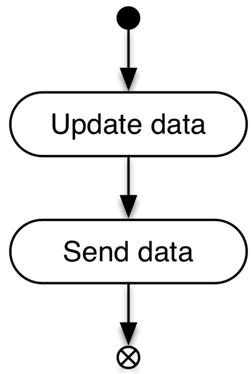

Figure 3: The proactive behavior of the C2C federate.

In the *Update data* activity, the attributes related to the C2C Federate such as position, attitude, height and power are updated whereas, in the *Send data* activity, they are published on the RTI.
The *reactive* behavior of the C2C federate is shown in Figure 4 through a UML Activity Diagram that presents all the activities performed.

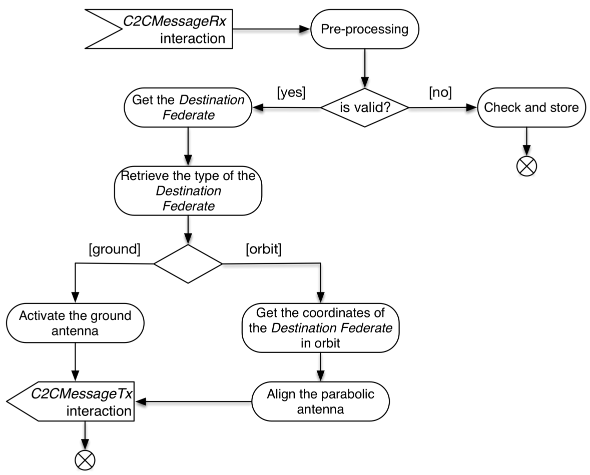

Figure 4: The reactive behavior of the C2C federate.

Upon the receipt of a *C2CMessageRx* interaction, the C2C federate pre-processes it so as to verify if the incoming interaction is valid or not (*Pre-processing*). If not, the C2C performs a check to track down both problems and errors occurred during the *Pre-processing* activity and then stores the results (*Check and store*). Otherwise, the information related to the destination entity is gathered (*Get the Destination Federate*) as well as its type that can be either *ground* or *orbit* (*Retrieve the type of the Destination Federate*). This information is used by the C2C Federate so as to use the right antenna for forwarding the data packet. If it is a *ground* entity, the antenna used for managing the surface-to-surface communications is activated (*Activate the ground antenna*), and then an "*C2CMessageTx – data*" interaction is sent. Otherwise, the C2C gets the coordinates of the destination entity (*Get the coordinates of the Destination Federate in orbit*), positions and orientates the *parabolic antenna* (*Align* *the parabolic antenna*), which is used for managing surface-orbit communications, and then sends an "*C2CMessageTx – data*" interaction.

4.2. 3D model

Figure 5 shows the 3D model of the C2C Federate created according to the specifications provided by the DON team (The Simulation Exploration Experience project) in order to allow the visualization of the model with the DON visualization environment. More in details:

- The size of the 3D model must be less than 20k polygons;
- The 3D model must be exported as a Wavefront (*.obj*) file with only one Material Template Library (*.mtl*) file for the textures and materials;
- A *.txt* configuration file containing some information related to the visualization of the model need to be created.

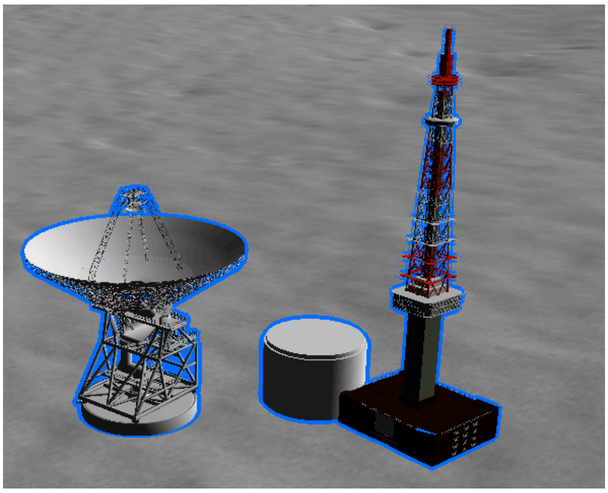

Figure 5: The 3D model of the C2C Federate.

