Evolving the SpaceFOM: Lessons Learned and Future Development

Björn Möller
Pitch Technologies AB
Repslaregatan 25
582 22 Linköping, Sweden
bjorn.moller@pitch.se

Edwin Z. Crues, Ph.D.
NASA Johnson Space Center
2101 NASA Parkway
Houston, TX 77058
edwin.z.crues@nasa.gov

Alfredo Garro
Depart. of Informatics, Modeling,
Electronics and systems Engineering (DIMES)
University of Calabria
via P. Bucci 41C
87036 Rende, Italy
alfredo.garro@dimes.unical.it

Alberto Falcone
Institute for High Performance Computing and Networking
Italian National Research Council
via P. Bucci
87036 Rende, Italy
alberto.falcone@icar.cnr.it

Daniel E. Dexter
NASA Johnson Space Center
2101 NASA Parkway
Houston, TX 77058
daniel.e.dexter@nasa.gov

Keywords: HLA, SpaceFOM, Lessons Learned

ABSTRACT: *For over fifty years, simulation has been a cornerstone technology for space missions. In many cases, one single simulator, developed by one team, is not enough to meet the simulation objectives. Astronaut crew training, docking, vehicle testing and complex mission planning all require several simulations to be connected and to interoperate, in order to achieve the goal of the simulation. To facilitate the development of interoperable simulations, an open standard, the SISO SpaceFOM was developed in collaboration between government, industry and academia. The standard was released in 2020.*

The initial release of SpaceFOM focuses on the core interoperability topics for space simulation: (i) specifying spatial data using several, well defined reference frames, (ii) managing time (including real-time, hard real-time, and as-fast-as-possible execution), (iii) composition of systems with subsystems such as space vehicles, and (iv) execution control, such as initializing, running, pausing, and terminating simulations.

Since its introduction, SpaceFOM has been successfully applied in high-profile projects such as NASA's Artemis program and its Gateway initiative, as well as in European efforts like the Harwell Laboratories' validation, verification and accreditation (VV&A) activities. To support the standard, a dedicated product support group has been established. It is actively engaged in promoting the standard, gathering user feedback, and planning future updates.

A number of requirements for the next version of SpaceFOM have been identified and proposed solutions are already under development. One key area is the object classes used for data exchange, where a richer set of classes have been proposed, based on practical use cases. For time management, the use of variable time resolutions and variable time steps is also proposed. For execution control, save and restore of simulations is *considered as well as improved robustness, to handle failing components. A number of clarifications are also under consideration.*

This paper provides both a forward-looking view of the technical developments underway and a retrospective on the standard's evolution. The authors emphasize that prioritizing core challenges and delivering a practical first version were crucial steps. While not every idea was implemented in the initial release, the timely delivery of a functional standard enabled real-world applications, the growth of a user community, and the collection of valuable insights to shape future versions. The authors look forward to working with the next version of SpaceFOM together with the growing user community.

1 Introduction

Spaceflight is inherently challenging, hazardous, and costly, with human spaceflight being even more demanding. To mitigate these challenges, the space industry extensively employs computer simulation across all stages of mission development, including concept design, engineering, training, and operations. Simulation enables the evaluation of mission concepts, optimization of system designs, and safe training for rare or dangerous scenarios. Distributed simulation further enhances capabilities by integrating models across organizations, facilitating collaboration and testing. To streamline integration and promote interoperability, open standards play a vital role by reducing development effort, fostering reuse of tools, and providing a neutral framework to share best practices within the industry.

This paper seeks to summarize practical experiences from the first version of the SpaceFOM [1] standard for distributed simulation in the space domain. This standard is based on a domain independent standard for distributed simulation, the High Level Architecture (HLA) [2, 3, 4]. It describes lessons learned, and a number of proposed technical improvements for the next version. The paper also provides a retrospective of the development and lessons learned.

2 Development of SpaceFOM

As described above, space systems simulation covers a broad spectrum of application domains and can benefit from distributed simulation technologies. One currently used standard for space systems distributed simulation it the Simulation Interoperability Standards Organization (SISO) Space Reference Federation Object Model (SpaceFOM). This SpaceFOM standard did not emerge fully formed in January 2020. It formed through an evolutionary development process over many years. Here, we will discuss two principal driving forces in SpaceFOM development: cooperative space exploration systems development between the United States (US) National Aeronautics and Space Administration (NASA), the European Space Agency (ESA), and the Japanese Aerospace Exploration Agency (JAXA); and a student distributed simulation coopetition called the Simulation Exploration Experience (SEE).

2.1 Early HLA efforts

Modeling and simulation (M&S) is a critical technology for NASA and has been since its inception in 1958. This is also true for ESA and other national space agencies around the world. As the world's space exploration capabilities advanced and as these space agencies began to collaborate more closely, distributed simulation technologies provided a solution for cooperative M&S while still protecting proprietary technologies. The Distributed Interactive Simulation (DIS) standard [5] was used in one of NASA's early distributed simulation development work done in the Software Technology Branch at NASA's Johnson Space Center (JSC) in the early 1990's. While DIS work continues, NASA and ESA have been exploring a variety of alternative simulation interoperability technologies including the High Level Architecture (HLA) [2].

In the late 1990's, the first elements of the International Space Station (ISS) were place into low Earth orbit by NASA and an international coalition of space agencies including the ESA, the Russian Space Agency (RSA), JAXA. RSA, ESA, and NASA developed an early HLA-based distributed simulation to model docking the ESA Automated Transfer Vehicle (ATV) with the Russian Segment of the ISS [6]. This work laid the groundwork for work between NASA and JAXA to support the development of a flight controller training simulation for the

JAXA H-II Transfer Vehicle (HTV) in the early 2000's [7]. This was a successful development project which supported many successful deployments of the HTV logistic supply vehicles to the ISS. It is also the foundation for NASA's current ISS visiting vehicle simulation capabilities.

In the mid to late 2000's, NASA started work on the Constellation program. The Constellation program was intended to return humans to the Moon and then go on to explore Mars. Like the ISS, Constellation was an international effort. The HLA-based distributed simulation work developed with the ISS program was adopted by the Constellation program and expanded to include design, analysis, development, integration, test, and verification. This resulted in a number of development project like the Distributed Space Exploration Simulation (DSES) and the Integrated Mission Simulation (IMSim) [8, 9, 10, 11]. Ultimately, the Constellation program was ended in 2010. Fortunately, the distributed simulation technology development was continued by the NASA Exploration Systems Simulations (NExSyS) team in the Simulation and Graphics Branch (ER7) in the Software, Robotics, and Simulation (ER) division at JSC. This work continues today with the Artemis program as described in Section 3.

