This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/WSC63780.2024.10838847, 2024 Winter Simulation Conference (WSC)

Proceedings of the 2024 Winter Simulation Conference
H. Lam, E. Azar, D. Batur, S. Gao, W. Xie, S. R. Hunter, and M. D. Rossetti, eds.

SIMULATION EXPLORATION EXPERIENCE (SEE) INTRODUCTORY TUTORIAL

Maziar Ghorbani[1], Anastasia Anagnostou[1], Nura T. Abubakar[1], Hridyanshu Aatreya[1], Damon Curry[2], and Simon J. E. Taylor[1]

[1]Modelling & Simulation Group, Dept. of Computer Science, Brunel University London, Uxbridge, UK
[2]Pitch Technologies US, Orlando, FL, USA

ABSTRACT

This paper presents an introductory tutorial based on the Simulation Exploration Experience (SEE) 2024, highlighting a collaborative effort by NASA, SISO and international academic partners to model lunar facilities and habitats through the High-Level Architecture (HLA) for distributed simulations. Focused on federating simulations of lunar infrastructure, this paper outlines methodical steps for creating and executing models that incorporate communication systems and 3D visualizations to support educational and research initiatives in space exploration. Reflecting on SEE 2024's advancements, the tutorial emphasizes significant progress in using simulation technology to promote innovation and collaboration across various scientific disciplines. This contribution, intended for discussion at the Winter Simulation Conference (WSC) 2024, showcases the role of HLA runtime infrastructure (RTI) in enabling realistic and interoperable simulation environments, enriching the discourse on simulation education.

1 INTRODUCTION

The Simulation Exploration Experience (SEE 2024) stands as a testament to the evolving landscape of educational methods, where collaborative design and advanced simulation technologies converge to create unique learning environments. This initiative, supported by NASA, the Simulation Interoperability Standards Organization (SISO) of academic and industry partners, offers students and educators a platform to explore the complexities of lunar habitat simulations using the HLA (IEEE 2010) for real-time, distributed simulations (Taylor 2019). At its core, the SEE program is not just about mastering the technicalities of simulation software or understanding the intricacies of space exploration; it is a comprehensive educational journey that challenges participants to think critically, collaborate effectively, and innovate within a structured yet dynamic environment.

Building on the insightful experiences from the SEE 2024, this paper presents a structured guide for educators and learners on designing and implementing simulated lunar habitats using HLA for distributed simulations. The guide has several key aims to clarify the process of creating engaging, realistic simulations that capture the challenges of lunar exploration and infrastructure building, to illustrate how HLA facilitates smooth interoperability and real-time teamwork among various simulation elements, and to employ advanced visualization techniques to elevate the learning experience and interaction. Furthermore, this paper intends to demonstrate the wide-ranging benefits of simulation technology in enhancing analytical thinking, and problem-solving skills, and fostering collaboration across disciplines. Through a detailed walkthrough for developing and executing moon simulations, this tutorial supports the broader objective of advancing simulation education and equipping learners for upcoming challenges in space exploration and related fields. This tutorial is structed as follows. Section 2 provides the background of the HLA and the SEE initiative and Section 3 discusses the technologies used in the SEE project. An example case study implementation is discussed in Section 4. Section 5 outlines how the assessment and evaluation are conducted and Section 6 concludes the paper.

This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/WSC63780.2024.10838847, 2024 Winter Simulation Conference (WSC)

2 BACKGROUND

In 1996, the Defense Modeling and Simulation Office (DMSO) produced an initial proposal for the HLA, a standard to support the reuse and interoperation of distributed simulations (Dahmann et al. 1997). This was formally ratified as the IEEE 1516-2000 Standard for Modeling and Simulation (M&S) High Level Architecture in 2000 (updated in 2010 and a new version is under development since January 2016). The HLA is actually a suite of standards. These include the definition of a framework and rules (IEEE 2010a), the definition of supporting software (the Runtime Infrastructure (RTI)) (IEEE 2010b), data definitions (the Object Model Template (OMT) (IEEE 2010c)) and a development process (IEEE 2010d). It is normal for a set of standards to have a supporting community. SISO has also produced many standards in support of the HLA (e.g., the SpaceFOM discussed later in this tutorial). Developing distributed simulations using the HLA requires that each simulation is a federate and the collection of interacting (or interoperating) simulations is a federation. The data that each simulation can send and expects to receive is defined in the federate's Federate Object Model (FOM) based on the general OMT (possibly extended by a variant dedicated to an application area such at the SpaceFOM). The FOM for each federate defines the variables and their types that can be updated. Federates interact through the RTI and the RTI is responsible for the overall synchronization and control of the federation. In distributed computing terms, all the common computing and communication services are placed in the RTI middleware. A limited number of commercially available RTI middleware and open source RTI middleware are available. Simplistically, an HLA-based distributed simulation works in the following way. Let us assume that there are 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 time 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.

The HLA is complex and requires expertise to develop federates and federations. SEE (then the Simulation Smackdown) was founded in 2011 as an attempt to develop skills in this area, particularly with a view to developing space simulations. SEE has been a cornerstone initiative in bridging the gap between academic theory and practical application in M&S. SEE employs the HLA standard to foster interoperability among disparate simulations, enabling the collaborative development of complex, large-scale simulated space missions encompassing the Moon, Cislunar Space, and Mars. This approach introduces participants to the intricacies of Real-Time Distributed Simulation (RTDS) that enhances their employability and technical skills through hands-on experience in a multidisciplinary, collaborative environment.

Reflecting the evolving landscape of space exploration, SEE has continually adapted its focus. Initially envisioning missions fifty years into the future, the program's trajectory pivoted in 2022 to align with NASA's Artemis plan, signaling a shift towards technologies, systems, and capabilities expected in the mid-2050s. SEE 2024 sharpens this focus further, targeting the establishment of lunar infrastructure and the groundwork for Martian expeditions, echoing NASA's Moon to Mars Campaign Strategy.

Integral to SEE's methodology is the emphasis on *systems* (the designs participants develop), *behaviors* (the functions these systems perform), and *interactions* (how the systems communicate), allowing for a flexible, creative approach to simulation that aligns with the Artemis architecture. This framework ensures that SEE serves as a platform for technical skill development and an arena for pioneering real-world applications of simulation technology in space exploration.

