Experiences from the SISO SpaceFOM at the European Space Agency

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

Tom Gray, Pitch Technologies Ltd
3 Churchill Court, Manor Royal
RH10 9LU Crawley
United Kingdom
tom.gray@pitchtechnologies.com

Steven Kay, Aron Kisdi, Karl Buckley, GMVNSL
HQ Building, Thomson Avenue, Harwell Campus
OX11 0GD Didcot
United Kingdom
skay@gmvnsl.com

Juan Delfa, European Space Agency
Fermi Avenue, Harwell Campus
OX11 0FD Didcot
United Kingdom
Juan.Delfa@esa.int

Keywords:
Space, SpaceFOM, ESA, HLA, Hardware-in-the-Loop, MATLAB, Simulink, Robotics, Autonomy, Mars

ABSTRACT: *The Harwell Robotic and Autonomy Facility (HRAF) activities funded by the European Space Agency (ESA) aim to provide advanced capabilities to support the development and testing of complex autonomous systems for the exploration of our Solar System. The outcome of one of these activities is a a flexible simulation environment allowing models and real hardware to be combined, compared and tested in a plug and play mode.*

HRAF has carried out three pilot studies on the use of simulation concepts. This paper presents experiences from Pilot 3, in particular from the task of developing a Federation specialized for space exploration scenarios.

The first scenario is concerned with Mars Sample Return (MSR). Specifically, the mission phase where the Orbiting Sample (OS) is retrieved by a Chaser spacecraft (ERO) in Mars orbit for later return and analysis on Earth. The guidance, navigation and control (GNC) functionality using Image based Navigation techniques is accompanied by a high-fidelity Physics "real-world" simulator. The second scenario is concerned with the soft, precision landing of a Spacecraft on a low gravity Near Earth Object (NEO).

The federation is based on the generic standard for distributed simulation: High Level Architecture (HLA IEEE 1516-2010), together with the associated SISO Space Reference FOM standard (SISO-STD-018 "SpaceFOM"). Different configurations of the Federation are constructed, the MSR Scenario considering a Model-in-the-Loop (MIL) and Hardware-in-the-Loop (HIL) configuration and the NEO Scenario implementing MIL, Processor-in-the-Loop (PIL) including Synthetic Image generation and HIL configurations. In many cases the same functionality is provided as MIL, PIL and HIL and federates can be exchanged between executions. The federation can also be run locally or distributed between ESA and contractor sites.

Preliminary conclusions are that a baseline federation has been successfully developed, which can be reused and form a starting point for future experiments, and that the SpaceFOM was helpful in this integration. Some challenges experienced include how to integrate reused and complex Matlab/Simulink models in federations and how to integrate existing hardware with particular timing requirements. Some feedback to SISO is also planned for the SpaceFOM standard.

This project was funded by the European Space Agency under ESA Contract No. 4000124144/18/NL/PA.

1. Introduction

Simulation is widely used in the space domain, from concept development, analysis, engineering and testing to astronaut training and in support of actual flight [1]. Using simulation, space professionals can mitigate some of the danger and expense associated with spaceflight. Distributed simulation makes it possible to combine simulation models from different teams to simulate even more complex systems. This may be models from experts from the same organization or from different organizations, that provide components for a joint mission.

Integrating simulations into a distributed simulation is not without challenges. It can be both complex and costly, in particular if each integration effort starts from zero. Using a standardized approach makes it possible to reuse methodology, tools and not least knowledge, such as best-practices. If standards can be shared and be made open and publicly available, a community and an eco-system can evolve over time.

This paper presents experiences from applying the recently published SISO Space Reference Federation Object Model (SpaceFOM) [1] standard at the ESA-funded Harwell Robotic and Autonomy Facility in the UK.

1.1 SISO Standards

Simulation Interoperability Standards Organization (SISO) [3] is an organization that produces open international standards for simulation interoperability. SISO is an independent organization and anyone can participate in standards development activities. SISO is a sponsor of the Institute of Electrical and Electronics Engineers (IEEE) [4] that publishes several of the standards that SISO develops.

SISO standards are developed using a well-defined process, called the Balloted Product Development and Support Process (BPDSP) [5], which specifies two main phases for standards development: the development phase, carried out by a Product Development Group (PDG) and the support phase, carried out by a Product Support Group (PSG), which in turn can initiate the development of new versions of a standard.

SISO initially developed standards mainly for the defense domain, but later extended the scope, with standards for manufacturing for example. Initial defense simulation standards, such as DIS [6], had hard-coded information models for specific sub-domains, in particular defense vehicles (“platforms”) and soldiers. To be able to support a broader range of domains, a domain-independent interoperability standard, called the High Level Architecture (HLA) [7], was developed. HLA has since formed the basis for distributed simulation in many other domains, including space.