As these programs and project progressed, many lessons were learned, techniques developed, and software packages created [12, 13]. While there is a wealth of experience in the application of HLA-based distributed simulations in the form of standardized Federation Object Models (FOMs), none seemed to fit the particular and often uncompromising requirements of the space community. The SISO community began to recognize the need for a specialized FOM to support the space community in the early to mid 2010's.

2.2 Simulation Exploration Experience (SEE)

Even with the termination of NASA's Constellation program, there remained an avid interest in distributed simulation technologies for space exploration. Interest remained at NASA, ESA, and other space agencies but there was an enthusiastic excitement in the student bodies of technical university programs with a corresponding untapped wealth of innovation and imagination. SISO recognized this potential and started developing a cooperative distributed simulation competition (coopetition) that was originally called the Student Distributed Simulation "Smackdown" [14]. It was later named the Simulation Exploration Experience (SEE) [15].

SEE joins students, industry, professional associations, and faculty in collaborative college-level modeling and simulation learning. The SEE leadership formulates a general space related scenario for each year's event. Since its inception, this has typically been related to simulating aspects of a lunar exploration scenario. Although, some Mars elements have been explored. The students form university-centric teams that take on one or more aspects of the proposed scenario. The teams build simulations of their proposed aspects. These simulations participate as HLA-based federates in a large scale distributed simulation execution as the culminating event of the competition.

The initial SEE competition was held in the Spring of 2011 at the SISO/SCS 2011 Fall Simulation Innovation Workshop in Boston, Massachusetts, USA. SEE has held a competition every year since then, including through the COVID years where all the events were held virtually. Fortunately, distributed simulation is ideally suited to virtual meeting spaces and further demonstrated the strengths of the technology. This year's event will be the 15th straight year of SEE competition.

One of the early concerns from M&S professionals in SISO was that distributed simulation was too involved and too nuanced for undergraduate university students to be able to understand in the short time frame of the SEE competition cycles much less be able to build working federates of cooperative space systems. While understanding these concerns and the associated challenges, the SEE organizers felt that by limiting the scope of the scenarios and providing support from experts in the field, well developed tool sets, and support from supervising faculty, the student teams could succeed. Among other things, the SEE organizers developed space focused FOM that provided the basics for space systems modeling. They also developed and provided a simplified federation execution control strategy that the students could easily understand and implement. Over the years, a number of support tools and training materials have been developed and provided to the student teams. Much of this went into what became the SpaceFOM. In essence, SEE was the development and testing ground for the SpaceFOM (see Section 3.3).

2.3 Towards a first standard

The development of the SpaceFOM [1, 16] was initiated by members of previous efforts, with representation from government, industry and academia. SISO was selected as the standardization body. The development followed the SISO Balloted Product Development and Support Process (BPDSP) [17]. A SISO Product Nomination, describing the purpose and scope of the proposed standard was submitted and approved. The development included the analysis of previous efforts, drafting, testing, implementation and finally balloting. The standard was finally published in 2020.

2.4 Scope of SpaceFOM 1.0

The SpaceFOM standard is designed to address the unique requirements of distributed simulation for space missions. By providing a comprehensive framework, it facilitates interoperability, consistency, and realism in simulations across diverse federated systems. The key features of SpaceFOM are as follows:

- *Role and Responsibility Management.* SpaceFOM defines mechanisms for assigning and managing the roles and responsibilities of Federates within a Federation execution. This ensures clear collaboration and delineation of tasks among simulation components, preventing conflicts and improving the overall coherence of the simulation.

- *Data Type Management.* To support consistent data exchange, SpaceFOM manages common data types critical for the space domain. These data types provide a standardized foundation for communication and data interpretation, ensuring uniformity across heterogeneous systems.

- *Time Management.* Time management is a cornerstone of SpaceFOM, enabling the coordination of simulation timelines and time scales. The standard supports precise synchronization of events and ensures time homogeneity through specific time-stepped approaches. This is particularly vital for missions where accurate timing is essential, such as orbital maneuvers or docking procedures.

- *Flexible Reference Frame.* SpaceFOM incorporates a robust system for handling flexible reference frames. It supports accurate positioning and tracking of arbitrary celestial bodies and spacecraft, along with a standardized naming convention for operational reference frames. This ensures consistency in defining and utilizing space frames, a critical aspect of simulations involving relative positioning.

- *Support for Physical Entities.* SpaceFOM provides foundational support for simulating physical entities such as spacecraft, satellites, and astronauts. It also models physical interfaces (e.g., docking ports, sensors) to enhance simulation realism and operational functionality, enabling detailed interactions between simulated systems.

- *Synchronized Execution Control.* The standard ensures coordinated execution through a synchronized control strategy and framework. This allows simulation activities to proceed in a unified and deterministic manner, essential for maintaining fidelity in distributed simulations.

- *FOM Management.* SpaceFOM delivers a suite of Federation Object Model (FOM) modules necessary for SpaceFOM-compliant Federation executions. These modules provide the essential components for building interoperable simulations while offering flexibility for extensions and customizations.

SpaceFOM is defined through a combination of human-readable documentation, templates, and computer-parsable FOM module files. These components work together to ensure clarity, usability, and extensibility.

The text-based document serves as the descriptive backbone of the standard, detailing: Explanation of concepts, rules, and relationships critical for interoperability (Semantic content); Definition of responsibilities for coordinating Federation execution and handling critical data during initialization and runtime (Management roles); Codification of actions and relationships that enable seamless interaction among Federates (Rules and behaviors).

Two templates are provided to standardize agreements and compliance within a SpaceFOM-based Federation:

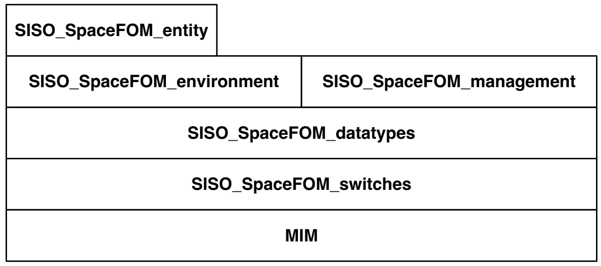


Figure 1: The SpaceFOM FOM modules and their relations.

- Federation Execution-Specific Federation Agreement (FESFA). It specifies the foundational aspects of a particular Federation execution, ensuring alignment among Federates.
- Federation Compliance Document (FCD). It documents each Federate's level of SpaceFOM compliance, aiding in integration and validation.