Supported by strategic partnerships with entities such as Pitch Technologies and NASA, and open to global participation, SEE cultivates a rich learning environment that emphasizes practical experience, project management, and collaborative innovation. The program culminates in a showcase event, allowing teams to demonstrate their achievements and receive feedback from industry and government partners, underscoring the educational and practical impacts of participation in the SEE program.

This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/WSC63780.2024.10838847, 2024 Winter Simulation Conference (WSC)

3 OVERVIEW OF THE SEE PROJECT

3.1 Essential Requirements for Participation

Participating in the SEE project demands a unique mix of skills and experiences, bringing together individuals from various disciplines such as engineering, computer science, and mathematics to engage in a comprehensive exploration of M&S. Candidates are expected to enter the project with a solid foundation in the core concepts of simulation, enhanced by hands-on experience in software development and 3D modeling, as well as capabilities in project management. The journey through the SEE project is one of significant skill development, where participants evolve into proficient simulation experts capable of addressing complex challenges. Through initial orientation sessions and collaborative labs, recruits are immersed in a learning environment that emphasizes community, innovation, and technical proficiency. This blend of technical training, access to advanced tools, and a culture of shared learning and collaboration equips participants with the expertise needed to contribute effectively to the project and prepare them for future professional roles in the field of modeling and simulation.

SEE Technologies and Simulation Tools: SEE provides participants with access to a range of cutting-edge technologies and simulation tools aimed at enhancing their learning experience and facilitating project development. Through tutorials, seminars, and regular meetings led by industry and academic professionals, participants are introduced to these technologies, ensuring they have the necessary knowledge and skills to effectively contribute to the program. Additionally, while SEE offers a suite of simulation tools, participants are encouraged to explore and utilize tools of their choice based on their preferences and project requirements.

HLA Starter Kit: SEE equips participants with a suite of simulation tools to support their learning and project development. One such resource is the SEE HLA Starter Kit (Falcone et al. 2017; Falcone and Garro 2016), 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. Video tutorials further assist developers in creating federates using the SKF. Released under the open-source Lesser GNU Public License (LGPL), the SKF simplifies federate development by addressing common HLA challenges and supporting interoperability with various HLA RTI implementations. Developed by the Systems Engineering and Integration (SEI) team at the University of Calabria, Italy, in collaboration with NASA Johnson Space Center (JSC), the SEE HLA Starter Kit exemplifies SEE's commitment to providing participants with accessible, cutting-edge simulation tools to enhance their project experiences.

TrickHLA: In addition to the SEE HLA Starter Kit, SEE offers access to 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. With its data-driven approach and straightforward Application Programming Interface (API), TrickHLA facilitates the transition of existing Trick simulations into HLA-distributed simulations, enabling seamless collaboration and interoperability across simulation environments. By leveraging TrickHLA, participants in the SEE program can enhance the scalability and interoperability of their simulations, contributing to the advancement of space exploration and modeling capabilities (NASA 2024a).

Trick Simulation Environment: 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. Trick's primary objective is to furnish a standardized suite of simulation capabilities, freeing users to focus on their domain-specific models without the burden of

This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/WSC63780.2024.10838847, 2024 Winter Simulation Conference (WSC)

managing simulation-specific tasks such as job sequencing, input file handling, or data logging (NASA 2024b).

JSC Engineering Orbital Dynamics (JEOD): 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. JEOD's capabilities extend across various flight regimes, encompassing trajectories from low Earth orbit to lunar operations, interplanetary missions, and deep space exploration. Users can simulate individual spacecraft trajectories and attitude states or integrate JEOD with larger simulation spaces, including spacecraft effectors and guidance, navigation, and control systems. JEOD accommodates simulations involving multiple spacecraft around a single central body or dispersed spacecraft around distinct central bodies, making it a versatile tool for exploring diverse space mission scenarios (NASA 2024c).

Distributed Observer Network (DON): The Distributed Observer Network (DON) represents a data presentation tool developed by NASA to disseminate and showcase simulation outcomes. Harnessing the immersive display capabilities of modern gaming technology, DON transports users into a fully interactive 3D environment populated with graphical models, enabling them to observe the evolution and interactions of these models over time within a designated scenario. Each scenario is powered by data generated from NASA simulation tools and exported in adherence to a standardized data interface specification. Built upon the Unreal Engine, DON combines the realism of gaming technology with the precision of NASA's simulation data, offering a unique and powerful platform for visualizing and comprehending complex simulation outcomes (NASA 2024d).

3.2 SEE Event Scenario

The SEE 2024 Event Scenario marks a significant evolution in the SEE program, emphasizing interactive learning in modeling and simulation with a focus on the Moon, Cislunar Space, and Mars since its inception in 2011. Aligning with NASA's Artemis plan, SEE 2024 shifts towards creating foundational lunar infrastructure and prepping for human Mars exploration, mirroring the objectives of NASA's Moon to Mars Campaign Strategy. This iteration encourages university teams globally to leverage open M&S standards, enabling their exploration concepts to integrate into a cohesive, large-scale simulated space mission. This approach fosters interoperability and real-time collaboration over the Internet and continues the tradition of contributing to the development of space simulation standards, such as SISO SpaceFOM (Crues et al. 2022), demonstrating and extending these capabilities worldwide.

Central to the 2024 scenario are the systems, behaviors, and interactions designed by participants, which are crucial for depicting a sustainable lunar base aimed at supporting a prolonged human presence on the Moon and serving as a gateway for Martian expeditions. The SEE 2024 scenario is set primarily around the Lunar South Pole, envisioning a base that supports the Artemis missions and possibly extends to Mars- related missions. By engaging in surface and subsurface exploration, infrastructure development, and the creation of an industrialized lunar village, SEE participants will tackle the challenges of establishing a lunar population, enhancing economic opportunities, and maximizing scientific returns. This endeavor aligns participant projects with Artemis architecture, offering a vivid, practical experience in the complexities of current and future space exploration.

3.3 Interactive Simulations

SEE has fostered collaboration through the creation of real-time simulations through interaction matrices to encourage dynamic exchanges between groups working on designated federates. Interaction matrices, complemented by baseball cards detail the descriptions of federates, data elements, and potential

This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/WSC63780.2024.10838847, 2024 Winter Simulation Conference (WSC)

interactions, facilitating teams' engagements. Groups are encouraged to delve into these cards to understand not just their contributions but also how they fit into the broader simulation ecosystem. This proactive exploration of federate functionalities and capabilities paves the way for teams to initiate contact, facilitating seamless interactions and integration of their designs. Encoded directly into their projects, these interactions come to life during the program's culminating event, where the federates interact in real time within the simulated environment. This approach enhances the technical and operational cohesion of the simulations that mirrors real-world collaborative efforts, where understanding and integrating diverse systems are crucial for success.