1.2 The High Level Architecture

The SpaceFOM builds upon the HLA standard, which is a generic and domain independent standard for simulation interoperability. One prominent feature of HLA is that it facilitates the development of information models, called Federation Object Models (FOM), for any simulation domain. A FOM describes the shared objects and interactions that are exchanged in a distributed simulation. A FOM can be provided as modules for better separation of concern, and to better support development by different teams. There are standardized FOMs, often known as reference FOMs, that can be extended using project specific FOM modules.

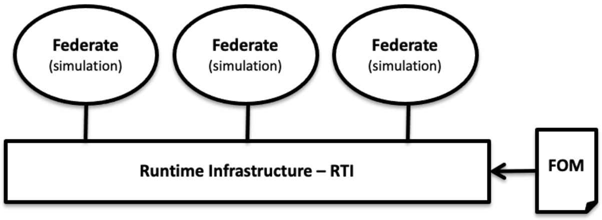


Figure 1: HLA federation

Some other key HLA concepts, as shown in Figure 1 include:

- Federate, which is a participating simulation, that exchanges information with other federates.
- Federation, which is the set of federates that simulate a shared scenario.
- Runtime Infrastructure (RTI). All federates connect to the RTI to use services, such as information exchange.

The RTI provides a generic set of services to perform the following:

- Federation Management, that manages a well-defined set of federates and offers synchronization and save/restore.
- Declaration Management, that enables federates to declare what information they wish to publish (send) and subscribe (receive).
- Object Management, whereby federates can share object instances with attributes and exchange interactions.
- Ownership Management, whereby federates can negotiate which federate that is responsible for updating what attributes of which object instances.
- Time Management, whereby timestamped updates and interactions can be exchanged and the time advancement of federates can be controlled. This makes it possible to guarantee that federates have all the data in time to perform correct calculations.
- Data Distribution Management, which makes it possible for each federate to filter incoming data based on its interest.

There are also additional federation management services in the Management Object Model, that makes it possible to inspect and adjust the federation programmatically. Each federation can choose which HLA services it needs to use to reach its objectives.

1.3 The SISO SpaceFOM

The SISO SpaceFOM is a new standard that was published in 2020. It builds upon HLA and provides a FOM that is tailored for simulating space scenarios. It also specifies how to use the HLA services. The main components are:

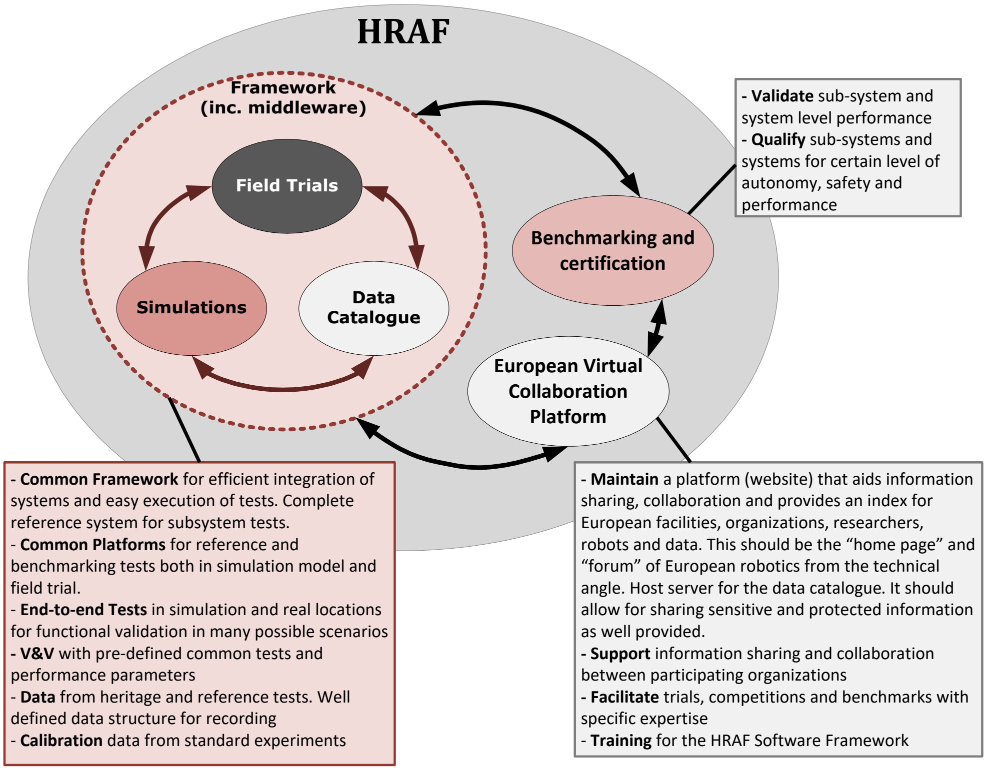
- Reference frames, whereby time-space coordinates can be expressed using a set of coordinated reference frames.
- Standard reference frames, frequently used for space simulation, for example SunCentricInterial and EarthMJ2000Eq.
- Standardized timelines and time representations for scenario time, epoch and more.
- Time management modes, such as real-time, scaled real-time, as-fast-as-possible and hard real-time.
- Execution control for multi-phase initialization, run, freeze and shutdown.
- Generic object classes for key concepts. These can be extended by subclassing. They include "PhysicalEntity" (that can be used as a basis for celestial objects), "DynamicalEntity (that can be used as a basis for space vehicles, astronauts, etc) and "PhysicalInterface" (that can be used as a basis for docking ports and parts of space vehicles, such as sensors, landing legs, etc).