SpaceFOM includes five XML-based modules, compliant with IEEE 1516-2010, organized by function to simplify development and extension (see Figure 1):

1. *SISO_SpaceFOM_switches*. It defines configuration settings and operational switches.
2. *SISO_SpaceFOM_datatypes*. It provides standardized data types for consistent communication.
3. *SISO_SpaceFOM_management*. It facilitates coordination and control of Federation execution.
4. *SISO_SpaceFOM_environment*. It models environmental parameters such as celestial bodies and orbital mechanics.
5. *SISO_SpaceFOM_entity*. It represents physical entities, their properties, and interactions.

These modules are built upon the *Management and Initialization Module (MIM)*, which includes the *Object Model Template (OMT)* tables. The MIM defines the Management Object Model (MOM), a critical component for controlling and monitoring Federation execution.

3 Practical Experiences from SpaceFOM 1.0

After its release, the SpaceFOM has been adopted by several programs in different organizations. This section describes some of the projects and summarizes practical experiences.

3.1 NASA: Returning to the Moon and on to Mars

As discussed previously, the architecture and content of the SpaceFOM was influenced by the experiences and needs of NASA's human spaceflight programs and projects. Now, after more than a 50 year absence, NASA is making plans to return humans to the surface of the Moon with the Artemis program. The Artemis program is part of a long term objective to carrying human exploration beyond the Moon and on to Mars in what is referred to as the Moon-to-Mars Architecture [18]. Artemis is a multi-agency, multi-corporation, multi-national program with a diverse set of key elements provided by a variety of partners. In an effort to promote the cooperative and peaceful exploration of space within the Artemis Program, NASA established the Artemis Accords in 2020, initially including 7 other signatory nations [19]. As of the writing of this paper, the Artemis signatories number 52. The Artemis Accords provide a common set of principles to enhance the governance of the civil exploration and use of outer space.

The Artemis program poses a challenge to traditional systems engineering and integration (SE&I) approaches. The multi-lateral nature of the program along with the number constituent elements and possible mission scenarios makes a traditional modeling and simulation approach based on monolithic domain focused simulations insufficient to the needs of the program. Fortunately, distributed simulation approaches, like those based on HLA and SpaceFOM, offer a possibility to create large scale integrated simulations of the proposed system of space systems. The NExSyS team has been developing modeling and simulation technologies, digital spacecraft systems models, integrated vehicle simulations, and large scale distributed simulations to address the needs of the Artemis program. The NExSyS team in developing the Artemis Distributed Simulation (ADS) [20, 21].

The general purpose of ADS is to create an integration framework for simulation interoperability based on openly available international standards that supports the development and analysis of NASA's Artemis Program architecture. This effort includes mission analysis, testing, integration, verification, training, and mission support [22]. ADS enables end-to-end Artemis mission analysis using authoritative partner element simulations while providing access to NASA developed element simulations and NASA test, integration, and training facilities.

3.2 Experiences in Europe

Several SpaceFOM projects have been initiated by the European Space Agency (ESA). One example is the HRAF3 (see [23, 24]) project at the Harwell Robotic and Autonomy Facility in the UK. The purpose of the federation is to support the development and testing of complex autonomous systems for Solar system exploration. A flexible simulation environment has been developed where models and real hardware are combined, compared and tested. One scenario that is simulated is the autonomous rendezvous and capture (RvC) of a passive object in Mars orbit in support of a sample return mission. Another scenario is precision landing of a spacecraft on a low gravity Near Earth Object. In the HRAF3 project, several subsystems, like sensors, mechanical systems and navigation systems were modeled using either simulations or by interfacing the real hardware. The SpaceFOM object class hierarchy was extended with subclasses to be able to represent the particular properties of these subsystems (see Section 4.1.1).

Another European project is the UK Space Agency project "REALM: Virtual Reality Aided Spacecraft Refueling for In-Orbit Servicing and Manufacturing (IOSM) Verification & Validation" [25] including VR environments based on the Godot game engine. ESA also sponsors a project for SpaceFOM integration with the SIMULUS-NG simulation platform [26] which would facilitate seamless communication and data exchange among various spacecraft simulation models running in the two environments. Over the years, ESA has used SIMULUS environment for simulating spacecraft behavior and ground systems interaction, enabling engineers to test procedures, train operators, and validate mission plans before launch. SpaceFOM opens up opportunities for ESA projects that increasingly require simulations of constellations and multi-element missions, often involving elements from multiple space agencies.

3.3 Experiences in Academia

Since 2011, SISO, in collaboration with NASA and various industrial and academic organizations, has hosted the annual SEE event (see Section 2.2). This international initiative provides undergraduate and postgraduate students with practical, hands-on experience in Modeling and Simulation (M&S) techniques, with a particular focus on Distributed Simulation (DS) systems that adhere to the IEEE 1516–2010 High-Level Architecture (HLA) standard, incorporating the emerging SpaceFOM standard [2].

The initial draft of SpaceFOM (version 0.1) was successfully implemented and tested during the 2017 SEE edition, marking a significant milestone in its development. Eleven universities from around the world participated, including the University of Alberta, University of Nebraska-Lincoln, Faculdade de Engenharia de Sorocaba (FACENS), University of Calabria, University of Genoa, University of Bordeaux, LMU Munich, Brunel University London, University of Liverpool, Jaipur National University, and New Bulgarian University. This edition simulated a lunar settlement facing a critical scenario: an asteroid on a collision course with the Moon. The exercise demonstrated SpaceFOM's initial capabilities to support interoperable simulations in complex, dynamic scenarios.

Based on insights gained from the 2017 SEE edition, the SpaceFOM underwent significant refinements to enhance the stability, robustness, and reliability of compliant Federates. The revised version (SpaceFOM v0.2) was introduced during the 2018 SEE event, where participants simulated settlements on both the Moon and Mars. This edition, held onsite in Sofia, Bulgaria, from May 8–10, 2018, saw the participation of ten universities collaborating both remotely and in-person. A notable feature of this simulation was the use of the Distributed Observer Network (DON), a NASA-developed real-time 3D visualization environment that monitors and displays updates from Federates in the simulation. Teams integrated their Federates with the DON Visualization Tool (DON-VT), enabling real-time observation of activities within the 3D environment, further enriching the immersive simulation experience.

Most SEE teams leveraged the SEE HLA Starter Kit [27], a development toolkit that is fully compliant with SpaceFOM and simplifies the process of developing HLA-based Federates. This toolkit provides essential functionalities for integrating HLA and SpaceFOM standards, significantly reducing the learning curve for students and enabling rapid prototyping and testing.

The SpaceFOM standard has proven particularly valuable in academic settings due to its extensive use of advanced HLAeatures, including time management and synchronization points, which are not ututilized as extensively in many simulations. This makes SpaceFOM an excellent platform for exploring and teaching advanced concepts in distributed simulation.