4 CASE STUDIES AND IMPLEMENTATION

As an active participant in SEE 2024, the Brunel team elected to utilize the numerous factors mentioned in the Spaceport Design on the Moon: From First Launch to the Future paper (Anderson et al. 2020) as a reference for our design. The paper outlines the crucial role the First Lunar Spaceport (FLS) would play in the establishment of a lunar civilization, enabling humans to land and launch from the same location twice. Furthermore, it highlights important considerations for the many challenges concerning this undertaking: an ideal base location in areas near the Shackleton Crater at the South Pole, multitasking equipment to reduce the pieces of infrastructure needed to carry out operations, and implications of the lunar environment on traditional safety concerns dealing with operational logistics. It also provides a glimpse into requisite systems such as regolith management across site locations and architectural concerns anent infrastructure intended for human habitation. We aim to incorporate these as per the event scenario, with an emphasis on interoperability, sustainability and effective operations thereby creating a framework for future lunar infrastructure developments. Instructions and troubleshooting are detailed on Brunel's How-to Guide.

Following a complete requirements analysis of the project goals, the design was delineated along the lines of essential on-site facilities. This resulted in four distinct federates: Communications Hub, Storage Depot, Launch Complex and Habitation Quarters. This strategic division placed the Communications Hub at the center of spaceport operations, handling directives among internal and external federates. To optimize design development efforts, the Brunel team was organized into two divisions: Programming and 3D Visualization. The former was tasked with creating a test federate, compatible with the HLA Starter Kit. The latter was in charge of conceptualizing Spaceport locations as 3D models for visualization purposes. The HLA federate codebase and 3D models can be found at (Brunel 2024). The repository is, however, kept private in the interest of safeguarding licensed third-party middleware. Access may be granted upon request.

Concerning project management, the SEE program uses the Businessmap software (formerly Kanbanize), a vital tool for workflow automation, tracking metrics and assisting teams adhere to deadlines. The teams are assigned a workflow board for communicating their project implementation progress throughout the event. The tasks proceed as follows:

1. **Draft Baseball Card:** Teams commence by conceptualizing and documenting the essential functionalities, data elements, and interactions of their federates, akin to creating a baseball card summary for each federate.
2. **Request and Install NASA DON:** Subsequently, teams request and install the DON software from NASA, enabling them to monitor and interact with simulations across the SEE network.
3. **Create Interaction Matrix:** An interaction matrix is then developed, detailing potential interactions between different federates, facilitating a clearer understanding of how they will communicate and collaborate within the simulation environment and other teams.
4. **Request and Install RTI Software:** Teams proceed to request and install the RTI software, crucial for managing the exchange of information among federates in the HLA framework.
5. **Run Simple Federate Locally:** With the RTI software in place, teams execute a simple federate locally to validate the basic functionality and ensure that the federate can operate within the intended simulation parameters.

This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/WSC63780.2024.10838847, 2024 Winter Simulation Conference (WSC)

6. **Connect to SEE VPN:** To participate in the collaborative simulation environment, teams connect to the SEE Virtual Private Network (VPN), securing their access to the shared simulation infrastructure.
7. **Test Federate Using RTI:** Following the successful local execution, federates are tested using the RTI in the collaborative environment, verifying their interoperability and performance within the larger simulation context.
8. **Complete SEE Event Dry Run:** Before the main event, a comprehensive dry run is conducted, allowing teams to simulate their participation, identify any issues, and make necessary adjustments to their federates and strategies.
9. **Integrate Testing with 3D Models:** Integration testing with 3D models follows, where the functionality of the federates is tested in conjunction with the visual models, ensuring a seamless blend of function and form.
10. **Fulfill SEE Event Participation:** Finally, all stages culminate in the actual participation in the SEE event, where teams implement their federates in a live, dynamic simulation, showcasing their collaborative efforts and technical achievements.

This workflow represents crucial steps in task initiation, progress monitoring and completion process, guiding teams from initial conceptualization through to live simulation, ensuring that all aspects of the federates' development and integration are methodically addressed and executed.

4.1 Baseball Card Design

Within the SEE program, the Baseball Card serves as a foundational tool for federates' conceptualization, documentation and presentation. This compact yet comprehensive format is divided into seven key sections, each providing essential insights into the federate design and functionality:

- **Team/Federate Details:** This section introduces the team and the specific federate, offering a snapshot of the federate's name and its intended role within the SEE simulation.
- **Model Description:** Here, a detailed narrative describes the federate's core functions, scope, and the rationale behind its creation, providing a clear understanding of its contribution to the simulation.
- **Simulated Behaviour:** This part elaborates on how the federate behaves within the simulation environment, including its dynamic responses and interactions with other federates.
- **Image:** Accompanying visuals or diagrams offer a visual representation of the federate, enhancing comprehension and engagement with the federate's design and operational context.
- **Tools:** This segment outlines the software, technologies, and methodologies employed in the development and operation of the federate, highlighting the technical underpinnings of the simulation.
- **Data Elements:** It details the specific data inputs and outputs associated with the federate, underscoring the information exchange that facilitates simulation interoperability and realism.
- **Interactions:** Finally, a summary of potential interactions with other federates is provided, illustrating the federate's role within the broader simulation ecosystem and its collaborative dynamics.

The Baseball Card acts as a blueprint for federate development as a communication tool, fostering understanding and collaboration among SEE participants. An example is shown in Figure 1.