The following code listing defines the information enclosed in the *.txt* configuration file:

```
id= "<entity_name>":
    filename= "<modelname.obj>":
        position: (0, 0, 0)
        orientation: (0, 0, 0, 1)
        scale: (1, 1, 1)
```

The *<entity_name>* tag contains the identification of the *ObjectClass* (IEEE Standard for Modeling and Simulation - High Level Architecture; Möller 2013), which is published by the Federate. The name of the 3D model file (*.obj*) is defined in the *<modelname.obj>* tag.

The *position*, *orientation* and *scale* define the default values for the 3D model. These attributes are self-explanatory, therefore they do not need insights.

5. THE ASTROUNI FEDERATE

ASTROUNI is an astronaut that operates in the oxygen factory placed on the Moon surface and is constantly monitored by the Astronaut Health Monitoring System (AHMS) (The Simulation Exploration Experience project).

The simulation scenario in which ASTRUNI is involved as well as its behavior and 3D model are described in the following subsections.

5.1. The "Suit-Alert" Scenario

The reference simulation scenario of the SEE Project (The Simulation Exploration Experience project) concerns a human settlement called Moon base composed of scientific equipment, storage buildings, rovers, and other elements to allow astronauts to live and work on the Moon.

Part of the SEE 2016 scenario is an emergency situation in which the AHMS detects a malfunction of the ASTROUNI's suit.

In particular, when the AHMS detects that malfunction, it sends (through the C2C) an interaction with payload "*suit-alert-ongoing*" to ASTROUNI that, to avoid health risks, walks toward and enters the oxygen factory room where it shelters awaiting for help.

When ASTROUNI is in the factory, it sends a *C2CMessageRx* interaction with payload "*request-room*" to the *LuHa* Federate (see Section 3) so as to receive help.

The *LuHa* Federate asks the *Rover* (see Section 3), through a *C2CMessageRx* interaction with payload "*astronaut-pick-up*", to pick up and take ASTROUNI to the medical center.

Once *LuHa* receives positive feedback from the *Rover*, it sends a *request-room-ok* message to ASTROUNI that exits the oxygen factory room and gets in the *Rover*.

Note that all the above introduced communications are mediated by the C2C Federate (see Section 4).

5.2. Behavior

The life cycle of the ASTROUNI Federate has been developed by using the functionalities provided by the *SEE HLA Starter Kit*, as well as the C2C Federate (see Subsection 4.1).

The *proactive* behavior of the ASTROUNI federate is shown in Figure 6 through a UML Activity Diagram.

In the *Get target's information* activity, the information related to the *target* component that needs to be repaired is retrieved. If the *target* is empty, which means that the repair activities have been completed, ASTROUNI reaches the Moon base (*Reach the moon base*) and its life cycle ends. Otherwise, the component is repaired by ASTROUNI (*Repair the component*) and the information related to the astronaut such as the position, attitude and health state are updated and published on the RTI (*Update data*).

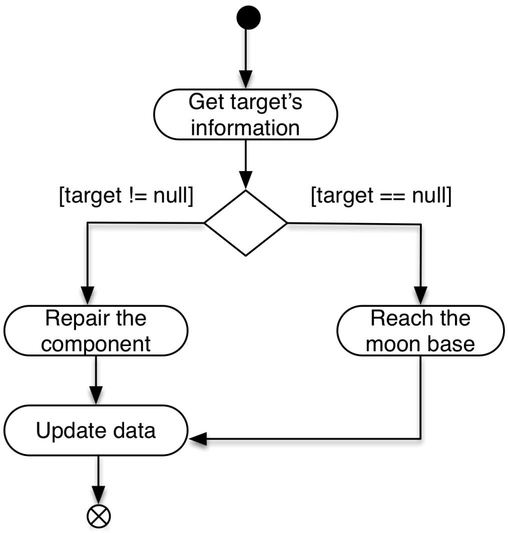

Figure 6: The proactive behavior of the ASTROUNI Federate.