The SEE initiative well exemplifies how academia and industry collaboration fosters innovation, equipping the next generation of engineers and researchers with the skills needed to tackle complex, interdisciplinary challenges. It is expected that SpaceFOM's innovative patterns will continue to inspire broader adoption and cross-domain applications [16].

3.4 Supporting the ongoing use through a PSG

Once the standard was released a Product Support Group (PSG) was formed. The purpose of this group is to respond to questions, such as requests for interpretation, provide guidance, promote the standard and collect experiences from users. Proposed changes to the standard are collected in a dedicated Product Change Requests (PCR) format and analyzed. The PSG also reviews the standard periodically and can submit a product nomination for development of a new version. When a new version is developed, the collected PCRs serve as input for further development.

4 Towards a New SpaceFOM Version

There was a conscious decision made by the SpaceFOM Product Development Group (PDG) to keep the first version of SpaceFOM as simple as possible while capturing the fundamental constructs necessary to support space systems simulation. The philosophy was to establish a solid core set of datatypes and execution control policies first, deploy and test the base SpaceFOM, and then work both general and domain specific extensions in subsequent versions. The original SpaceFOM standard was released in January 2020. After 5 years of experience, some of which is discussed above in Section 3, the SpaceFOM Product Support Group (PSG) is looking toward the next SpaceFOM version.

The SpaceFOM PSG has been working on a number of PCRs over the past few years. While the domain, topic, and complexity of these PCRs varies, we will discuss the following three general areas of work in this paper: space systems modeling and datatype extensions, time management extensions, and execution control extensions.

4.1 Extended Datatypes

As stated above, the original SpaceFOM defined only the base datatypes, Object Classes, and Interaction Classes required for a functional but flexible modeling of space systems. Even before the original SpaceFOM was released in 2020, distributed space systems simulations existed and were in active use that required datatypes and object models not included in the SpaceFOM. This was not an oversight. The SpaceFOM PDG realized this and expected these operational systems to inform additions to the SpaceFOM base set. In the mean time, these deployed and developing systems could adopt the SpaceFOM and migrate their simulations. This section presents two important development activities that use the SpaceFOM and have identified possible extensions.

4.1.1 Object Classes for ESA HRAF3

A source of inspiration for the object class hierarchy of the next SpaceFOM version is the HRAF3 project [23]. This is an engineering level simulation where different subsystems are modeled.

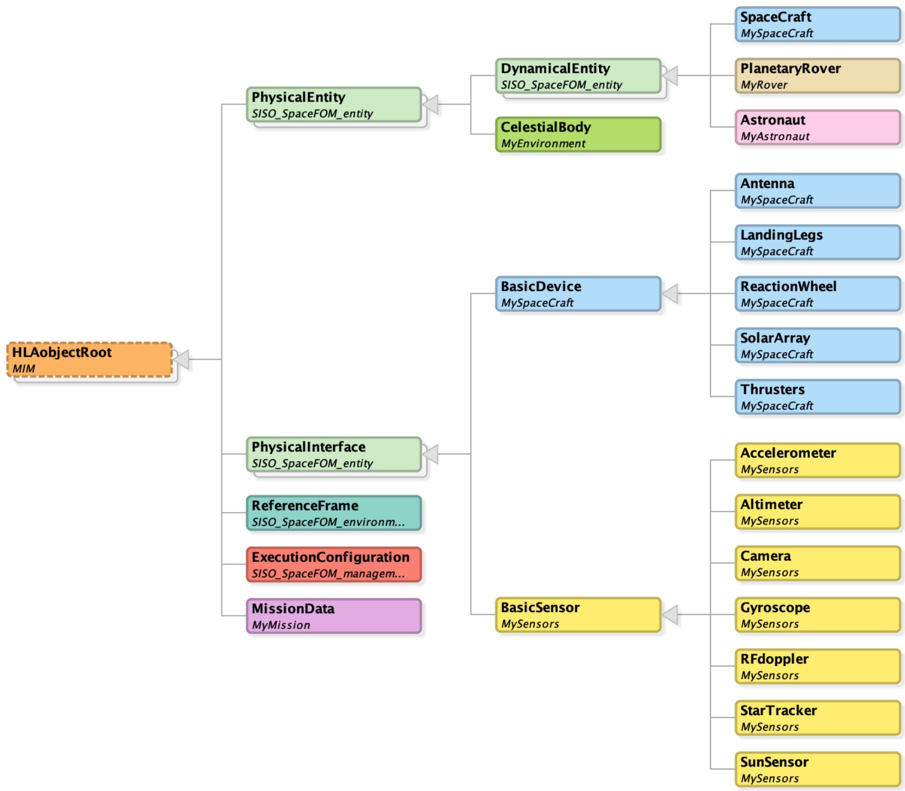


Figure 2: Generalized version of HRAF3 Object Class tree

Figure 2 shows a generalized version of the HRAF class hierarchy. Note the subclasses of the *PhysicalInterface* where sensors and other devices are modeled. Object instances of these classes are then combined into different vehicles. Additional subclasses can then be used to model and evaluate for example different types of sensors. Completely new types of classes are also added, such as MissionData.

4.1.2 Artemis Distributed Simulation Extensions

As discussed in Section 3.1, the NExSyS team is building the ADS framework to support the diverse simulation-based design, development, analysis, test, and integration needs of the Artimis program. While HLA and SpaceFOM provide the substantive necessary technology basis for ADS, additional common datatypes, message definitions, and execution protocols are required. These extensions constitute the ADS Federation Object Model (FOM). Some of these extensions and new items are specific to Artemis and the unique or specialized needs of the constituent Artemis elements. However, some of these are more general in nature and might make good additions to future SpaceFOM versions.

Details of the ADS FOMs are presented in a companion paper to this paper which includes the examination of the ADS FOM modules, ADS base datatypes, ADS SpaceFOM Object Class extensions, new ADS Object Classes, and new ADS Interaction Classes [22]. The ADS FOM is organized into the following constituent ADS FOM module files:

`ADS_FOM_datatypes.xml`. Additional base datatypes.

`ADS_FOM_environment.xml`. Datatypes associated with environmental models.

`ADS_FOM_systems.xml`. Datatypes associated with generalized space systems.

`ADS_FOM_graphics.xml`. Datatypes used for driving graphics systems.

`ADS_FOM_astronaut.xml`. Datatypes used for humans operating in a space environment.

`ADS_FOM_lander.xml`. Datatypes used by generalized planetary landing systems.

`ADS_FOM_rover.xml` Datatypes used by generalized planetary surface mobility systems.