4.2 Interaction Matrix Concept

The SEE program employs an Interaction Matrix to map the intricate web of interactions between different federates within the simulation. This matrix, structured in a tabular format, outlines the potential

This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/WSC63780.2024.10838847, 2024 Winter Simulation Conference (WSC)

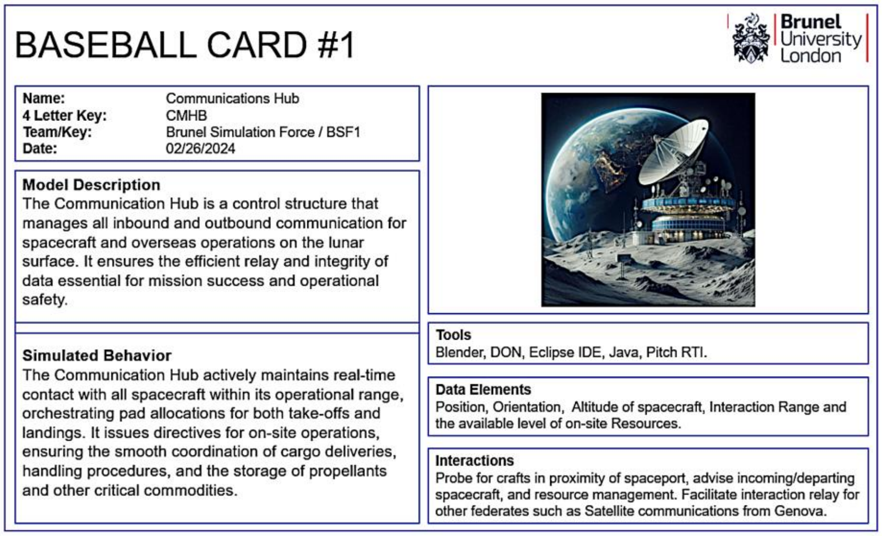

Figure 1: Brunel University Baseball Card for Communication Hub.

communications and dependencies across a diverse range of federates, ensuring a coherent and synchronized simulation landscape. 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.

For instance, Brunel team's Communication Hub serves as a central node within this matrix, facilitating a broad spectrum of interactions with other federates designed by various teams, including those from other universities. This year, our design has garnered considerable interest, notably:

- A university team designing a satellite that orbits the moon has proposed interactions with our Communication Hub, intending to relay data elements such as regular updates, altitude, longitude, and orbital periods. This satellite aims to enhance communication between Rockets and Landers, relying on our hub for data transmission.
- Another team is focused on a Cable Car system designed to transport passengers from the Spaceport to Habitation Quarters. They seek to communicate regular updates through our hub, including schedules and arrival times, ensuring seamless transit operations within the lunar habitat.
- Additional collaborations involve teams working on various vehicles, all of which require integration with our Communication Hub to facilitate essential communications throughout the simulation environment.

Our interaction matrix has evolved to encapsulate the internal dynamics of our federates and incorporate external interactions with federates from other teams, significantly expanding our role within the SEE program.

4.3 Run-Time Infrastructure (RTI)

Federate implementation within the SEE program is significantly enhanced by leveraging the RTI provided by Pitch Technologies, specifically the Pitch pRTI and Pitch Booster. As a cutting-edge simulation infrastructure, Pitch pRTI adheres to the Open International HLA IEEE 1516-2010 standard (IEEE 2010), providing interoperability solutions for simulation and training. Their suite of products, including Pitch pRTI, Pitch Visual OMT, Pitch Commander, and others, support a wide range of applications in civilian and defense sectors, enhancing acquisition and analysis through distributed simulations.

This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/WSC63780.2024.10838847, 2024 Winter Simulation Conference (WSC)

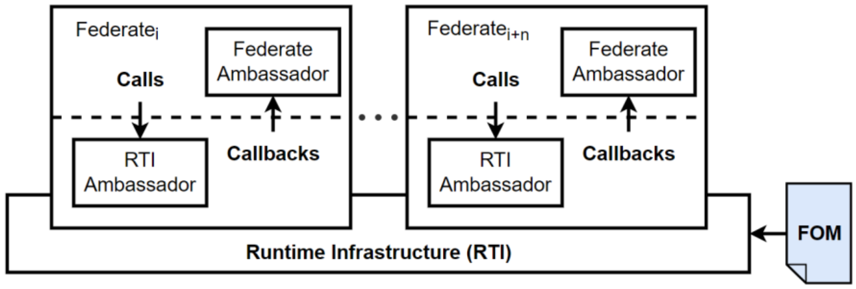

Figure 2: Overview of Runtime Infrastructure.

The RTI acts akin to a service bus, offering essential services such as information exchange, event synchronization, and coordination to all connected Federates. The FOM is the language spoken within a Federation, detailing the data types and interactions that can be exchanged among Federates. It defines the structure of the simulation data and the rules governing their interaction. This structured communication is facilitated through a Publish and Subscribe mechanism, where federates declare their intent to share (publish) data, or their interest to receive (subscribe to) data from others. This mechanism ensures that each federate receives only the information it requires, without being overloaded with irrelevant data, thereby optimizing the simulation's performance.

Central to the communication within HLA are the Federate Ambassador and RTI Ambassador. These are interfaces provided by the RTI that allow federates to interact with the RTI services. The Federate Ambassador handles callbacks from the RTI to the federate, such as notifications about other federates' actions or changes in the simulation state. Conversely, the RTI Ambassador allows federates to issue commands to the RTI, such as sending updates or requesting data. Figure 2 illustrates the core components and functionalities of RTI, highlighting its architecture in information exchange.

For comprehensive guidance on utilizing Pitch RTI, the Pitch Evolved User's Guide, Pitch Technologies (2024) offers detailed instructions and best practices, equipping users with the knowledge to deploy and manage their simulations within the HLA framework effectively.

4.4 Starter Kit Framework

The SEE SKF (Falcone at al. 2017; Falcone and Garro 2016) is a comprehensive solution specifically tailored for space simulations within the HLA distributed environment. It enables various Federates to work together seamlessly through its advanced RTI. The SKF is designed to offer a sophisticated set of functionalities that are essential for managing complex space simulations, divided into two key parts: Space and Core.

The Space part of the SKF is responsible for handling fundamental space elements and the algorithms that manipulate them, which are categorized into four components:

1. **Reference Frames:** Offers data on International Celestial Reference Frames and includes algorithms and functionalities for their management.
2. **Celestial Bodies:** Provides features for representing the position, geometry, and characteristics of space objects, and includes a factory module for creating instances of predefined planets with specific characteristics.
3. **Space Time Manager:** Defines mechanisms to handle epochs and dates across various time standards, such as Terrestrial Time (TT) and Universal Time Coordinate (UTC), and defines common epochs such as Julian Epoch (JE) and j2000 Epoch.
4. **Math Utils:** Supplies miscellaneous mathematical functions that assist in managing both space elements and simulation time.