The *reactive* behavior of the ASTROUNI Federate is shown in Figure 7 through a UML Activity Diagram that presents all the performed activities.
Upon the receipt of a *C2CMessageTx* interaction, the ASTROUNI Federate pre-processes it to get the payload of the message (*Pre-processing*). If it is *"suit-alert-ongoing"*, ASTROUNI goes to the oxygen factory where it waits for help. A request message is created (*Create a request message*) and sent to the *LuHa* Federate when the astronaut reaches the room.
Otherwise, if the payload is *"request-room-ok"*, ASTROUNI exits the oxygen factory room (*Leave the oxygen factory*) and gets in the *Rover* (*Reach the Rover*).

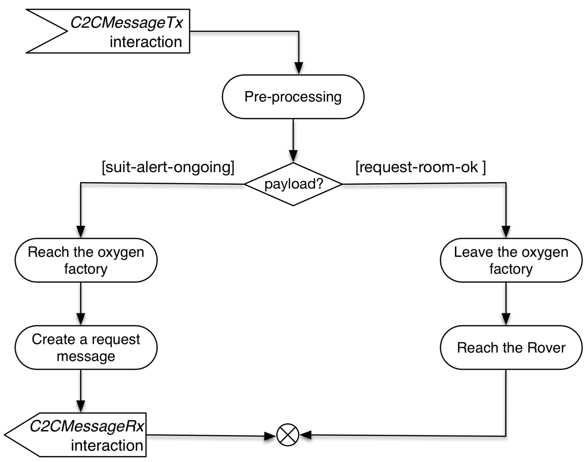

Figure 7: The reactive behavior of the ASTROUNI Federate.

5.3. 3D model

Figure 8 shows the 3D model of the ASTROUNI Federate.

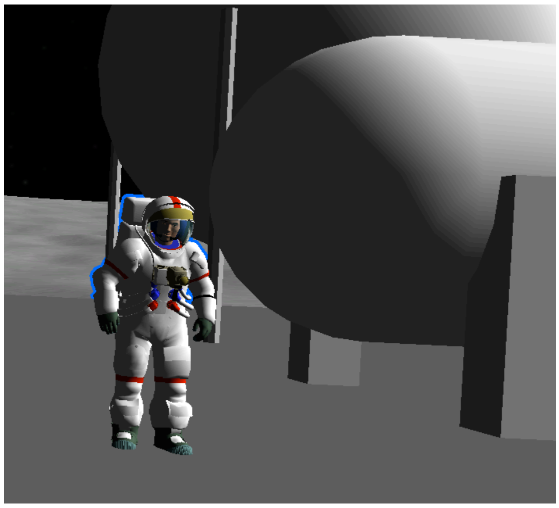

Figure 8: The 3D model of the ASTROUNI Federate.

As well as the 3D model of the C2C federate, it has been created according to the specifications provided by the DON team (see Section 4.2).

6. THE ORBITASTROUNI FEDERATE

OrbitASTROUNI is an astronaut that operates in orbit, mainly in proximity of a "Space Exploration Vehicle" engaged in a EVA, and, as well as *ASTROUNI,* is constantly monitored by the AHMS.
The behavior and 3D model are described in the following subsections.

6.1. Behavior

The *proactive* behavior of the OrbitASTROUNI Federate is defined by using the capabilities provided by the *SKFAbstractFederate* class of the SEE-SKF (Falcone, Garro 2016; Falcone, Garro 2015).
Figure 9 shows the *proactive* behavior through a UML Activity Diagram.

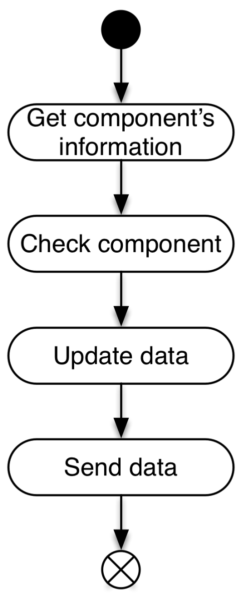

Figure 9: The proactive behavior of the OrbitASTROUNI Federate.