The general hierarchy of the ADS FOM Object Classes is shown in Figure 3. This figure shows the dependence of the ADS FOM on the SpaceFOM *PhysicalEntity*, *DynamicalEntity*, and *PhysicalInterface* Object Classes. The relationships to the SpaceFOM base datatypes are not show but are significant and presented in detail in [22].

Note the similarities and differences between the HRAF3 FOM described in Section 4.1.1 and Figure 2 to the ADS FOM described above and shown in Figure 3. The SpaceFOM PSG will be exploring the similarities with an interest in finding common extensions to add in the next version of the SpaceFOM.

4.2 Time Management

4.2.1 Dynamically variable time steps

Although the SpaceFOM standard was designed specifically for the space domain, its current version has a significant limitation that affects the simulation of interplanetary missions: the Federation Time Step is fixed and cannot be modified during execution [1]. The existing standard supports conservative, time-stepped simulations using uniform and static time increments. This is achieved through the utilization of HLA Time Management services, which synchronize the progression of participating Federates via consistent HLA Logical Time (HLT) increments.

This approach enables both paced (e.g., real-time) and non-paced (e.g., as-fast-as-possible) Federation Executions. A Pacing Federate enforces deterministic behavior and ensures strict time consistency across all participants. While effective for many scenarios, this rigidity poses challenges when simulating the diverse and dynamic temporal requirements of interplanetary space missions.

Different mission phases often demand varying levels of temporal granularity. For example, critical operations like launches, orbit insertions, and dockings require fine-grained time steps to capture precise interactions and ensure accurate results. Conversely, less dynamic phases, such as interplanetary cruise or deep-space transit, can be adequately represented with coarser time steps, significantly reducing computational overhead. Additionally, unanticipated events during seemingly uneventful mission phases may necessitate an immediate shift from coarse-grained to fine-grained time steps to handle the emerging complexity effectively.

To address this limitation, a proposed extension to the SpaceFOM standard introduces the concept of Variable Federation Time Steps. As detailed in [28], this enhancement enables dynamic adjustment of time-step granularity during a simulation, allowing for seamless transitions between fine-grained and coarse-grained time steps. By aligning temporal resolution with mission-specific requirements, this capability enhances both the accuracy and flexibility of simulations, particularly during critical phases where temporal adaptability is paramount.

The proposed solution is designed with backward compatibility in mind, ensuring that existing SpaceFOM-based simulations can integrate the new functionality without significant rework. The transition to a variable time-step mode, referred to as Time-Granularity Change (TGC), is managed in a controlled and deterministic manner. The Master Federate plays a central role in coordinating these transitions, leveraging synchronization points, Execution Control Object (ExCO) updates, and wait loops. The ExCO [1], a core HLA object class in SpaceFOM, facilitates coordination by defining parameters for managing Federation timelines and transition states, ensuring consistency and preventing time-related conflicts.

Federates can request time-step adjustments based on their specific operational needs, but the final decision and implementation rest with the Master Federate to maintain overall Federation coherence. This approach balances flexibility with the rigor required for high-fidelity simulations.

While the immediate focus of this enhancement is to support complex space mission simulations, the underlying principles have broader applicability. Looking ahead, further developments aim to integrate time-stepped

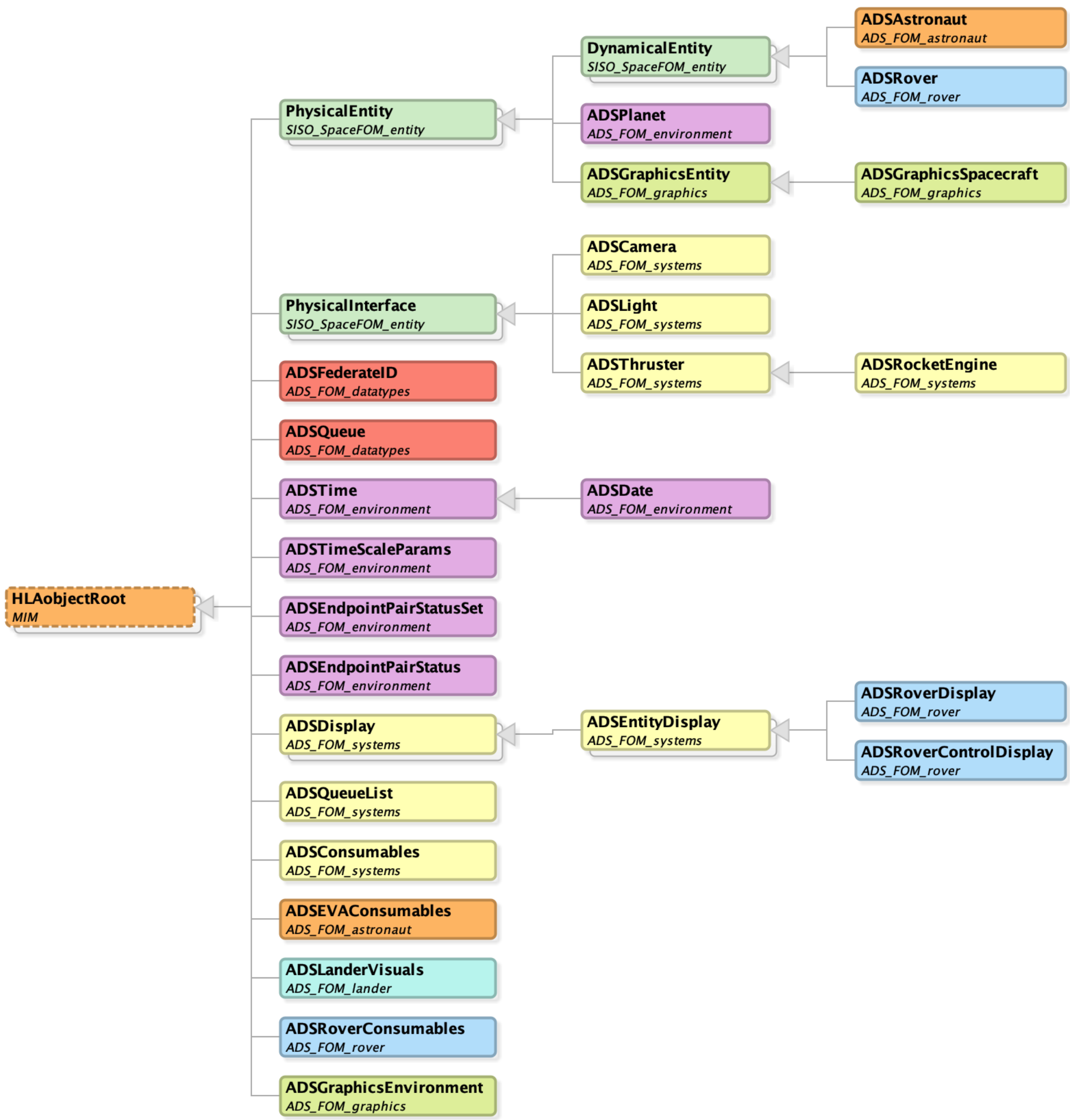


Figure 3: **ADS FOM Object Class Hierarchy**

and event-driven simulation approaches. This integration would allow critical, real-time events to be handled dynamically within a broader framework of time-stepped progression, combining the strengths of both methodologies.