This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/WSC63780.2024.10838847, 2024 Winter Simulation Conference (WSC)

The SKF's application layer is designed for ease of integration and use. Classes such as '*SEEAbstractFederate*' and '*SEEAbstractAmbassador*' define the behavior of SKF-based Federates, allowing simulation developers to focus on the specific behavior of their space federate without concerning themselves with the low-level details managed by the SKF. The behavior of a federate is bifurcated into proactive and reactive components, with the proactive part dealing with lifecycle management and the reactive part handling RTI callbacks regarding subscribed interactions and objects. With the SKF, space mission federates can be developed with precision and attention to the specificities of space environments, ensuring that federates can communicate effectively and operate cohesively within the larger HLA federation.

RTI Components: In the context of the SEE program, establishing the Pitch Technologies' pRTI locally is crucial for enabling a seamless simulation experience. At the heart of this setup lies the Central RTI Component (CRC), colloquially known as RTIexec. The CRC is instrumental in orchestrating the federation's operations, offering a dual interface (graphical and command-line) that allows for comprehensive monitoring and control of the simulation's execution. It serves as a central node, managing the interplay between the Local RTI Components (LRCs) and ensuring federates are well-informed and synchronized for the duration of the simulation.

RTI Implementation: RTI can be implemented in multiple programming languages, including C++ and Java, or any language that can interface with C++ or be wrapped in Java. The diversity of implementation languages within the same federation is supported, allowing for flexibility and creativity in federate developments.

SISO and Space FOM: The Space Reference Federation Object Model (Space FOM) (Möller et al. 2020), as mandated by the SISO, exemplifies the principles of generality, stability, and supportability within the SEE project. This standard ensures that space simulation components across different platforms can interoperate within a broad and varied user community, fostering an inclusive and extensive environment for educational and research-based simulations.

Time Synchronization: Time synchronization and temporal coordination allow for seamless runtime interaction among federates, representing various components of space missions, such as satellites, rovers, and lunar bases. It ensures that all federates operate within the same timeframe, which is essential for the accuracy and reliability of the distributed simulation, enabling a true-to-life representation of space missions that require precise timing and coordination between various operational elements.

Networking: pRTI employs TCP/IP, the industry-standard protocol, for networking. Configurations can vary from Ethernet-based LANs, WiFi connections, or even a loop-back network interface for simulations running on a single computer. The address of the CRC needs to be specified for the connection, which can be done via a DNS service, direct IP address, or a loop-back address.

4.5 Creating a Federate Project

Creating a federate in Eclipse IDE according to the SKF-based Framework and user guide begins with establishing a clear project structure. Key directories include *src* for source code, lib for libraries such as '*booster1516.jar*' and '*prti.jar*', *fom* for FOM files, *config* for federate/federation configuration, and *log4j* for logging. Libraries are then integrated into the project's *classpath*, central to the federate's communication within the HLA infrastructure. Configuration and XML files are carefully edited to inscribe the federation's details and the federate's unique identity.

The '*LunarRoverFederate*' class, as part of the core package provided by the SKF-based SEE Starter Kit, is a sophisticated embodiment of a federate designed for simulation within a distributed HLA environment, specifically tailored for the SEE program. This class extends the '*SEEAbstractFederate*', leveraging the SKF functionalities to interact seamlessly within the HLA federation.

The configuration and start-up process of this Federate within a SEE federation involves a detailed series of steps:

This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/WSC63780.2024.10838847, 2024 Winter Simulation Conference (WSC)

1. **Framework Configuration:** The SEE SKF is configured using the provided settings, setting the stage for the federate's interaction within the HLA environment.
2. **RTI Connection:** Establish a connection to the RTI with specific local settings designator details, such as the CRC host and port, differing based on the RTI implementation (e.g., MAK or PITCH).
3. **Federation Execution Joining:** The federate joins the SEE Federation execution, marking its entry into the collaborative simulation.
4. **Subject Subscription:** It subscribes to relevant updates, enabling it to receive and process information vital for its operations within the federation.
5. **Late Joiner Check:** Verifies if the federate is a late joiner, a requirement for SEE Teams to ensure synchronization with the federation's timeline and activities.
6. **Synchronization Point Waiting:** Awaits the announcement of the 'initialization completed' synchronization point, indicating readiness for the next phase of simulation activities.
7. **Execution Control Object Subscription:** Subscribes to Execution Control Object Class Attributes and waits for the discovery of these attributes to understand the federation's execution control dynamics.
8. **Execution Configuration Request:** Actively requests updates for the Execution Configuration until it is successfully received and processed.
9. **Mode Transition Request Publishing:** Publishes the Mode Transition Request (MTR) interaction, facilitating transitions between different execution modes within the federation.
10. **Object and Interaction Management:** Engages in the publication and subscription of Object Class Attributes and Interaction Class Parameters, further integrating into the federation's operational flow.
11. **Object Instance Name Reservation:** Initiates the reservation of all federate object instance names, ensuring unique identifiers within the federation.
12. **Federate Object Instances Registration:** Registers federate object instances post name reservation success, solidifying its presence and role within the federation.
13. **HLA Time Management Setup:** Establishes HLA Time Management protocols, including querying the GALT (Generalized Advance Logical Time) and computing the HLTB (High-Level Time-Bound), aligning the federate's temporal operations with the federation.
14. **Synchronization Achievement:** Strives to achieve the 'objects discovered' synchronization point, ensuring the federate is in sync with the federation's state and ready for execution.
15. **Simulation Execution Initiation:** Finally, initiates the simulation execution, marking the start of active participation in the SEE program's simulation activities.

These steps collectively ensure the Federate is correctly configured, connected, and synchronized with the SEE federation, prepared for the complex tasks of simulating lunar exploration activities.

4.6 Running a Federate Locally

In the subsection, we delve into how this federate functions within the confines of a local setup and prepares to integrate into the broader SEE federation, highlighting its connectivity to the DON visualization platform. To facilitate this interaction and ensure that our federate's actions are accurately reflected in the DON, we enlisted the support of two auxiliary projects: SpaceMaster and SpaceReferenceFramePublisher.

Spacemaster Integration: The SpaceMaster project acts as a central hub within the SEE federation, orchestrating the overall simulation flow and ensuring that time synchronization, state management, and interaction coordination are maintained across all federates. By incorporating SpaceMaster into our setup, we enable the Federate to adhere to the federation's execution control protocols, ensuring that it operates in harmony with the collective simulation objectives and timelines.