In the *Get components's information* activity, the information related to the *component* that needs to be checked is retrieved. After that, the component is checked by OrbitASTROUNI (*Check component*) and the information related to both the astronaut (e.g. position, attitude, health state, etc.) and status of the component are updated (*Update data*) and published on the RTI (*Send data*).
The *reactive* behavior of the ASTROUNI Federate is shown in Figure 10 through a UML Activity Diagram that presents the all activities performed.
Upon the receipt of a *C2CMessageTx* interaction, the OrbitASTROUNI Federate pre-processes it to get the information related to the component that needs to be checked (*Pre-processing*). In the *Add component* activity the information retrieved in the previous one is added in the check list of the astronaut.

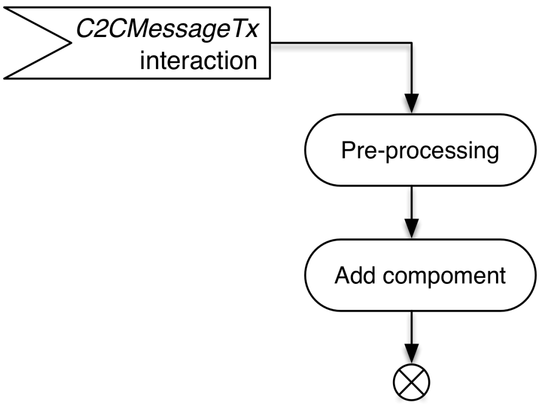

Figure 10: The reactive behavior of the OrbitASTROUNI Federate.

6.2. 3D model
Figure 11 shows the 3D model of the OrbitASTROUNI Federate.

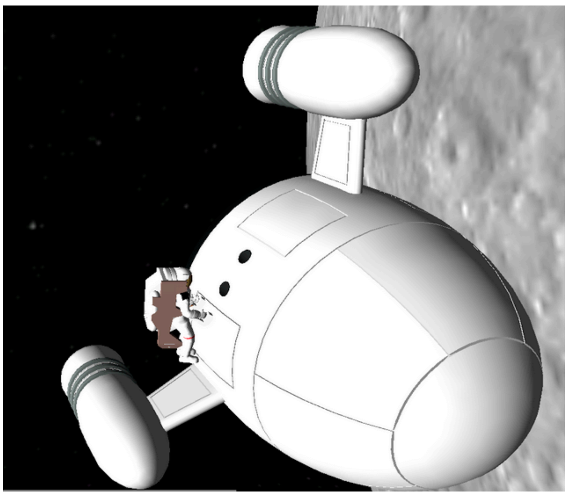

Figure 11: The 3D model of the OrbitASTROUNI Federate.

As well as the other federates, it has been created by using 3DS MAX following the advices provided by the DON team (see Section 4.2).

DISCUSSION AND CONCLUSIONS
The paper has presented the experience of the "SMASHTeam" of the University of Calabria in participating to the 2016 edition of the SEE (Simulation Exploration Experience) event (formerly named *"Smackdown"*). Specifically, the Team has been in charge of the development of the *Command and Control Center* (*C2C*) able to provide flexible communication services to the other entities populating a Moon base simulated scenario as well as two astronauts: *ASTROUNI*, which operates in the oxygen factory placed on the Moon surface and that is constantly monitored by the Astronaut Health Monitoring System (AHMS); and *OrbitASTROUNI*, which operates in orbit mainly in proximity of a "Space Exploration Vehicle" and, as well as *ASTROUNI,* is constantly monitored by the AHMS. Moreover, the team developed for each Federate a 3D model so as to interact with the *Distributed Observer Network* (*DON*) environment.
To achieve the results presented in the paper, the "SMASHTeam" team has faced with several difficulties mainly related to the coordination and collaboration issues that inevitably arise when several and geographically distributed teams are jointly involved in a project with tight deadlines.
The "SMASHTeam" as well as the teams of the universities of Bordeaux (France), Brunel (London, UK) and the Faculdade de Engenharia de Sorocaba, FACENS (Brazil) developed their SEE Federates by using the *SEE HLA Starter Kit*. Through the features provided by the Kit, the teams were able to improve the reliability of their SEE Federates and reduce the problems arising during the final integration and testing phases of the SEE event.
Moreover, the "SMASHTeam" offered support to the above mentioned teams, during all the steps of the SEE project, by providing ready-to-use examples, video tutorials and technical documentation for using the Kit.
The Team is currently working on the further experimentation, refinement and enhancement of the proposed solutions with the aim to participate to the next edition of SEE. Furthermore, the Team is focusing on the new version of the SEE HLA Starter Kit in which new services, such as the transfer of ownership of Objects and communication protocols over the standard Publish and Subscribe (P/S) mechanism, will be developed.
Hopefully, the experience and related solutions shared by the "SMASHTeam" of the University of Calabria Team could encourage and guide the participation of new teams to the next editions of this exciting and challenging event.