The SpaceFOM also provides a template for a Federation Execution Specific Agreement (FESFA). This is used to document parameters that are specific for a particular federation, like reference frames used, time step, etc. Another template, called Federate Compliance Declaration (FCD) can be used to specify the capabilities of a federate. This is useful for assessing the suitability of a federate in a new federation or scenario.

2. Harwell Robotic and Autonomy Facility

The Harwell Robotic and Autonomy Facility (HRAF) is a facility funded by the European Space Agency (ESA) that provides advanced capabilities to support the design, development, verification and validation of complex autonomous systems, examples of which are planetary rovers, high precision landing, and remote sample collection, manipulation and analysis. These types of systems are critical for enabling future planetary exploration missions. The facility has three main elements:

a flexible simulation environment that allows models and real hardware to be combined and compared in a plug and play mode, a service to run field trials, and a data archive of the results acquired. Although built for space activities, the facility has been designed to be flexible so that it could also be used to support autonomy, verification and validation in other sectors [9]



- **Validate** sub-system and system level performance
- **Qualify** sub-systems and systems for certain level of autonomy, safety and performance

- **Common Framework** for efficient integration of systems and easy execution of tests. Complete reference system for subsystem tests.
- **Common Platforms** for reference and benchmarking tests both in simulation model and field trial.
- **End-to-end Tests** in simulation and real locations for functional validation in many possible scenarios
- **V&V** with pre-defined common tests and performance parameters
- **Data** from heritage and reference tests. Well defined data structure for recording
- **Calibration** data from standard experiments

- **Maintain** a platform (website) that aids information sharing, collaboration and provides an index for European facilities, organizations, researchers, robots and data. This should be the "home page" and "forum" of European robotics from the technical angle. Host server for the data catalogue. It should allow for sharing sensitive and protected information as well provided.
- **Support** information sharing and collaboration between participating organizations
- **Facilitate** trials, competitions and benchmarks with specific expertise
- **Training** for the HRAF Software Framework

Figure 2: The concept of HRAF

2.1 The HRAF Pilot Federations

The development of the distributed simulation capability within the Harwell Robotics and Autonomy Facility has pursued an incremental approach during a series of pilot activities. Whereby additional (new, replacement or alternative) functionalities within the HRAF architecture are developed, or some specific autonomy functionalities may be applied and utilized using existing HRAF infrastructure. With this paradigm, it is foreseen that over successive pilots, the core architecture and therefore the capabilities of HRAF shall mature into well-defined, self-contained and exchangeable software modules providing critical infrastructure to support integration, verification and validation of autonomy components at system and mission level.

HRAF Pilot 1 addressed the overall requirements definition and the development of a core architecture for performing simulations and also a dynamic archive. It was at this time High Level Architecture (HLA) was selected as the middleware of choice. The pilot however stopped short of deploying this simulation in a distributed manner. Pilot 1 tested this architecture through application, using a long-range rover navigation scenario as a demonstration case. Pilot 1 pre-dates the conception of the SISO SpaceFOM and therefore a project specific FOM was created as part of the demonstration activity [7].

HRAF Pilot 2 added further functionality to the overall architecture of HRAF, in the areas of the dynamic archive as well as adding an additional, alternative robotic autonomy component for performing the Visual Odometry (VO) function of a planetary robot through integrating the SPARTAN system [10]. In addition, the element of field trial logistics within HRAF was expanded and built upon.

HRAF Pilot 3 (HRAF3), a currently on-going activity, has the aims of; introducing a new Federation using the SISO SpaceFOM standard, which is capable of performing distributed simulations (HLA-DS) in the domain of planetary exploration, as well as developing and validating a model-driven HLA-DS engineering framework for supporting the semi-automated generation of Federation Object Models (FOMs) for any space domain activity.

