Proceedings of the Operational Research Society Simulation Workshop 2025 (SW25)
M. Luis, A. Harper, T. Monks and, N. Mustafee, eds.
https://doi.org/10.36819/SW25.005

WANT TO BUILD A MOONBASE?
THE SIMULATION EXPLORATION EXPERIENCE (SEE)

Maziar Ghorbani
Simon J. E. Taylor
Nura T. Abubakar
Hridyanshu Aatreya
Anastasia Anagnostou

Brunel University London
Modeling & Simulation Group
Department of Computer Science
Uxbridge, UB8 3PH, UK
maziar.ghorbani@brunel.ac.uk

Damon Curry

Pitch Technologies US
37 North Orange Avenue
Suite 545
Orlando, Florida FL, 32801 USA
damon.curry@pitchtechologies.com

ABSTRACT

The Simulation Exploration Experience (SEE) is an annual international initiative that brings together teams of students, researchers, and professionals to collaboratively design and implement distributed simulations for space exploration. The project allows student teams from universities worldwide to develop and integrate simulations of critical systems such as transportation, communication, and resource management. This paper outlines how students design and contribute to different aspects of the moonbase simulation, utilizing High-Level Architecture (HLA) for simulation interoperability and integrating 3D visualizations. Additionally, it highlights the educational benefits of the SEE, offering students hands-on experience in distributed simulation technologies, space mission planning, and collaborative problem-solving, with support from NASA volunteers, the Simulation Interoperability Standards Organization (SISO), and other industry partners.

1 INTRODUCTION

For over a decade, the Simulation Exploration Experience (SEE) (SEE, 2019, 2024) has run annually, enabling teams of researchers, graduates, and undergraduate students to collaborate in building a distributed simulation of a moonbase and its associated elements, such as drones, robots, spaceships, and satellite networks. SEE is supported by volunteers from NASA, the Simulation Interoperability Standards Organization (SISO) (Möller et al., 2020), universities, and industry partners. Student teams from around the world gain hands-on experience with distributed simulation technologies and develop expertise in the High-Level Architecture (HLA) for simulation interoperability (Taylor, 2019).

Distributed simulation (Okan Topçu, 2017) is a method of simulating complex systems by integrating multiple independent components across different locations. These simulations rely on a networked framework, where each component (or federate) interacts in real-time while being managed by a central simulation infrastructure. Distributed simulation is widely used in defence, aerospace, and space exploration, where large-scale, interconnected systems must operate cohesively (Dahmann, Fujimoto, and Weatherly, 1997). SEE provides students with practical exposure to these concepts, allowing them to experiment with HLA, a key standard for enabling simulation interoperability.

This paper provides an overview for educators and students about the SEE and the process of creating a distributed simulation of a lunar habitat using HLA. Section 1 introduces the SEE program and its goals. Section 2 provides background information on HLA exploring the technologies used in the SEE project. Section 4 discusses the Baseball Card and Interaction Matrix as tools for federate conceptualization and communication. Section 5 guides readers on getting started with SEE and running a federate locally. Section 6 provides more details on transitioning from local to remote communications. Finally, Section 7 concludes the paper with key takeaways.

2 BACKGROUND

The High-Level Architecture (HLA) was developed to simplify and standardise the creation of distributed and interoperable simulations. It defines a set of standards that ensure different simulation components can communicate effectively. These include integration rules, a Runtime Infrastructure (RTI) for managing data exchange (IEEE, 2010b), an Object Model Template (OMT) for structuring shared data, and a development process to guide implementation (IEEE, 2010a).

The Simulation Interoperability Standards Organization (SISO) (Möller et al., 2020) serves as the primary community supporting and advancing HLA. Like many industry standards, HLA benefits from an active community that drives its evolution. SISO plays a key role in maintaining and refining the standard, with a new version currently in development (Crues et al., 2022), while also contributing to the creation of domain-specific standards to enhance simulation interoperability across different fields.

Figure 1 illustrates the key components of an HLA-based distributed simulation, including the federates and their interactions. One crucial element shown is the RTI Ambassador, which serves as the interface between a federate and the RTI. The RTI Ambassador allows federates to send and receive simulation data, interact with other federates, and synchronize execution according to HLA rules.