4.2.2 Configurable Time Resolutions

The SpaceFOM standard was designed to use an HLT with a fixed resolution with a quantum unit of microseconds and is represented as an HLAInteger64Time data type. HLAInteger64Time is equivalent to HLAInteger64BE (big-endian 64-bit integer). This gives HLT a range of approximately ±292,271 Julian years. This resolution and range should accommodate all space simulations based on Newtonian mechanics. However, a microsecond resolution is not adequate to represent some commonly used scheduling frequencies. For example, to represent Timestamp Order (TSO) messages at 128 Hz (7812.5 microseconds/cycle) requires a time resolution exceeding the fixed microsecond units currently specified in the standard.

Fortunately, HLAInteger64Time data type can represent other time resolutions but with a trade-off in available range. For example, an HLT with nanosecond resolution could be represented with a range of approximately ±292 Julian years and would be adequate for most space simulations scenarios even of long duration. However, any HLT resolution value must be adopted by all federates in a federation execution. At a minimum both the FESFA and FCD would need to document the federation HLT resolution. Taking the concept one step further, an automated approach can be used to ensure all federates are using the same HLT resolution by extending the ExCO definition to include an HLT resolution attribute. During the SpaceFOM Initialization process the ExCO will get reflected to all joined federates.

4.3 Execution Control and Robustness

4.3.1 Save and Restore

The original SpaceFOM only provided HLA Save and Restore guidance based on the native HLA capability; no additional requirements of design are provided. However, using the HLA Save and Restore APIs alone will not ensure a reliable and robust result because repeatable Save and Restore success relies on mode transition synchronization, time homogeneity, and requires the exact set of federates at Save be present at Restore.

To ensure repeatability and time homogeneity, a SpaceFOM-compliant federation execution should only execute an HLA Save and/or Restore when the federation execution is in Freeze mode. Since all federation execution freezes occur on common HLA Logical Time Boundaries (HLTBs), all time managed federates will be in a time consistent state. In addition, executing the Save and/or Restore in freeze will avoid potential issues with advancing time during the Save or Restore operations.

Save is straightforward and just requires the coordination to freeze before initiating the Save. Restore can take place at either startup or at freeze and requires the exact set of federates to be joined at the time of the Save to be successful. Coordinating the Restore process involves getting to the exact set of joined federates and includes steps for error recovery. Managing the joined federates by waiting for federates to join and pruning unnecessary federates by forcing them to resign can be automated but also adds complexity. Adding error recovery makes the Save and Restore more robust and reliable but also increases complexity. Several Coordinated HLA Save and Restore designs have been created that trade off automation and error recover versus complexity and this is still a work in progress.

4.3.2 Loss of ExCO

SpaceFOM defines an execution control design that incorporates the use of a singleton Execution Configuration Object (ExCO) to coordinate state changes such as going to freeze or shutting down. The federate with the Master role as defined in SpaceFOM is responsible for managing execution control and sending updates to the ExCO. The Master federate performs a critical function and it would be detrimental if it unexpectedly leaves the federation execution. A SpaceFOM simulation should not run without a Master federate or an ExCO.

So how do we detect the loss of the Master federate without adding unnecessary complexity? The simplest approach is to detect the when the ExCO instance gets removed and then resign. Only the Master federate owns the ExCO so when the remove object instance service callback is received this covers the case of a Master federate resigning or the case of the ExCO inadvertently being removed. Nothing needs to be known about which federate is the Master federate.

4.3.3 Designated Late Joiners

SpaceFOM defines a coordinated initialization process for standing up a distributed space simulation. Depending on when a federate joins the initialization process it will be categorized as either an Early Joiner or Late Joiner. The Master federate waits for all required federates to join before allowing the initialization process to start and this will be the set of Early Joiner federates. Late Joiner federates are those that arrive after the coordinated initialization process is underway or has completed. Early Joiners are required to participate in the discovery of the Root Reference Frame, multiphase initialization, and the coordinated transition to run. All required federates will always be Early Joiners. Federates that are not required can either be an Early Joiner or a Late Joiner depending on when they start.

Non-required federates need to implement both the Early Joiner and Late Joiner initialization paths, which adds some complexity. Also, the only way to ensure a non-required federate will always be a Late Joiner is by starting it only after the last required federate has joined. It has been proposed that SpaceFOM support Designated Late Joiners than can start at any time and follow the Late Joiner initialization path.

One proposed design creates additional initialization steps only for the Designated Late Joiners, is backwards compatible and does not change anything for Early Joiners. Designated Late Joiners can start at any time and will still need to detect if they have been indicated to be either an Early or Late Joiner. If indicated to be a Late Joiner, then the Late Joiner Initialization path is followed. If indicated to be an Early Joiner, then the Designated Late Joiner should achieve all synchronization points to not deadlock the Early Joiner initialization process and then follow the Late Joiner initialization path once the 'initialization_completed' synchronization point is announced.

4.4 Other comments

In addition to the above, technical PCRs, there has also been requests for clarifications from the users, for example of the interpretation of certain attribute values. Editorial improvements are also common during the revision of a standard and may apply to the next revision.

The development of PCRs is ongoing and new PCRs with improvements and additions are still welcome.

5 Discussion and Conclusions

The SpaceFOM standard represents a significant milestone in the advancement of distributed simulation within the space domain. Building on proven solutions and practical, real-world experiences, its development was strongly supported by a collaborative engineering approach that was critical to its success. The first version of SpaceFOM has already been successfully applied in various scenarios and extended to address evolving needs, demonstrating its robustness and adaptability.

The SISO SpaceFOM Product Support Group (PSG) is committed to further improving the standard, broadening its scope, and providing proven solutions to meet the growing demands of the space industry. From the outset, the SpaceFOM Product Development Group (PDG) envisioned the standard as an evolving framework. By focusing the first version on core constructs, they laid a solid foundation for future development while addressing immediate challenges faced by the industry.

The authors underscore the importance of prioritizing core challenges and delivering a practical initial version. While not every idea could be implemented in the first release, the timely delivery of a functional standard enabled its adoption in real-world applications, fostered the growth of a dedicated user community, and provided invaluable insights for shaping future iterations.