3. Scenarios for HRAF3

Two mission scenarios were selected at the outset of the Pilot activity: “Autonomous Rendezvous and Capture in Mars Orbit” and “Precision landing of a spacecraft on a low gravity Near Earth Object (NEO)”. Chosen by the European Space Agency, these scenarios reflect modern, complex scenarios, which are being actively developed and rely on autonomy components within a larger Guidance, Navigation and Control (GNC) system for successful completion of critical mission phases.

3.1 Mars Sample Return - Autonomous Rendezvous and Capture in Mars Orbit

On Earth orbit, both manned and unmanned missions have demonstrated docking and even capture at large scale, however as part of the Mars Sample Return (MSR) campaign, it is planned to perform an autonomous Rendezvous and Capture (RvC) of a passive object in Mars Orbit. The passive object, called the Orbiting Sample, will contain samples from the Martian surface and be placed into orbit onboard a small launcher, the Mars Ascent Vehicle (MAV), before being released ready for capture. An orbiter, the Earth Return Orbiter (ERO), shall track and rendezvous with the Orbiting Sample and then performing a capture maneuver using Onboard Autonomy. The RvC phase described is a part of a larger Mars Sample Return campaign, which is a collaboration between NASA and ESA [13].

3.2 Precision landing of a spacecraft on a low gravity Near Earth Object

Several asteroid sample return missions are being developed or planned in the near future (JAXA’s Hayabusa-2, NASA’s Osiris-Rex, ESA’s Marco Polo & Marco Polo-R). The intended targets are small (~1 km) near-Earth asteroids. The unknown properties of the surface and the reduced size of the asteroid demand very high landing accuracy (few meters), making the verification & validation of the GNC and Autonomy solution for the Descent & Landing (D&L) phase critical.

GMV, during previous ESA and European Council H2020 activities, successfully developed, matured and performed the Verification & Validation following the full design, development and verification/validation cycle on a FES simulator and GNC solution with visual based relative navigation for the Descent & Landing phase onto a Near Earth Object [13], which was adopted for implementation in the HRAF Framework. The scenario considers the D&L onto the binary asteroid system 1996FG3, also the subject of the previous ESA MarcoPoloR mission study [15].

3.3 Model-, Processor- and Hardware-in-the-Loop Simulators

The scenarios required that the design, development and verification/validation (DDVV) approach conventionally applied to GNC and associated Autonomy solutions be implemented using HLA-DS principles. Specifically, the activity introduces the capability of performing simulation of both scenarios using the following key steps in the DDVV cycle:

- at Model-in-the-Loop (MIL) level, using reference models of the GNC / Autonomy algorithms alongside a high-fidelity Functional Engineering Simulators (FES) including Physics, Craft and Sensor models. Typically, tools such as Matlab/Simulink shall be used for the development of such simulators.
- at Processor-in-the-Loop (PIL) level (for the NEO scenario), the FES-validated GNC / Autonomy algorithms shall be translated (typically to C-code using autocoding techniques) to flight representative avionics hardware for non-real-time or real-time testing. The testbench is completed with the introduction of the FES (which may be autocoded and deployed onto Real-time Hardware) as the “Real World” model, completed with representative interfaces to the Onboard Software (OBSW). Additional components may augment the testbench as required, such as the inclusion of a suitable Image Generation tool when visual-based algorithms are under test.
- at Hardware-in-the-Loop (HIL) level, the PIL testbench is extended, and simulated environments and sensors (e.g. cameras) are replaced with dynamic conditions in “realistic” conditions on-ground, typically using robotic arms to implement the computed kinematics (and contact dynamics if required) in an environment representative of the flight conditions.

Given the huge complexity of developing high-fidelity simulators for the selected scenarios from scratch, it was critical that existing simulators from previous or on-going GNC activities were reused (and adapted where required) during the HRAF activity, and integrated into the framework.

For implementation of the Mars Sample Return scenario in HRAF, the Model-in-the-Loop configuration considered the mission phases: Intermediate Range (~60km – ~5km range) → Short Range Phase (~5km – 100m) → Forced Motion Capture

(100m – 0m). A GMV developed Simulink based FES and GNC Closed-Loop Simulator was selected for reuse in this configuration. The simulator was developed and successfully demonstrated by GMV through previous ESA activities[12]. To reduce the computational effort required in this long scenario, behavioral models of the Image Processing (Short Range and Long Range) part of the Navigation chain is adopted. During the execution of simulations for this configuration, federates shall be distributed physically between the GMV-UK site and the ESA/ECSAT site, both located in the Harwell Campus, UK.

The MSR Hardware-in-the-Loop configuration considered only the Forced Motion Capture phase (from ~60m → 0m), due to physical limitations in scaling the scenario within the Robotic Arm facility, and made use of real cameras "in the loop" mounted onto the robotic arm. This configuration introduces the additional GMV-ES site, which hosts the GMV owned and operated Robotic Arm facility: Platform-Art®, located in Madrid, Spain [11].