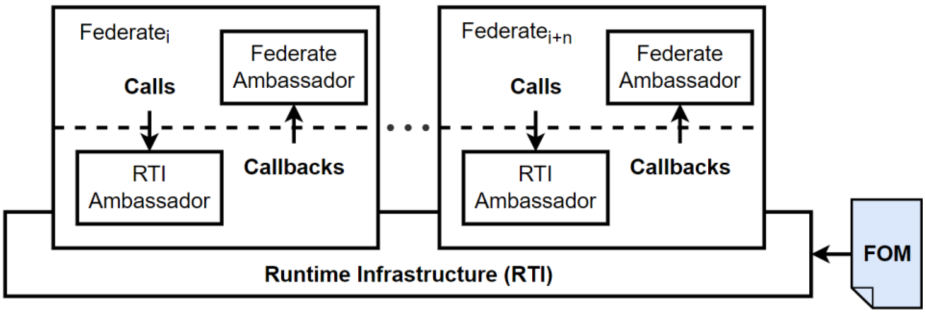

Figure 1: Components of an HLA-based Distributed Simulation

Interaction between the federates is via communicated data exchange (e.g., messages) which are defined in each Federate Object Model (FOM) based on the general OMT. Federates interact through the HLA RTI middleware and the RTI is responsible for the overall synchronization and control of the federation. For example, there could be an HLA-based distributed simulation of several drones exploring the moon's surface. Each federate simulates its own drone and reacts to the positions of the others. Before performing its simulation for the next period, a federate will "ask" the RTI if there are any updates of the position of the drones from the other federates (as defined by the FOM). If there are updates, the RTI passes these to the federate and then that federate updates its simulation (e.g., the new positions of the other drones). The federate will then perform its simulation and then inform the RTI of the new position of its own drone. The other federates will do this in parallel at the same time. Federates can be on the same or different geographically dispersed computers (SEE has federates running in several different countries).

Despite the benefits of HLA, developing a distributed simulation remains a complex task. The HLA shares some conceptual similarities with Service-Oriented Architecture (SOA) (Laskey & Laskey, 2009), particularly in how distributed components interact. However, while SOA is commonly taught in undergraduate and master's computer science programs, HLA is less frequently included in standard curricula, as its primary focus is on simulation interoperability rather than general software architecture.

Unlike topics such as networking or SOA, HLA is not widely covered in traditional degree programs, and dedicated courses on HLA tend to appear at the postgraduate level. Some universities offer specialized MSc courses that include HLA as part of their distributed simulation or modelling and simulation programs. Examples include the MSc in Modelling & Simulation at the Naval Postgraduate School (USA) and the MSc in Computational Science and Engineering at Cranfield University (UK), both of which include elements of HLA-based distributed simulation in their coursework.

There are very few opportunities to study HLA in traditional Operational Research (OR) degree programs, as OR primarily focuses on optimisation, decision-making models, and analytical methods rather than distributed simulation frameworks. However, some OR applications, such as military logistics, supply chain optimisation, and emergency response planning, have started integrating simulation-based approaches, including HLA-driven distributed simulations, to model complex systems more effectively.

Recognizing this gap, SEE was established in 2011 (originally as the Simulation Smackdown) to provide both postgraduate and undergraduate students with hands-on experience in HLA-based distributed simulation. Beyond technical training, the program was also designed to foster a uniquely international collaboration, enabling students from diverse backgrounds to work together on the future of space exploration.

SEE has had scenarios on the Moon, Cislunar Space, and Mars. Cislunar space refers to the region of space between Earth and the Moon, including the Moon's orbit and areas where future space operations, such as lunar gateways and orbital refuelling stations, may be established.

As space exploration evolves, SEE has aligned its objectives with NASA's Artemis plan, marking a shift towards future-oriented technologies, systems, and capabilities expected to emerge by the mid-2050s. This transition is reflected in SEE 2024's scenario, which focuses on establishing lunar infrastructure while laying the groundwork for future Martian expeditions, in alignment with NASA's Moon to Mars Campaign Strategy.

To support these simulations, SEE teams have access to a wide range of distributed simulation technologies and software solutions. These tools, which include both open-source and commercial software, enable teams to build, test, and refine their simulations effectively. Many commercial providers generously donate software licenses for the duration of SEE, allowing participants to explore innovative simulation environments. The following section provides an overview of the key technologies available to SEE teams.