Looking ahead, the PSG encourages broader participation from the community in the ongoing development of subsequent versions. Such collaboration is vital to ensuring that SpaceFOM continues to address emerging needs and leverage collective expertise.

The authors are excited to contribute to the development of the next version of SpaceFOM alongside the expanding user community. Together, they aim to advance the capabilities of distributed simulation and further strengthen its role in the space industry's ongoing efforts to tackle complex challenges.

Acronyms

ADS Artemis Distributed Simulation framework

API Application Programming Interface

ATV ESA Automated Transfer Vehicle

dt Federate Time Step

CCT Computer Clock Time

DON Distributed Observer Network

DON-VT DON Visualization Tool

ER NASA JSC Software, Robotics, and Simulation Division

ESA European Space Agency

ExCO SpaceFOM Execution Control Object

FCD Federate Compliance Document

FESFA Federation Execution Specific Federation Agreement

FOM Federation Object Model

HITL Human-In-The-Loop

HLA High Level Architecture

HLT HLA Logical Time

HLTB HLA Logical Time Boundary

HRAF Harwell Robotic and Autonomy Facility

HTV JAXA H-II Transfer Vehicle

HWITL Hardware-In-The-Loop

IEEE Institute for Electrical and Electronics Engineers

JAXA Japanese Aerospace Exploration Agency

JSC Johnson Space Center

M&S Modeling and Simulation

NASA National Aeronautics and Space Administration

NExSyS NASA Exploration Systems Simulations team

PCR Product Change Request

PDG SISO Standard Product Development Group

PSG SISO Standard Product Support Group

RSA Russian Space Agency

RTI Runtime Infrastructure

SEE Simulation Exploration Experience

SISO Simulation Interoperability Standards Organization

SpaceFOM Space Reference Federation Object Model

TGC Time Granularity Change

TSO Timestamp Order

VV&A Verification, Validation, and Accreditation

References

[1] E. Crues, D. Dexter, A. Falcone, A. Garro, M. Madden, B. Möller, and D. Ronnfeldt. *SISO-STD-018-2020: Standard for Space Reference Federation Object Model (SpaceFOM)*. Simulation Interoperability Standards Organization (SISO), 2019.

[2] Simulation Interoperability Standards Organization/ Standards Activities Committee (SISO/SAC). *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules*. Technical Report IEEE-1516-2010, The Institute of Electrical and Electronics Engineers, 2 Park Avenue, New York, NY 10016-5997, August 2010.

[3] Simulation Interoperability Standards Organization/ Standards Activities Committee (SISO/SAC). *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Federate Inferface Specification*. Technical Report IEEE-1516.1-2010, The Institute of Electrical and Electronics Engineers, 3 Park Avenue, New York, NY 10016-5997, August 2010.

[4] Simulation Interoperability Standards Organization/ Standards Activities Committee (SISO/SAC). *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Object Model Template (OMT) Specification*. Technical Report IEEE-1516.2-2010, The Institute of Electrical and Electronics Engineers, 3 Park Avenue, New York, NY 10016-5997, August 2010.

[5] Simulation Interoperability Standards Organization/ Standards Activities Committee (SISO/SAC). *IEEE Standard for Distributed Interactive Simulation*. Technical Report IEEE-1278-2015, The Institute of Electrical and Electronics Engineers, 2 Park Avenue, New York, NY 10016-5997, August 2015.

[6] L. Arguello, L. Dwedari, G. D. Lauderdale, A. Vankov, and P. Chliaev. ESA-NASA Distributed Simulation Experiment: First Results and Lessons Learned. In *Proceeding of the European Simulation Interoperability Workshop (EURO-SWIG)*, London, England, June 2001. SISO, SISO.

[7] Grace D. Lauderdale, Edwin Z. Crues, Danielle V. Snyder, and David A. Hasan. A Feasibility Study for ISS and HTV Distributed Simulation. In *AIAA Modeling and Simulation Technology Conference*, Austin, Texas, August 2003. AIAA, AIAA.

[8] Robert G. Phillips, Daniel E. Dexter, David A. Hasan, and Edwin Z. Crues. An Overview of the Distributed Space Exploration Simulation (DSES) Project. In *Spring 2007 Innovation Workshop and Conference*, Norfolk, Virginia, March 2007. SISO, SISO.

[9] Robert G. Phillips, Daniel E. Dexter, David A. Hasan, and Edwin Z. Crues. A Coordinated Initialization Process for the Distributed Space Exploration Simulation (DSES). In *Spring 2007 Innovation Workshop and Conference*, Norfolk, Virginia, March 2007. SISO, SISO.

[10] Victoria I. Chung, Edwin Z. Crues, Michael G. Blum, Catherine A. Alofs, and Juan M. Busto. An Orion/Ares I Launch and Ascent Simulation - OneSegment of the Distributed Space Exploration Simulation (DSES). In *AIAA Modeling and Simulation Technologies Conference and Exhibit*, Hilton Head, South Carolina, August 2007. AIAA, AIAA.

[11] Esther H. Jennings, Michael G. Blum, Juan M. Busto, Victoria I. Chung, Edwin Z. Crues, Daniel E. Dexter, David A. Hasan, and Joe Hawkins. Integrating Collaborative Distributed Simulations for Space Exploration Missions. In *AIAA Modeling and Simulation Technologies Conference and Exhibit*, Honolulu, Hawaii, August 2008. AIAA, AIAA.

[12] Edwin Z. Crues, Alexander Lin, and David A. Hasan. Integrating HLA into the Trick Simulation Development Environment. In *AIAA Modeling and Simulation Technology Conference*, Austin, Texas, August 2003. AIAA, AIAA.

[13] Robert G. Phillips and Edwin Z. Crues. A Feasibility Study for ISS and HTV Distributed Simulation. In *Fall 2005 Innovation Workshop and Conference*, Orlando, Florida, September 2005. SISO, SISO.

[14] Priscilla R. Elfrey, Gregory Zacharewicz, and Marcus Ni. Smackdown: Adventures in Simulation Standards and Interoperability. In *Proceedings of the 2011 Winter Simulation Conference*, Phoenix, Arizona, December 2011. WSC, IEEE.

[15] Inc. Simulation Exploration Experience. Simulation Exploration Experience. SEE Website: `https://simulationexplorationexperience.org`, September 2025.

[16] Edwin Z. Crues, Dan Dexter, Alberto Falcone, Alfredo Garro, and Björn Möller. SpaceFOM - a robust standard for enabling a-priori interoperability of HLA-based space systems simulations. *Journal of Simulation*, 16(6):624–644, 2022.