Frame Publisher Connectivity: To accurately convey the spatial dynamics of our lunar rover within the DON, the '*SpaceReferenceFramePublisher*' project is indispensable. It is responsible for publishing the reference frames – essentially the spatial context in which our rover operates to the DON. These reference

This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/WSC63780.2024.10838847, 2024 Winter Simulation Conference (WSC)

frames provide the necessary spatial orientation, allowing the DON to correctly position and animate our rover's movements within the lunar terrain visualization. Through this project, every action performed by this Federate, from simple maneuvers to complex exploration activities, is translated into visual cues that are understandable and meaningful within the visualized space environment.

Visualization Integration Flow: By intricately weaving the functionalities of the SpaceMaster and SpaceReferenceFramePublisher into the federate's local execution, we establish a robust framework as follows:

1. **Initialization:** Upon starting the federate locally, initialization routines ensure that both the Space-Master and SpaceReferenceFramePublisher projects are engaged. This sets up a coherent operational environment where time management and spatial reference frames are in sync with the federation's requirements.
2. **Interaction Handling:** As the federate executes its operational logic, interactions, representing various events and states, are generated. These interactions, especially those related to spatial changes, are directed towards the SpaceReferenceFramePublisher.
3. **Visualization Mapping:** The SpaceReferenceFramePublisher processes these interactions, mapping them onto the appropriate reference frames. This mapping is critical for ensuring that the visualization platform accurately represents the rover's location, orientation, and movements.
4. **DON Integration:** With the spatial context established and interactions processed, the visualization data is then relayed to the DON. This ensures that each frame displayed on the DON is a true representation of the federate's current state and activities.
5. **Continuous Synchronization:** Throughout the simulation, continuous synchronization with the SpaceMaster ensures that the federate's operation remains aligned with the federation's execution mode. Simultaneously, ongoing communication with the SpaceReferenceFramePublisher guarantees that the visual representation is consistently updated in response to the federate's actions.

This robust framework supports local testing and development to ensure readiness for integration into the SEE Federation and DON visualisation platform.

4.7 Running a Federate Remotely

For running a federate remotely in the SEE Federation, we transition from a local setup to integrating our federate within the broader context of the SEE Federation's distributed simulation environment. This integration is an important step towards participating in both dry run testing and the final event, ensuring our federate's seamless operation within simulating lunar exploration missions.

Connecting to SEE Federation: The process of moving from a local testing environment to the SEE Federation involves a critical step: connecting to the SEE-provided VPN. This VPN establishes a secure and dedicated network channel, enabling federates from various locations to communicate and synchronize as if they were operating within a single, unified network environment.

The necessary configuration adjustments for Remote Operation are as follows:

1. **VPN Integration:** Initially, ensure that the VPN client is correctly configured and connected to the SEE Federation's provided VPN service. This connection is essential for accessing the distributed simulation network, where the federates, including ours, interact.
2. **IP Address Update:** With the VPN connection established, the next crucial step involves adjusting the federate's configuration settings to align with the remote operation requirements. Specifically, the local settings designator within the federate configuration, which previously pointed to local or intranet CRC addresses, needs to be updated. Replace these addresses with the IP address or hostname of the CRC provided for the SEE Federation event. This adjustment ensures that our

This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/WSC63780.2024.10838847, 2024 Winter Simulation Conference (WSC)

federate can locate and communicate with the federation's central coordination component over the VPN.

3. **Federation Execution Joining:** Once the federate configuration aligns with the remote CRC settings, the federate is ready to join the federation execution. This step is akin to the local setup but now occurs within the broader scope of the SEE Federation's distributed network, facilitated by the VPN connection.
4. **Dry Run and Final Event Participation:** Successfully connecting to the federation allows the federate to participate in scheduled dry run tests. These tests are crucial for identifying and resolving potential issues before the final event. Subsequently, the federate is also prepared for active participation in the final event, contributing its simulation capabilities to the collective federation efforts.

By carefully following these steps and ensuring a secure connection to the SEE federation's VPN, adjusting the federate's configuration to match the remote CRC settings, and actively engaging in dry run tests, federates can smoothly transition from local development to full-scale remote operation within the SEE federation. This approach guarantees that our federate, now thoroughly tested locally, is ready to contribute its functionality to the ambitious goals of the SEE program during the dry run and culminating in the final event.

4.8 Interoperability Tests

During the interoperability testing phase, all SEE federates successfully connect and interact within the distributed simulation environment. This critical phase validated the seamless integration and communication between diverse federates, each representing unique aspects of the lunar exploration mission. The tests focused on ensuring that data exchange, synchronization of actions, and collaborative mission objectives were met without any hitches, demonstrating the robustness of the federate's design and its compatibility within the SEE federation. Achieving this milestone underscored the collective capability to simulate complex space exploration scenarios, paving the way for a realistic and cohesive simulation experience.

4.9 Resource Utilization

While specific tests and observations on resource utilization within the SEE project have not yet been conducted for our federate, understanding and managing the computational demands it places on the system, such as CPU usage and memory consumption, are essential for maintaining simulation efficiency and stability. Future evaluations will focus on these aspects, particularly during various operational phases of the federate within the federation, to identify potential optimization. Whether it's through refining data exchange protocols to reduce CPU load or addressing memory inefficiencies to curb consumption, these assessments aim to enhance the federate's scalability and performance. By tracking resource usage across different simulation scenarios and analyzing their impact, the goal is to optimize the federate's design for resource efficiency. This approach ensures functional alignment with the SEE program's objectives that contribute to the distributed simulation's success by minimizing resource strain, ultimately leading to a more seamless and stable simulation environment.

4.10 Latency and Throughput

In larger simulations, such as those conducted within the SEE project, addressing the challenges of latency and throughput becomes critical to maintaining the fidelity and responsiveness of the distributed environment. One effective strategy to overcome these challenges involves leveraging multiprocessing utilities such as the Message Passing Interface (MPI) (Clarke et al. 1996) and process parallelization techniques. These tools facilitate efficient data distribution and task execution across multiple computing nodes, significantly enhancing the simulation's ability to handle complex computations and large datasets

This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/WSC63780.2024.10838847, 2024 Winter Simulation Conference (WSC)

with minimal delays. By partitioning the simulation workload and enabling concurrent processing, MPI and parallelization improve the overall throughput and reduce latency, ensuring timely data exchange and synchronization among federates. This approach is particularly beneficial in scenarios where the federate's interactions and data exchanges are intensive, requiring a robust framework to support high-performance computing needs. Implementing such technologies paves the way for more scalable and efficient simulations, allowing the SEE program to achieve its ambitious objectives without compromising on speed or accuracy.