Various RTI implementations exist, including both open-source and commercial options. For example, Pitch Technologies provides Pitch pRTI, a licensed HLA RTI that is fully compliant with the HLA standard. SEE teams benefit from Pitch's support, which includes access to Pitch pRTI and additional productivity tools such as Pitch Visual OMT and Pitch Commander (Pitch Technologies, 2024).

HLA RTI (Open Source and Commercial): Various RTI implementations exist, including both open-source and commercial options. For example, Pitch Technologies provides Pitch pRTI, a licensed HLA RTI that is fully compliant with the HLA standard. SEE teams benefit from Pitch's support, which includes access to Pitch pRTI and additional productivity tools such as Pitch Visual OMT and Pitch Commander (Pitch Technologies, 2024). While commercial RTI solutions like Pitch pRTI are widely used in industry and supported in SEE, open-source alternatives are also available. Teams that learn the fundamental concepts of HLA using licensed tools can later transition to open-source RTI implementations, such as TrickHLA or other freely available HLA-compliant solutions. This flexibility allows teams to continue developing HLA-based simulations beyond their participation in SEE.

HLA Starter Kit (Open Source): The SEE HLA Starter Kit was designed to streamline the development of HLA federates within the SEE project. This kit includes the Starter Kit framework (SKF) for Java-based federate development, along with comprehensive technical documentation, user guides, and reference examples (Falcone et al., 2017).

TrickHLA (Open Source): TrickHLA, a middleware solution supporting the IEEE 1516 HLA simulation interoperability standard within the Trick Simulation Environment. TrickHLA simplifies the integration of HLA into simulations by abstracting away the complexities of distributed simulation, allowing users to focus on simulation development without requiring extensive HLA expertise (NASA, 2024c).

Trick Simulation Environment (Open Source): The Trick Simulation Environment, originating from NASA's Johnson Space Center, stands as a robust framework for simulation development, offering a versatile platform for constructing applications across various stages of space vehicle development. With Trick, users can efficiently generate simulations for diverse purposes, including initial vehicle design, performance assessment, flight software development, dynamic load analysis of flight vehicles, and training in virtual or hardware-in-the-loop environments (NASA, 2024b).

JSC Engineering Orbital Dynamics (Open Source): JSC Engineering Orbital Dynamics (JEOD) is a sophisticated simulation tool tailored to integrate with the NASA Trick Simulation Environment. It specializes in generating vehicle trajectories through the solution of complex numerical dynamical models, segmented into four key categories. Environment models encapsulate the vehicle's surrounding conditions, while Dynamics models focus on integrating equations of motion. Interaction models represent the vehicle's interactions with the environment, complemented by a suite of mathematical and orbital dynamics Utility models (NASA, 2024a).

Distributed Observer Network (DON): DON is a data presentation tool developed by NASA to visualize and showcase simulation outcomes. By leveraging modern gaming technology, DON provides an interactive 3D environment where users can observe the evolution and interactions of graphical models within a simulated scenario (NASA, 2024a). DON is available through a license request from NASA, allowing SEE teams to access and integrate it into their simulations. However, other data streaming and visualization tools can also be used to achieve similar functionality. Open-source alternatives and commercial solutions exist for streaming simulation data and creating real-time 3D visualizations, providing flexibility for teams that may not have access to DON.

3 PARTICIPATING IN THE SIMULATION EXPLORATION EXPERIENCE

The SEE term runs from January to March/April with weekly 2–3-hour sessions. A live virtual event (takes place at the end of SEEE, where the teams join to run their distributed simulations. Onboarding begins in September, during which teams can get advice about the requirements and the technologies involved in SEE. Teams interested in participating in SEE first look at the documents and videos available at Simulation Exploration Experience Website (SEE, 2024), where they can see examples and details of the scenario and details on how to apply. Brunel's How-to Guide (Taylor et al., 2023) also provides further explanation.

Brunel began developing its SEE simulation in 2022 with a conceptual design, followed by the creation of a simple Lunar Rover Federate in 2023. Building on this foundation, the team expanded the simulation to include four federates in 2024, representing key on-site lunar base facilities: a Communications Hub, Storage Depot, Launch Complex, and Habitation Quarters. The HLA federate codebase and 3D models can be found at (Ghorbani, 2024).