ACKNOWLEDGMENTS

The authors would like to thank Edwin Z. Crues (NASA JCS) for his precious advice and suggestions in the development of the *HLA Development Kit* (The HLA Development Kit project) and its SEE specific extension (the *SEE HLA Starter Kit*). A special note of thanks goes also to all the NASA staff involved in the SEE Project: Priscilla Elfrey, Stephen Paglialonga, Michael Conroy, Dan Dexter, Daniel Oneil, to Björn Möller (PITCH Technologies), and to all the members of SEE teams.

Authors would also like to thank the other members of the "SMASHTeam" of the University of Calabria: Marco Inzillo, Rina Mary Mazza and Andrea Tundis.

REFERENCES

3D Game Engine, Unity3D. Available from: http://unity3d.com [accessed 10 May 2016].

Anagnostou A., Chaudhry N.R., Falcone A., Garro A., Salah O., Taylor S.J.E., 2015. Easing the development of HLA Federates: the HLA Development Kit and its exploitation in the SEE Project. Proceedings of the 19th IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications (ACM/IEEE DS-RT 2015), pp. 50-57. October 14-16, Chengdu (China).

Anagnostou A., Chaudhry N.R., Falcone A., Garro A., Salah O., Taylor S.J.E., 2015. A Prototype HLA Development Kit: Results from the 2015 Simulation Exploration Experience. Proceedings of the ACM SIGSIM Conference on Principles of Advanced Discrete Simulation (ACM SIGSIM PADS 2015), pp. 45-46. June 10-12, London (United Kingdom).

Autodesk 3DS MAX. Available from: http://www.autodesk.com/products/3ds-max/ [accessed 10 May 2016].

Banks J., 1998. Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice. Wiley, New York (USA).

Falcone A., Garro A., 2016. The SEE HLA Starter Kit: enabling the rapid prototyping of HLA-based simulations for space exploration. Proceedings of the Simulation for Planetary Space Exploration (SpringSim-SPACE), Part of the 2016 Spring Simulation Multi-Conference (SpringSim 2016). April 3-6, Pasadena (California, USA).

Falcone A., Garro A., 2015. On the integration of HLA and FMI for supporting interoperability and reusability in distributed simulation. Proceedings of. the Symposium on Theory of Modeling and Simulation - DEVS Integrative M and S Symposium, DEVS 2015, Part of the 2015 Spring Simulation Multi-Conference (SpringSim 2015), pp. 9-16. April 12-15, Alexandria (Virginia, USA).

Falcone A., Garro A., Longo F., Spadafora F., 2014. Simulation Exploration Experience: A Communication System and a 3D Real Time Visualization for a Moon base simulated scenario. Proceedings of. the 18th IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications (ACM/IEEE DS-RT 2014), pp. 113-120. October 1-3, Toulouse (France).

Falcone A., Garro A., Tundis A., 2014. Modeling and Simulation for the Performance Evaluation of the On-Board Communication System of a Metro Train. Proceedings of the 13th International Conference on Modeling and Applied Simulation, (MAS 2014). September 10-12, Bordeaux, (France).

Fortino G., Garro A., Mascillaro S., Russo W., 2007. ELDATool: A Statecharts-based Tool for Prototyping Multi-Agent Systems. Proceedings of the WOA 2007 - 8th AI*IA/TABOO Joint Workshop "From Objects to Agents": Agents and Industry: Technological Applications of Software Agents, pp. 14-19. September 24-25, Genoa (Italy).