5 ASSESSMENT AND EVALUATION

Throughout our journey with the SEE program, we encountered and navigated through a myriad of challenges, particularly in relation to the SKF. The necessity for a modernized framework became apparent as we sought to enhance usability and integration with evolving technologies such as Unreal Engine. We identified a transition towards a Maven-based project structure as a crucial step forward, facilitating streamlined dependency management and significantly simplifying the project setup process. This approach, coupled with the adoption of the new version of DON, promised a more immersive simulation experience but also highlighted the complexities of configuration and the risk of misconfiguration due to the intricate web of dependencies. By envisioning and advocating for a simplified, automated framework, we aim to alleviate these challenges, enabling participants to focus on the educational and exploratory objectives of the simulation with greater ease and efficiency.

The project's evolution brought forth valuable lessons and best practices, essential for guiding future endeavors in distributed simulation environments. Key among these was the importance of early and ongoing interoperability and performance testing, ensuring compliance with federation standards while optimizing resource efficiency. We also underscored the value of fostering an environment of collaboration and knowledge sharing, enhancing the learning experience and cohesion among participants. Furthermore, the critical role of comprehensive documentation and user guides became evident, significantly reducing setup errors and operational challenges. These insights, garnered from our experiences within the SEE program, form a cornerstone of our recommendations for future projects. By embracing these practices and continually engaging in reflective improvement, we can enhance the quality, reliability, and impact of distributed simulation projects, contributing to the advancement of simulation technology and its applications in education and research.

6 CONCLUSIONS

Reflecting on our journey with the SEE program, the Brunel team achieved significant technical milestones that have broad implications for the future of distributed simulations in educational contexts. Key among these was the successful integration of complex systems, overcoming specific technical challenges, and the realization that an updated platform could greatly streamline the initial setup process for participants. This advancement provides them with more time to experiment with new concepts, fostering innovation and creativity within the simulation environment.

Our approach to enhancing collaboration and interoperability within the SEE project focused on creating a supportive and engaging community. By organizing sessions that were both interesting and friendly, we facilitated an environment where participants felt comfortable sharing ideas and working collaboratively. The availability of resources to all, coupled with quick and efficient issue resolution, underscored the value of collaborative development. The creation of a lab environment further encouraged teamwork and mutual learning among participants, illustrating the potential of collaborative efforts in advancing simulation technology.

On the performance and optimization front, we recognized the critical importance of resource management, latency reduction, and scalability. Our vision for the future includes developing a simplified and automated framework to reduce the complexity and enhance the enjoyment of participating in the SEE program. Looking ahead, we aim to establish our own SEE Virtual Organization, equipped with accessible servers that allow participants to conduct their testing and design work more effectively.

We plan to share our experiences and insights at the Winter Simulation Conference, aiming to reach a broader audience and inspire more potential participants. Our journey underscores the impactful convergence of technical innovation, collaborative development, and educational enhancement, paving the way for the next generation of space exploration simulations. Through these efforts, we contribute to the growing body of knowledge in simulation technology and its applications in education, highlighting the SEE program's role as a catalyst for change and advancement in the field.

SEE welcomes new teams. Please contact the authors if you are interested.

ACKNOWLEDGMENTS

We would like to thank the Modelling & Simulation Group and the Department of Computer Science at Brunel University London for their continuous support throughout the project as well as the preceding Brunel teams for sharing their knowledge and experiences. We also thank the entire SEE team and Pitch Technologies for providing the technology, leadership and technical support.

REFERENCES

Anderson, A., M. Fehlinger, K. Harris, R. Rogers, and D. Valletta. 2020. "Spaceport Design on the Moon: From First Launch to the Future". In *Proceedings of the Accelerating Space Commerce, Exploration, and New Discovery (ASCEND) Conference*, November 16th-18th, Las Vegas, NV, 4189 https://doi.org/10.2514/6.2020-4189.

Brunel. 2024. SEE SmackDown 2024 GitHub Repository. https://github.com/mzrghorbani/SEE-2024, accessed 24th April 2024.

Clarke, L., I. Glendinning, and R. Hempel. 1994. "The MPI Message Passing Interface Standard". In *Programming Environments for Massively Parallel Distributed Systems,* edited by Decker, K.M. and R.M. Rehmann, 213-218. Monte Verità. Birkhäuser, Basel https://doi.org/10.1007/978-3-0348-8534-8_21.

Crues, E.Z., D.E. Dexter, B. Möller, A. Falcone, and A. Garro. 2022. "SpaceFOM: An Interoperability Standard for Space Systems Simulations". In *2022 IEEE Aerospace Conference (AERO)*, March 5th-12th, Big Sky, MT, USA, 1-10 https://doi.org/10.1109/AERO53065.2022.9843262.

Dahmann, J.S., R.M. Fujimoto, and R.M. Weatherly. 1997. "The Department of Defense High Level Architecture". In *1997 Winter Simulation Conference (WSC)*, 142-149 https://doi.org/10.1145/268437.268465.

Falcone, A., A. Garro, S.J.E. Taylor, A. Anagnostou, N.R. Chaudhry, and O. Salah. 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.

Falcone, A. and A. Garro. 2016. "The SEE HLA Starter Kit: Enabling the Rapid Prototyping of HLA-based Simulations for Space Exploration". In *Proceedings of the Modeling and Simulation of Complexity in Intelligent, Adaptive and Autonomous Systems 2016 (MSCIAAS 2016) and Space Simulation for Planetary Space Exploration (SPACE 2016)*, April 3rd-6th, Pasadena, CA, 1-8 https://doi.org/10.22360/SpringSim.2016.MSCIAASSPACE.001.

IEEE. 2010. "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-- Federate Interface Specification". In *IEEE Std 1516.1-2010 (Revision of IEEE Std 1516.1-2000)*, 1-378 https://doi.org/10.1109/IEEESTD.2010.5557728.

IEEE. 2010a. "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-- Framework and Rules – Redline", In *IEEE Std 1516-2010 (Revision of IEEE Std 1516-2000) - Redline*, 1-38 https://doi.org/10.1109/IEEESTD.2010.5953411.