For the Near Earth Object (NEO) scenario Model-in-the-Loop configuration the FES and GNC algorithms are utilized as a Simulink Closed-Loop Simulator. Again, behavioral Image Processing for the relative navigation is used to reduce computational processing requirements. The same GMV-UK and ESA/ECSAT sites are used during MIL simulations. At Processor-in-the-Loop level, the Image Generation tool, PANGU [16] shall be integrated into HRAF, providing realistic synthetic images of the descent. PANGU is deployed as a component at the ESA/ECSAT site. The FES is autocoded and deployed onto a dSPACE DS1006 Real-time computer, which is then also integrated with the Federation for this configuration and is located at the GMV-UK site. The GNC solution may be included at either a Software-in-the-Loop level using an autocoded version of the solution, but harnessed inside a Simulink model for rapid utilization, or at a Processor level using the software deployed to a representative LEON2 processor.

Lastly, at the Hardware-in-the-Loop level for the NEO scenario, the Robotic Arm Facility, GRALS, located at ESA/ESTEC is introduced into the architecture, complete with camera mounted to the robot arm and a realistic terrain mockup of the landing site. The FES and GNC options remain the same as with the PIL configuration.

4. The HRAF3 Federation

4.1 Federation Architecture and selection of Federates

The first step in defining the Federation Architecture was to evaluate the simulators intended for integration into the HRAF architecture. As the goal was to create a distributed simulation using these simulators, it was important to evaluate where the Closed-loop simulators could be broken down into smaller simulation components, allowing them to be deployed in a distributed manner.

Through identifying commonalities between the simulator implementations and considering the concepts defined during Pilot 1, it was possible to define a (largely) common set of federate types that would be responsible for specific functional aspects of the overall simulator.

- **GNC, VBN and IP** federates being responsible for the onboard software, autonomy and image processing models/hardware.
- **Craft** federates are responsible for implementing models of the Spacecraft onboard sensors and actuators.
- **Environment** federates responsible for implementing the required Ephemerides.
- **Physics** federates are responsible for models of physical effects (e.g. gravitational or perturbations) (known as the Dynamics, Kinematics, Environment (DKE) respectively) applied to the participating objects (e.g. Spacecraft).
- **Instrumentation** federates are responsible for implementing additional sensors may perform some scenario-specific function, such as a Navigation camera or Laser Altimeter. Additionally,
- **System Operations** federates include the Master-Pacer (fulfilled by the Space Master, provided by Pitch), a Data Logger / Mission Display to view and collect the scenario data for later analysis, and a 3D visualizer to allow visualization of the scenario. The federate responsible for interfacing with the Robotic Arms during HIL configurations is considered part of "System Operations" also.

Given that this new Federation considered five different simulation configurations, it was important to consider the selection of Federates such that there was a balance between the numbers of federates and the development and test complexity required for any given federate. For a federate to support multiple configurations of the Federation, this implies supporting various Multi-Phase Initialization (MPI) sequences, interfaces and potentially different types of internal interface to simulation models and/or hardware.