Fortino G., Garro A., Russo W., 2004. From Modeling to Simulation of Multi-Agent Systems: an integrated approach and a case study. Proceedings of. the Second German Conference on Multiagent System Technologies, Lecture Notes in Artificial Intelligence (LNAI) vol. 3187, pp. 213–227, Springer-Verlag, Berlin (Germany).

Fujimoto R.M., 2000. Parallel and distributed simulation systems. Wiley, New York (USA).

IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Federate Interface Specification, IEEE Standard 1516-2010.

Kuhl F., Weatherly R., Dahmann, J., 1999. Creating Computer Simulation Systems: An Introduction to the High Level Architecture. Prentice Hall.

Modeling and simulation coordination office. Available from: http://www.msco.mil [accessed 10 May 2016]

Möller, B., 2013. The HLA Tutorial v1.0. Pitch Technologies.

Taylor S.J.E., Revagar N., Chambers J., Yero M., Anagnostou A., Nouman A. and Chaudhry N.R., 2014. Simulation Exploration Experience: A Distributed Hybrid Simulation of a Lunar Mining Operation. Proceedings of the 18th IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications (ACM/IEEE DS-RT 2014), pp. 107-112. October 1-3, Toulouse (France).

The HLA Development Kit project. Available from: https://smash-lab.github.io/HLA-Development-Kit/ [accessed 10 May 2016]

The SEE HLA Starter Kit project. Available from: https://code.google.com/archive/p/see-hla-starterkit/ [accessed 10 May 2016]

The Simulation Exploration Experience project. Available from: http://www.exploresim.com [accessed 10 May 2016]

AUTHORS BIOGRAPHY

Alberto Falcone

Alberto Falcone received a Bachelor degree and a M.Sc. degree in Computer Engineering from the University of Calabria (Italy), in 2007 and 2011, respectively. He received a Master title in Industrial Research from the same institution in 2013. He is a PhD student in Information and Communication Engineering for Pervasive Intelligent Environments at Department of Informatics, Modeling, Electronics and Systems Engineering (DIMES) at University of Calabria. His main research interests include Modeling and Simulation of Systems. He is currently Visiting Researcher (from January to October 2016) at NASA Johnson Space Center (JSC), working with the Software, Robotics, and Simulation Division (ER). He has been involved since 2014 as a Team Leader in the SEE project.

Alfredo Garro

Alfredo Garro is an Associate Professor of Computer and Systems Engineering at the Department of Informatics, Modeling, Electronics and Systems Engineering (DIMES) of the University of Calabria (Italy). He is currently Visiting Professor (from January to October 2016) at NASA Johnson Space Center (JSC), working with the Software, Robotics, and Simulation Division (ER). From 1999 to 2001, he was a researcher at CSELT, the Telecom Italia Group R&D Lab. From 2001 to 2003, he worked with the Institute of High Performance Computing and Networking of the Italian National Research Council (CNR). On February 2005 he received the PhD Degree in Systems and Computer Engineering from the University of Calabria. From January 2005 to December 2011, he was an Assistant Professor of Computer and Systems Engineering at the DIMES Department (formerly DEIS) of the University of Calabria. His main research interests include: Modeling and Simulation, Systems and Software Engineering, Reliability Engineering. His list of publications contains about 100 papers published in international journals, books and proceedings of international and national conferences. In 2014, he founded the Departmental Research Laboratory "System Modeling And Simulation Hub Lab (SMASH Lab)". He is vice chair of the Space Reference Federation Object Model (SRFOM) Product Development Group (PDG) of SISO. He is the Technical Director of the "Italian Chapter" of INCOSE (International Council on Systems Engineering). He is a member of the Executive Committee and National Coordinator for Italy in the MODRIO European Project. He is the Technical Leader for UNICAL in the Open Source Modelica Consortium (OSMC). He is a Member of the CINI National Lab on Cyber Security and of the Technological District on Cyber Security (DCS). He is the Faculty Advisor and Member of the Executive Committee of the Simulation Exploration Experience (SEE) project. He is involved in the activities of the IEEE Computer Society, IEEE Reliability Society and IEEE Aerospace and Electronic Systems Society.