In terms of project management, the SEE program utilizes Businessmap (Kanbanize, 2024), a software platform designed to streamline workflows through Kanban-based task management. Formerly known as Kanbanize, Businessmap enables teams to visualize their progress, automate workflows, and track key performance metrics, helping them maintain efficiency throughout the project lifecycle. By using customizable boards, task dependencies, and automated notifications, SEE participants can coordinate tasks, manage resources effectively, and ensure adherence to project milestones and deadlines.

Active teams participating in SEE follow a structured process to develop, test, and integrate their federates within the distributed simulation environment. The journey begins with the conceptualization phase, where each team defines the key functionalities, data elements, and interactions of their federates. This is documented using a Baseball Card, a summary format that outlines the role and behavior of each federate in the simulation.

Once the conceptual groundwork is established, teams proceed to set up the necessary software infrastructure. They first request and install NASA's DON software, which allows them to monitor and interact with simulations across the SEE network. Following this, an Interaction Matrix is developed to map out the relationships between federates, ensuring clear communication and synchronization within the simulation. The RTI software is then installed, enabling federates to exchange data efficiently within the HLA framework.

With the software environment in place, teams move on to initial testing and connectivity setup. They begin by running a simple federate system, locally, verifying its basic functionality and ensuring that it can operate within the intended simulation parameters. Once this is confirmed, they connect to the SEE Virtual Private Network (VPN), which provides a secure connection for federates to access the SEE networks, ensuring reliable remote communications to the federation infrastructure.

The next phase involves collaborative testing and integration. Teams conduct interoperability tests using RTI, ensuring their federates function correctly within the larger simulation. To prevent any technical issues, a comprehensive dry run is organized before the final event. During this phase, teams refine their federates, resolve potential integration challenges, and make necessary adjustments based on peer feedback. This is followed by 3D model integration, where the federates' functionality is validated in conjunction with visual representations, ensuring seamless interaction between the simulation components.

The SEE experience culminates in the final event, where all teams deploy their federates in a live, distributed simulation. This event serves as a showcase of their technical achievements, collaboration, and problem-solving efforts throughout the SEE program. By following this structured approach, teams gain hands-on experience with distributed simulation technologies, enhancing their understanding of HLA-based interoperability while contributing to the broader simulation ecosystem.

During the early stages of SEE, teams define objectives, coordinate with others, and engage in active learning through discussions and problem-solving. Weekly meetings facilitate peer-to-peer feedback, helping students refine their skills while fostering collaboration and knowledge-sharing. A key aspect of SEE's pedagogy is its mentorship model, where experienced teams guide newcomers, supported by volunteers from NASA, SISO, and industry partners. This structured yet flexible learning environment bridges the gap between theory and practice, while guest speakers introduce students to advanced topics in Modeling & Simulation, enriching their exposure to real-world applications.

4 BASEBALL CARD AND INTERACTION MATRIX

The "Baseball Card" has become a key part of a team's activities as it serves as a tool for federate conceptualization, documentation, and presentation. Usually, there is one Baseball Card per federate as shown in Figure 2:

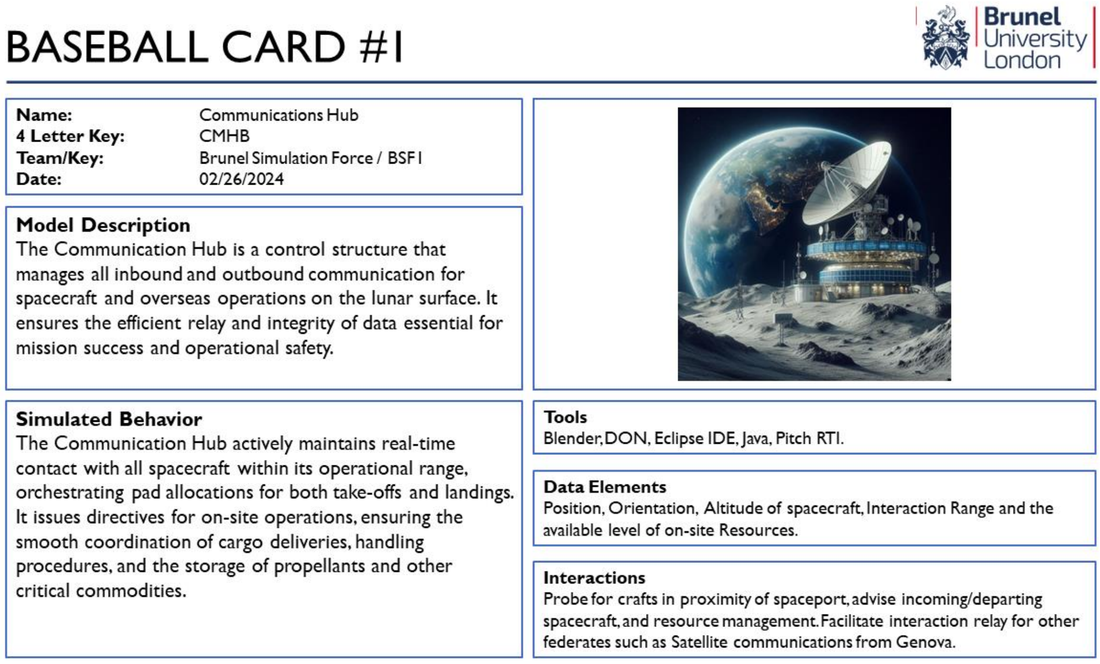

Figure 2: Brunel University Baseball Card for a Communications Hub

The Baseball Card serves as both a documentation and communication tool for teams participating in SEE. It provides a structured way to conceptualize and present each federate, ensuring clarity in design, functionality, and interactions. Each Baseball Card captures key aspects of a federate, allowing teams to quickly understand the role and behaviour of different simulation components.

At its core, the Baseball Card introduces the team and federate details, offering a snapshot of the federates identity and its intended function within the SEE simulation. It then provides a model description, explaining the federates primary objectives, operational scope, and the rationale behind its development. This is complemented by an overview of the simulated behaviour, which details how the federate interacts within the distributed simulation environment, including its responses to external inputs and its engagement with other federates.

To enhance clarity and engagement, the Baseball Card incorporates visual elements, such as diagrams or images, that illustrate the federates design and functionality. Additionally, it outlines the software tools and technologies used in the federates development, ensuring transparency in the simulation's technical foundation. The data elements section specifies the inputs and outputs that the federate processes, highlighting how it exchanges information within the simulation framework. Finally, the interactions section maps out the federates connections with other teams and federates, providing insight into its collaborative role in the broader SEE ecosystem. SEE employs an Interaction Matrix to map the interaction between simulations. Structured in a tabular format, it outlines the potential communications and dependencies across federates and helps to ensure a coherent and synchronized federation as well as giving a clear picture of what interacts with what. The first row and column of the matrix list the names of the federates involved, with the intersecting cells designated to detail the specific interactions between these entities.

The Interaction Matrix plays a crucial role in defining the relationships between federates within the SEE simulation, ensuring smooth communication and coordination. A prime example of this is the Brunel team's Communication Hub, which serves as a central node for coordinating interactions among various simulation components of the moonbase.

One key interaction involves the University of Genova, which has developed a lunar satellite designed to relay critical data such as altitude, longitude, orbital periods, and regular status updates. This satellite acts as a communication bridge between rockets and landers, transmitting essential information through the Communication Hub. Similarly, FACENS has focused on developing a Cable Car system, which facilitates passenger transport between the Spaceport and the Habitation Quarters. To ensure seamless transit operations, the Cable Car team relies on the Communication Hub to transmit schedule updates and arrival times, enabling real-time coordination within the lunar environment. Additionally, the University of Central Florida (UCF) has contributed to the development of a lunar rover. The lunar rover enables exploration and logistics support within the simulation. This system integrates with the Communication Hub, allowing for data exchange and improved operational efficiency.

Beyond these specific examples, multiple teams working on various lunar vehicles also integrate with the Communication Hub. These vehicles depend on the hub to facilitate essential data exchanges, ensuring consistent and reliable communication across the simulation. By mapping these interactions, the Interaction Matrix provides a structured framework for managing interoperability between federates, ultimately enhancing the realism and functionality of the SEE simulation. The interaction matrix has evolved to encapsulate the internal dynamics of our federates and incorporate external interactions with federates from other teams.