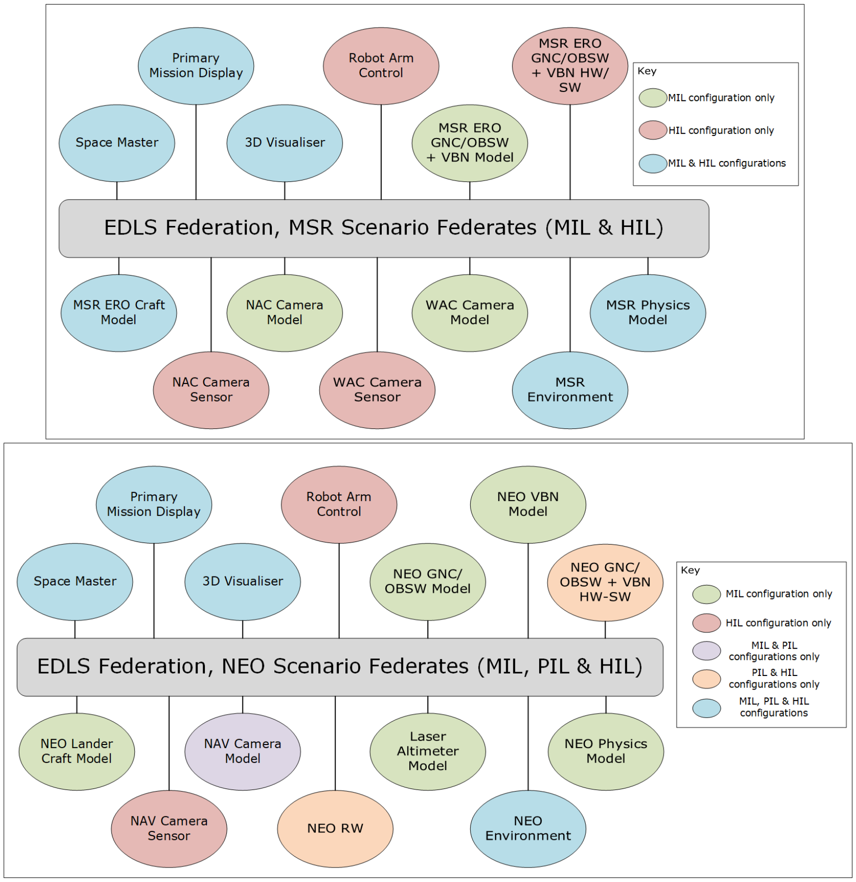


Figure 3: EDLS Federation considered per Scenario

4.2 FOM Design

Figure 4 gives an overview of the main FOM modules in HRAF3.

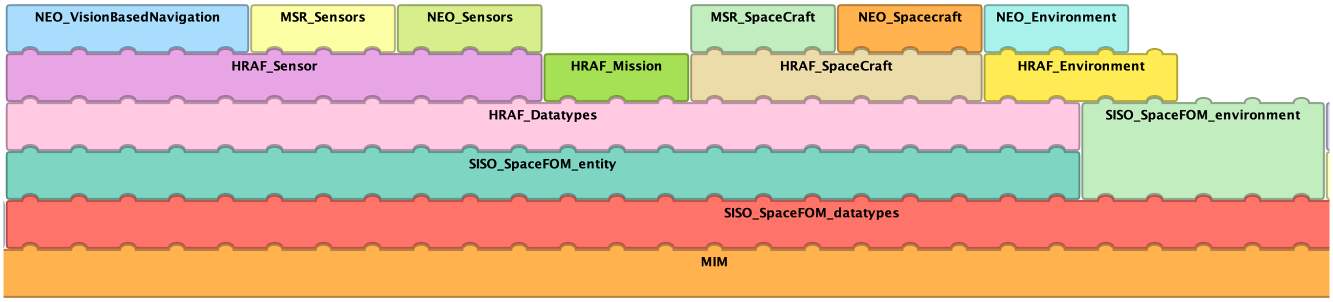


Figure 4: Main HRAF3 FOM modules

The standardized SpaceFOM modules have names prefixed by “SISO_SpaceFOM” and the HRAF modules are prefixed with “HRAF”. There are specific extensions for the Mars Sample Return (MSR) and Near Earth Object (NEO) scenarios in separate modules. This can be observed both for the HRAF_Sensor and the HRAF_SpaceCraft modules.

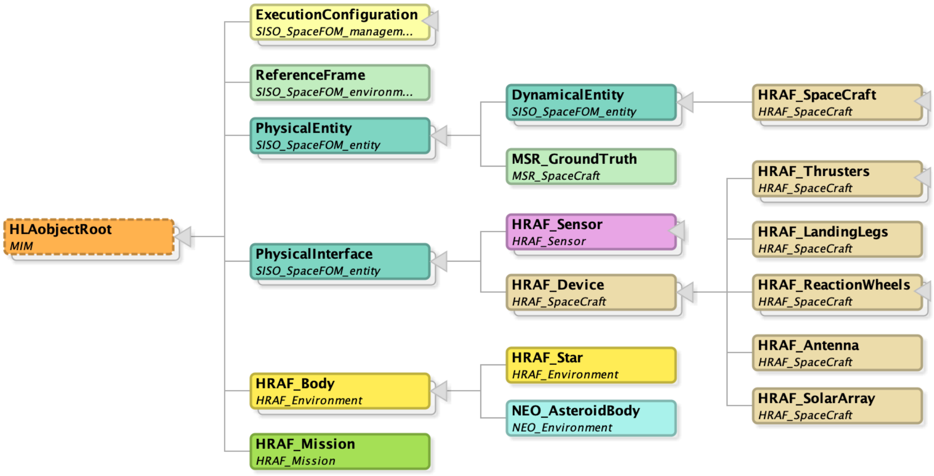


Figure 5: HRAF object class overview

Figure 5 gives an overview of key object classes in the HRAF FOM. Spacecrafts are the only dynamic entities in this FOM. There are two types of physical interfaces: sensors and general devices, like thrusters, antennas, etc. These can then be attached to an entity. Note that the mission information (objective, phase, start time, etc) is stored in an object instance which means that several parallel missions can be simulated.