IEEE. 2010b. "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-- Federate Interface Specification – Redline", In *IEEE Std 1516.1-2010 (Revision of IEEE Std 1516.1-2000) - Redline*, 1-378 https://doi.org/10.1109/IEEESTD.2010.5954120.

IEEE. 2010c. "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-- Object Model Template (OMT) Specification – Redline", In *IEEE Std 1516.2-2010 (Revision of IEEE Std 1516.2-2000) - Redline*, 1-112 https://doi.org/10.1109/IEEESTD.2010.5953408.

IEEE. 2010d. "IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP)", In *IEEE Std 1730-2010 (Revision of IEEE Std 1516.3-2003)*, 1-79 https://doi.org/10.1109/IEEESTD.2011.5706287.

Möller, B., A. Rydell, E.Z. Crues, D. Dexter, A. Falcone, and A. Garro. 2020. "SISO Space Reference FOM-Tools and Testing". In *SISO 2020 Simulation Innovation Workshop*, 10-14 February 10th-14th, Orlando, FL, JSC-E-DAA-TN76473.

NASA. 2024a. *TrickHLA*. https://github.com/nasa/TrickHLA, accessed 24th April 2024.

NASA. 2024b. *Trick Simulation Environment*. https://github.com/nasa/trick, accessed 24th April 2024.

This article has been accepted for publication in a future proceedings of this conference, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/WSC63780.2024.10838847, 2024 Winter Simulation Conference (WSC)

NASA. 2024c. *JSC Engineering Orbital Dynamics (JEOD)*. https://github.com/nasa/jeod, accessed 24th April 2024.

NASA. 2024d. *The Distributed Observer Network*. https://software.nasa.gov/software/KSC-13775, accessed 24th April 2024.

Pitch Technologies. 2024. *Pitch pRTI™ User's Guide*. https://www2.pitch.se/pRTI1516e/Releases/v5.3.2.1/MxdozAZrF/docs/Users%20Guide/prti_evolved_users_guide.pdf, accessed 24th April 2024.

SEE. 2024. *SEE 2024 Event Scenario*. https://simulationexplorationexperience.org/events/, accessed 29th April 2024.

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.

AUTHOR BIOGRAPHIES

MAZIAR GHORBANI is an Associate Lecturer in Computer Science at Brunel University London. His research spans mathematics, electronics engineering, computer science, and experimental physics. He has contributed to various European and international projects, including the High-Luminosity Upgrade of the Large Hadron Collider, Horizon, STAMINA, SEAVEA, and ITFLOWS, focusing on large-scale simulations and predictive models. He has also been involved in the Simulation Exploration Experience (SEE) program as a team co-lead, applying his simulation technologies expertise. He has authored and coauthored several publications in high-energy physics, distributed high-performance computing, agent-based simulations, and broader interdisciplinary subjects. His email address is maziar.ghorbani@brunel.ac.uk.

ANASTASIA ANAGNOSTOU is a Senior Lecturer in the Department of Computer Science at Brunel University London and the co-lead of the Modelling & Simulation Group. Her main research interests lie in the areas of Advanced Computing Infrastructures for M&S, Open Science and M&S for Global Challenges primarily in Industry and the Healthcare sector. At the Winter Simulation Conference (WSC), she served as Chair of the PhD Colloquium Committee in 2018 and was Track Co-Chair for the Healthcare, Introductory Tutorials and Hybrid Simulation Tracks. She is also Program Chair for WSC 2025. She is ACM SIGSIM Vice Chair and Associate Editor of the Journal of Simulation. Her email address is anastasia.anagnostou@brunel.ac.uk.

NURA TIJJANI ABUBAKAR is an Associate Lecturer in the Department of Computer Science at Brunel University London. His research interests lie in the domain of inter-cloud distributed simulation. Before joining Brunel, he was a lecturer and Head of the Department of Innovation Technology at Jigawa State Institute of Information Technology (Informatics) Kazaure, Nigeria. He holds a Bachelor of Science in Computing & Software Engineering from Informatics Singapore (2008) and a Master of Science in Software Engineering from Oxford Brookes University (2011). He also holds a PhD from Brunel University London (2022). Nura has been team co-lead in Simulation Exploration Experience (SEE) for several years. He is a peer reviewer for the Journal of Simulation (JOS) and Winter Simulation Conference (WSC). He was involved in the STAMINA research project and has worked on the European Cloud Orchestration at the Level of Application (COLA) HORIZON2020 and Science Gateways Africa projects. His ORCID ID is https://orcid.org/0000-0002-4216-3057. His email address is Nura.Abubakar2@brunel.ac.uk.

HRIDYANSHU AATREYA is a student and Associate Teaching Assistant at Brunel University London. His areas of interest in research include operating systems, computer networks, simulation modeling and distributed systems. In SEE 2024, Hridyanshu served as the Brunel team's project architect. His email address is 2200096@brunel.ac.uk.

DAMON CURRY has 30 years of experience in the simulation industry specializing in distributed training systems, 3D visualization, 3D terrain, and simulated radio. He is President of Simulation Exploration Experience Inc, the non-profit corporate entity that sponsors the Simulation Exploration Experience (SEE) activities for university level STEM education for distributed simulation. Damon is Pitch Technologies' manager for business development in North America. In addition, he serves as President of the Simulation Interoperability Standards Organization (SISO). Damon is inventor of two realtime image processing techniques for virtual reality with two patents awarded. Prior to working in the simulation industry, he served 16 years with the US Air Force, including software engineering on cruise missiles, avionics engineering for the F-16, and radio maintenance for HF, VHF, UHF and other radio systems. Damon is a graduate of The Ohio State University with a Bachelor of Science in Electrical Engineering. His email address is damon.curry@simexplore.org.

SIMON J E TAYLOR is a Full Professor at Brunel University London. He is also Co-Director of the Modelling & Simulation Group at the same University. His research has impacted over 3 million students and 300 universities in Africa, over 30 SMEs and several large enterprises including the Ford Motor Company and Sellafield. He co-founded the Journal of Simulation and the UK Simulation Workshop series and continues to be an editor at the Journal. He is a former chair and a member of ACM SIGSIM's Steering Committee. He is a member of the Computer Simulation Archive Advisory Committee and Executive Chair of the Simulation Exploration Experience. He continues to be interested in new technologies and simulation, international development and Open Science. His email address is simon.taylor@brunel.ac.uk and his website is https://www.brunel.ac.uk/people/simon-taylor.