5 GETTING STARTED

Participating in SEE is a continuous learning process, where teams develop expertise in distributed simulation through hands-on experience. Mastering the SEE Starter Kit Framework (SKF) is a crucial step for teams choosing to use it, as it simplifies HLA-based space simulations by providing structured tools and functionalities (Falcone et al., 2017; Falcone & Garro, 2016). SKF accelerates development, reduces technical barriers, and enables seamless federate integration.

However, SEE teams have flexibility in selecting their simulation tools. While our team has chosen to explore SKF, other teams may opt for alternatives such as TrickHLA, which also supports HLA-based simulation development. The choice of tools depends on team objectives, experience, and technical requirements. This section focuses on SKF, but similar principles apply to other frameworks used within the SEE ecosystem.

A fundamental component of SKF is its Reference Frames module, which manages the International Celestial Reference Frame (ICRF), a fixed coordinate system based on distant quasars. This ensures precise orbital mechanics and spacecraft trajectory simulations. The Celestial Bodies module supports space object representation, allowing teams to create planetary models, while the Space Time Manager synchronizes epochs across time standards (e.g., TT, UTC, J2000). SKF also includes Mathematical Utilities to assist in spatial and temporal calculations, ensuring accuracy in federate behaviour.

SKF's application layer simplifies development through core classes such as *SEEAbstractFederate* and *SEEAbstractAmbassador*, allowing teams to focus on federate behaviour rather than low-level RTI management. Federate operations are divided into proactive lifecycle management and reactive RTI interactions, ensuring smoother execution and interoperability.

Understanding RTI is essential for simulation coordination. The Central RTI Component (CRC) orchestrates federation operations, offering both graphical and command-line interfaces for monitoring. RTI APIs support multiple languages (C++, Java, or wrappers for other languages), providing flexibility in federate development. Teams can also integrate the SISO Space Reference Federation Object Model (Space FOM) (Möller et al., 2020) for standardized federate interactions.

SEE participants receive access to SKF tutorials and a presentation from SKF developers on best practices. Using Eclipse IDE, teams set up federates, integrating essential libraries (booster1516.jar, prti.jar), configuration files, and logging tools. The *LunarRoverFederate* class, part of the SKF Starter Kit, allows teams to drive a simple lunar rover and share its coordinates.

Federate Configuration & Execution

To join the SEE Federation, federates:

1. Configure the SKF environment and apply settings for HLA integration.
2. Connect to the RTI, specifying the CRC host and port (e.g., MAK, Pitch Technologies).
3. Join the SEE Federation execution, subscribing to required updates.
4. Synchronize with the federation timeline and confirm readiness.
5. Manage execution control, updating configurations and publishing mode transitions.
6. Handle object and interaction management, ensuring seamless data exchange.
7. Set up HLA Time Management, aligning simulation timing and synchronization points.
8. Launch simulation execution, integrating federates into the SEE distributed environment.

To visualize 3D objects, the simulation interoperates with the Distributed Observer Network (DON) using *SpaceMaster*, *SpaceReferenceFramePublisher*, and *HLA2DON*. *SpaceMaster* ensures synchronization across federates, while *SpaceReferenceFramePublisher* maps spatial interactions to DON, allowing for real-time visualization. *HLA2DON* further facilitates the integration of HLA simulation data into the DON environment, ensuring seamless data translation and rendering.

All these tools are provided by SEE as part of its GitHub repository, giving teams access to essential software components for integrating their federates with DON and enhancing the visual representation of simulation interactions. This structured process ensures federates are accurately integrated, synchronized, and effectively visualized within the SEE environment.

Our team is actively working on developing additional federates to expand the scope of our SEE participation. While these efforts go beyond the scope of this edition, we plan to document our experiences, techniques, and lessons learned in future editions. This will provide deeper insights into the challenges, methodologies, and best practices we have explored in advancing HLA-based simulations within SEE.

6 RUNNING A FEDERATE REMOTELY

To operate a federate remotely within the SEE Federation, teams must configure their system to establish a secure network connection while ensuring seamless integration into the distributed simulation environment. This process begins with VPN activation, which provides a secure communication channel, allowing teams to access the SEE network. However, HLA remains responsible for managing the actual interactions between federates, ensuring synchronisation and data exchange within the simulation.

Once the secure network connection is established, teams must configure their federate settings to ensure proper integration into the SEE Federation. This involves updating the local settings designator, which previously referenced local or intranet CRC addresses, to the designated IP address or hostname provided by the SEE Network Admin. This adjustment allows the federate to access the SEE network infrastructure and establish communication.