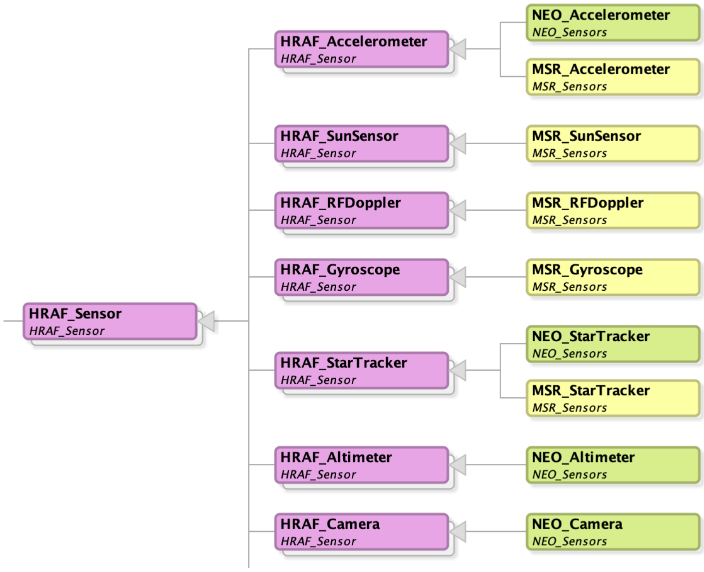


Figure 6: HRAF sensor classes

Figure 6 provides more details on the HRAF sensor classes, which are subclasses of the PhysicalInterface class in the SpaceFOM. Attributes that are specific for each scenario (MSR, NEO) are described in subclasses of the general classes.

The HRAF FOM represents the extension of the SpaceFOM to create a set of common “HRAF” Object and Interaction classes, agnostic to the scenario, obtained by evaluating the data transmission requirements for each simulator to support distribution and identifying structures common to both. Any additional data structures or extensions to existing classes, not defined by the common Data Types, Object classes and Interaction classes were introduced in scenario-specific modules.
As the utilization of Objects and Interactions is not defined by the standards, a schema for determining when to apply one or the other to a specific interface was decided. This results in the Interactions being utilized for commands and any transmission whereby it is not desirable to use any data that might be out of date.

As such, all commands published by the GNC, those being for controlling the thrusters, reaction wheels, and sensors for both scenarios were sent using Interactions, in addition to any antenna or solar array commands modelled in the simulators. This was also applied to those interfaces from the GNC dealing with the statuses of equipment. Despite not strictly commanding the devices, the MSR scenario included data for setting enabling or disabling individual elements of a cluster of devices, such as the thrusters and sensors, and these too were sent using Interactions.

Whilst the FOM was designed to follow best practices, there are examples of compromises existing in the final version. The use of existing simulators for the activity forced the FOM to be created around the simulators, rather than the architecture informing the interfaces and flow of a system developed in tandem to the federation. This resulted in extra Objects and Interactions for the sole purpose of providing feedback to components, due to the implementations of the simulators that would otherwise not be included.

The decision to use Interactions came about after comparing Interactions and shared ownership when confronted by the simulators modifying data that was owned by another federate once the components were distributed. The decision came down to the performance and complexity of the two methods. Whilst conceptually simpler, sharing objects between two federates was determined to be more expensive in both development and execution time, and as such the additional Interactions were used.

4.3 Federate Software and Interface Design

To develop the large number of federates considered in the new Federation in an efficient and modular way, a component-based design approach was adopted to promote the sharing of configurable, flexible and modular libraries between the federate executables, thus minimizing the volume of software code required to implement each Federate. Written in Visual C++ in Windows, a number of shared, highly configurable libraries (DLLs) may be linked to the Federate under development as required. Where required, unique software components specific to a given Federate are introduced to complete the functionality required. The general concept of the federate software design is shown in Figure 7.

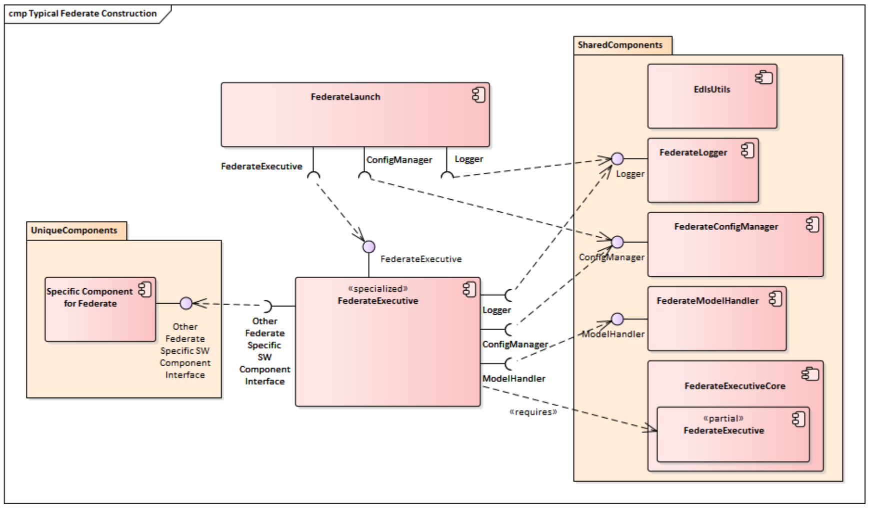


Figure 7: Federate software design concept

To interface the simulator components (models, external software, hardware) with the corresponding federate(s), a number of internal interfaces were defined which can be invoked via configuration. The preferred protocol being UDP-IP, selected to minimize latency and for its simplicity.