[17] Simulation Interoperability Standards Organization. SISO-ADM-003-2011: Balloted Products Development and Support Process. `http://www.sisostds.org`. Accessed: 2025-01-20.

[18] National Aeronautics and Space Administration. Moon to Mars Architecture. `https://www.nasa.gov/moontomarsarchitecture/`.

[19] National Aeronautics and Space Administration. The Artemis Accords. `https://www.nasa.gov/artemis-accords/`.

[20] Keaton Dodds, Edwin Z. Crues, and Jason Harvey. Using Open Standards and NASA Open Source Simulation Tools to Model Artemis Base Camp Mission Timelines. In *2023 IEEE Aerospace Conference*, Big Sky. Montana, March 2023. IEEE, IEEE.

[21] Keaton Dodds, Paige Whittington, Edwin Z. Crues, James Gentile, and Nicolas Trevino. Using Distributed Simulation Capabilities to Support Studies and Mission Operations for NASA Artemis Program. In *2025 IEEE Aerospace Conference*, Big Sky. Montana, March 2025. IEEE, IEEE.

[22] Edwin Z. Crues, Paige Whittington, and Keaton Dodds. Development of the Artemis Distributed Simulation FOMs. In *SISO 2025 Simulation Innovation Workshop*, Orlando, Florida, February 2025. SISO, SISO.

[23] B. Möller, T. Gray, A. Kisdi, K. Buckley, and J. Delfa. Experiences from the SISO SpaceFOM at the European Space Agency. In *SISO 2021 Simulation Innovation Workshop*, 2021.

[24] S. Kay, A. Kisdi, K. Buckley, P. Bocciarelli, A. D'Ambrogio, I. Dankiewicz, J. Kemp, J. Delfa, and J. Ocon. Development of a Distributed Simulation Environment and Model Driven Engineering Framework to Support the Verification and Validation of Complex Autonomy Components. In *International Astronautical Congress, IAC 2021*, 2021.

[25] The United Kingdom Government. UK Space Agency blog. Enabling Technologies Programme: Aiding Exploration. `https://space.blog.gov.uk/2024/10/07/enabling-technologies-programme-aiding-exploration/`. Accessed: 2025-01-20.

[26] G. Weblin, R. Spowage, and B.Shirnia. Integrating HLA and SpaceFOM in SIMULUS-NG, In press 2025.

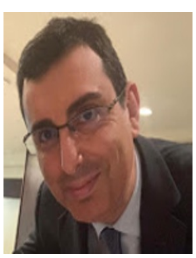
[27] A. Falcone, A. Garro, S.J.E. Taylor, A. Anagnostou, N.R. Chaudhry, and O. Salah. Experiences in Simplifying Distributed Simulation: the HLA Development Kit Framework. *Journal of Simulation*, 11(3):208–227, 2017.

[28] R. E. De Grande, A. Falcone, and A. Garro. Adding Flexibly in Distributed Simulation of Space Missions by Enhancing the SpaceFOM standard. In *International Symposium on Distributed Simulation and Real Time Applications, DS-RT 2024*. IEEE - Institute of Electrical and Electronics Engineers Inc., October 2024.

Author Biographies

Björn Möller: *is the President and co-founder of Pitch Technologies. He leads the development of Pitch's products. He has more than twenty-five years of experience in international high-tech R&D companies, in areas such as modeling and simulation, artificial intelligence and web-based collaboration. Björn Möller holds a M.Sc. in Computer Science and Technology after studies at Linköping University, Sweden, and Imperial College, London. He currently serves as the chair of the SISO SpaceFOM Product Development group, the chair of the SISO RPR FOM Product Development Group. and the vice chair of the SISO HLA Evolved Product Development Group.*

Edwin Z. {Zack} Crues: *received B.S., M.S., and Ph.D. degrees in Aerospace Engineering from the University of Texas, Austin in 1983, 1985, and 1989 respectively. Zack has over 35 years of professional experience in developing spacecraft simulation and simulation technologies internationally. Zack currently works at NASA Johnson Space Center in Houston, Texas as a Space Systems Simulation Architect in the Simulation and Graphics Branch where he leads the development of simulation technologies and the application of those technologies in the simulation of NASA's current and proposed human spacecraft. He was a founding member of the Simulation Exploration Experience (SEE) in 2011 and still participates as a technical advisor (`https://simulationexplorationexperience.org`). Zack is also a member of the SISO Executive Committee and the SpaceFOM Product Support Group.*



Alfredo Garro: *Alfredo Garro received a PhD in Systems and Computer Engineering from the University of Calabria (Italy), where he is currently an Associate Professor of Computer and Systems Engineering with the DIMES. In 2016, he was Visiting Scientist at NASA JSC, working with the Software, Robotics, and Simulation Division (ER). His main research interests include M&S, multi-agent systems, systems and software engineering, and reliability engineering. His list of publications contains more than 120 papers published in international journals, books, and proceedings of international and national conferences. He is the co-founder (in 2014) and director (from 2018) of the Departmental Research Laboratory "System Modeling And Simulation Hub Lab (SMASH Lab)". He is vice chair of the SpaceFOM PDG and PSG of SISO. He was the President of the "Italian Chapter" of INCOSE. He is a member of the Executive Committee of the SEE (Simulation Exploration Experience) project. He is involved as an IEEE Senior Member in the activities of the IEEE Computer Society, IEEE Reliability Society, and IEEE Aerospace and Electronic Systems Society.*

Alberto Falcone: *is a researcher at the Institute for High Performance Computing and Networking (ICAR) of the Italian National Research Council (CNR). He received a PhD in Systems and Computer Engineering from the Department of Informatics, Modeling, Electronics, and Systems Engineering (DIMES) of the University of Calabria (Italy). From January to October 2016 he was a Visiting Researcher at NASA JSC, working with the Software, Robotics, and Simulation Division (ER). His main research interests include M&S, distributed simulation, systems and software engineering. He is an Associate Editor for the Journal SIMULATION: Transactions of The Society for Modeling and Simulation International. He served as Tutorial Chair for the 2019 Spring Simulation Conference (SpringSim'19).*

Daniel E. Dexter: *is an engineer in the Spacecraft Software Engineering Branch in the Software, Robotics and Simulation Division of the Engineering Directorate at NASA's Johnson Space Center in Houston, Texas. Daniel has over 30 years of software and simulation development experience including the areas of digital signal processing, image processing, neural networks, distributed supercomputing, embedded flight software, and distributed simulations. Daniel is the principal developer of the TrickHLA software package, a NASA developed middleware software package for using the HLA distributed simulation standard with NASA standard M&S tools.*