With these configurations in place, the federate is ready to join the SEE Federation’s distributed simulation. This step follows the same setup as the local configuration but now operates within the federation’s broader network. Connecting to the federation enables teams to participate in scheduled dry run tests, which help identify and resolve potential issues while ensuring interoperability before the final event. Once testing is complete, the federate is fully prepared to integrate into the live distributed simulation, contributing to the SEE program's collaborative efforts.

A properly configured federate ensures a seamless transition from local development to full-scale remote operation. Aligning the federate settings with the remote CRC requirements and participating in dry run tests allows teams to validate functionality and troubleshoot early. This structured approach ensures readiness for both the dry runs and the final event, where teams fully integrate into the SEE real-time distributed simulations event.

7 CONCLUSION

This tutorial has provided an overview of the Simulation Exploration Experience (SEE), offering insights into its scope and the challenges new teams may encounter. SEE presents a unique opportunity for students to engage with distributed simulation technologies, develop HLA-based interoperability skills, and collaborate on an international scale. While the learning curve can be steep, the enthusiastic support from volunteers, experienced teams, and industry professionals ensures that newcomers can quickly adapt and contribute meaningfully to the project.

Beyond technical skill development, SEE serves as a valuable educational model for simulation pedagogy. It fosters an environment of experiential learning, where students acquire not only technical expertise but also teamwork, problem-solving, and project management skills. The mentorship model within SEE, where experienced teams assist new participants, reinforces the importance of peer learning and knowledge-sharing. Furthermore, by working in a distributed, real-world simulation setting, students gain exposure to collaborative software development practices and the challenges of synchronization, debugging, and integration in complex systems.

Reflecting on our participation in SEE, several key lessons have emerged that can benefit future teams. Early preparation is critical, securing software licenses, installing tools, and troubleshooting technical issues early prevents delays later in the project. Team structure also plays a crucial role in efficiency; assigning roles based on individual skills and interests (e.g., software development, 3D modeling, and simulation design) enhances productivity. Regular communication, both internally and with other SEE teams, is essential for successful HLA interactions, as early coordination ensures a clear roadmap for collaboration. Additionally, providing structured support for new members, such as software templates and introductory tasks, helps them integrate more effectively. Finally, continuous testing and validation are vital to ensure that the simulation performs as expected during the final event.

Brunel University's participation in SEE illustrates these principles in action. A team of second-year computer science students successfully completed most of their objectives while collaborating with 10-15 teams from various countries. While many SEE participants are postgraduate students, the program remains an invaluable learning experience for students at all levels, providing hands-on exposure to innovative simulation technologies and an opportunity to engage with a global community of researchers and engineers.

We encourage you to watch the latest SEE final event video (SEE Event, 2024) to witness the creativity and achievements of SEE teams.

ACKNOWLEDGMENTS

We thank the SEE team volunteers for providing technology, expertise and technical support. We also acknowledge the excellent work done by all teams from all SEE "years"!

REFERENCES

Ghorbani, M. (2024). SEE-GitHub. https://github.com/mzrghorbani/SEE2024

Crues, E. Z., Dexter, D. E., Möller, B., Falcone, A., & Garro, A. (2022). SpaceFOM: An Interoperability Standard for Space Systems Simulations. 2022 IEEE Aerospace Conference (AERO), 1–10. https://doi.org/10.1109/AERO53065.2022.9843262

Dahmann, J. S., Fujimoto, R. M., & Weatherly, R. M. (1997). The Department of Defense High Level Architecture. Proceedings of the Winter Simulation Conference (WSC), 142–149. https://doi.org/10.1145/268437.268465

Falcone, A., Garro, A., Taylor, S. J. E., Anagnostou, A., Chaudhry, N. R., & Salah, O. (2017). Experiences in Simplifying Distributed Simulation: The HLA Development Kit Framework. Journal of Simulation, 11(3), 208–227. https://doi.org/10.1057/s41273-016-0039-4

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

IEEE. (2010b). IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)–Framework and Rules.

Kanbanize. (2024). Businessmap. https://businessmap.io/

Laskey, K. B., & Laskey, K. (2009). Service oriented architecture. WIREs Computational Statistics, 1(1), 101–105. https://doi.org/10.1002/wics.8

Möller, B., Rydell, A., Crues, E. Z., Dexter, D., Falcone, A., & Garro, A. (2020). SISO Space Reference FOM-Tools and Testing. SISO 2020 Simulation Innovation Workshop, JSC-E-DAA-TN76473.

NASA. (2024a). Trick Simulation Environment. https://github.com/nasa/trick

NASA. (2024b). TrickHLA. https://github.com/nasa/TrickHLA

NASA. (2024c). DON. https://public.ksc.nasa.gov/don/

NASA. (2024d). JEOD. https://github.com/nasa/jeod/

Okan Topçu, H. O. (2017). Guide to Distributed Simulation with HLA (H. O. Okan Topçu, Ed.). Springer.

Pitch Technologies. (2024). Pitch2024. https://pitchtechnologies.com/products/

SEE. (2019). SEE 2019. https://www.liophant.org/conferences/2022/see/

SEE. (2024). SEE 2024. https://simulationexplorationexperience.org/

Taylor, S. J., Anagnostou, A., Abubakar, N., Saha, A., & Ghorbani, M. (2023). Simulation Exploration Experience: How-to Guide. Available at: https://brunel.figshare.com/articles/conference_contribution/Simulation_Exploration_Experience_How-to_Guide/26293426?file=47659741

Taylor, S. J. (2019). Distributed Simulation: State-of-the-Art and Potential for Operational Research. European Journal of Operational Research, 273(1), 1–19. https://doi.org/10.1016/j.ejor.2018.04.032

SEE Event 2024. Simulation Exploration Experience. https://www.youtube.com/live/7Bcr8zwpZVY?si=Si0JZtekbNpj34dc

AUTHOR BIOGRAPHIES

MAZIAR GHORBANI is an Associate Lecturer in Computer Science at Brunel University London, holding a PhD in Experimental Physics (2022). His research spans mathematics, engineering, computer science, and physics, contributed to High-Luminosity LHC Upgrade, STAMINA, SEAVEA, and ITFLOWS projects. He co-leads the SEE team, applying his expertise in simulation technologies. His email is maziar.ghorbani@brunel.ac.uk.

SIMON J. E. TAYLOR is a Full Professor and Co-Director of the Modeling & Simulation Group at Brunel University London. His research has impacted millions of students and hundreds of universities, as well as SMEs and large enterprises, Ford and Sellafield. He co-founded the Journal of Simulation, the UK Simulation Workshop series, and serves as Executive Chair of SEE. He is a former ACM SIGSIM Chair and Program Chair for WSC 2025. His email is simon.taylor@brunel.ac.uk.

NURA TIJJANI ABUBAKAR is an Associate Lecturer in Computer Science at Brunel University London. His research focuses on inter-cloud distributed simulation, and he has co-led SEE for several years. He previously served as Head of the Department of Innovation Technology at Jigawa State Institute of Information Technology, Nigeria. He has contributed to STAMINA, HORIZON2020, and Science Gateways Africa. His ORCID ID is https://orcid.org/0000-0002-4216-3057, and his email is nura.abubakar2@brunel.ac.uk.

HRIDYANSHU AATREYA is a student and Associate Teaching Assistant at Brunel University London. His research interests include operating systems, computer networks, and distributed simulation. In SEE 2024, he was the project architect for the Brunel team, focusing on HLA frameworks and gaming technology integration. His email is 2200096@brunel.ac.uk.

DAMON CURRY has 30 years of experience in distributed simulation, 3D visualization, terrain, and simulated radio. He is President of Simulation Exploration Experience Inc., North America Business Development Manager at Pitch Technologies, and President of SISO. He holds two patents in real-time image processing for VR. A former US Air Force engineer, he worked on cruise missiles, F-16 avionics, and radio systems. He holds a BSc in Electrical Engineering from The Ohio State University. His email is damon.curry@simexplore.org.

ANASTASIA ANAGNOSTOU is a Senior Lecturer at Brunel University London and Co-Lead of the Modeling & Simulation Group. Her research spans Advanced Computing for M&S, Open Science, and Healthcare Simulation. She has held leadership roles at WSC and serves as ACM SIGSIM Vice Chair and Associate Editor for the Journal of Simulation. She is Program Chair for WSC 2025. Her email is anastasia.anagnostou@brunel.ac.uk.