Largely, the internal interface strategy employed is to force the interfaced simulator components into a blocked state prior to receiving new data and therefore preserve the internal states of the model/software/hardware. Once new data is received, the simulator component may run and produces new outputs, which are received by the federate and object updates and the sending of interactions may be performed.

A synchronization mechanism also using UDP was introduced to the broken down Simulink models to ensure the federate knows when the model is ready to receive the first datagram. Where the ability to enforce a dedicated synchronization mechanism is not possible (e.g. when reusing existing avionics hardware with defined interface definition) then unique interfaces are introduced in the Federation to support this integration.

To introduce specialized software and hardware into the federation, such as Ethernet cameras and the Image Generation tool PANGU, the corresponding APIs were included and/or wrapped into the respective software libraries handling the interface to the tool, therefore maintaining the modular and abstracted paradigm employed.

5. Discussion

This section describes some challenges experienced in the HRAF3 project as well as solutions. The reason for many of these challenges is the reuse of existing models, where their original design did not take federation requirements into account.

5.1 Federating MATLAB simulations

A number of federates were provided as Matlab/Simulink models. Integrating these proved to be a time consuming and technically challenging task. Splitting the models required careful attention to signal paths and consideration for feedback logic that, if not handled appropriately, may modify the behavior of the simulator during federation execution. To enforce the required blocking logic in Simulink, UDP receiving (including synchronization at startup) and UDP transmitting blocks were developed as SFunctions.

Using blocking UDP receiving calls provided the required preservation of the model's internal state (specifically those which include integrator logic) while waiting for new data to be received. Care had to be taken, and additional preparation made, to ensure that blocking behaviors did not interfere with the native Fixed-step solver (typically ode3 or ode4) of the full simulator, which should be applied to the broken down models to ensure numerical consistency. Use of UDP interfaces with Simulink proved to lend itself well to supporting development and testing efforts, meaning that the creation of test stubs, mocks and stand-ins was very simple in scripting languages such as Python, which was useful for quick testing during development and deployment.

The nature of Simulink based simulator components (and all other software/hardware components reused in the new Federation) is such that they require all the input data to complete the interface definition before they should be run to generate new output data. This therefore enforces a strict interface definition at RTI level, and defines a fixed order of execution between the Federates in the Federation, per configuration. This principle extends to MPI initialization process, where each of the Simulink models were run for the first time. In limited cases, it was found that the order of execution during running was not compatible with initialization as data would not be available in time. To overcome this, the functionality to load "default" values via configuration file was added to inject initial conditions where required.

5.2 Micro stepping HLA Logical Time

One challenge in HLA Time Management was encountered. Consider the case where all federates use a common time step (also known as frame-based simulation). In each time step, each federate calculates and sends data that is provided to other federates as a starting point for the next time step. The RTI guarantees that all updates for a given time step is provided to each federate before it starts calculating data for the following time step, or to be more exact, before the RTI grants it to that time. The problem in this federation is that the calculation within a given time step is split between several federates, that need to exchange data in a controlled way. In the HLA community this is known as the "zero lookahead" case.

The solution was to define several distinct micro steps for the HLA logical time, for each given scenario time step. Consider the case with a required execution sequence of first federate A, then federate B then federate C. For the scenario time 23 seconds, which in SpaceFOM is expressed as HLA Logical time 23 000 000 (microseconds), federate A would execute at 23 000 000, federate B would execute at 23 000 001 and federate C at 23 000 002. In their time calculations they would disregard the micro step and consider the time to be 23 000 000. This approach gives all the benefits of HLA Time Management, while enabling federates to exchange data within the same scenario time step.

5.3 Hardware interfacing and timing for a Robotic Arm HIL Facility

A key design driver when considering the integration of Robotic Arm HIL facilities into the new Federation was to define an interface and integration strategy that would be agnostic to the specific facility, as two separate facilities must be allowed to run simulations of different configurations of the Federation. Each facility (Platform-Art® and GRALS) being different in architecture, interface, and requirements meant that at a design level, it was important to consider the interface at as high a level as possible, abstracting and treating the functionalities essential to operate such a facility as a black-box. Only a pose object for the arm in the appropriate reference frame would be given to the facility via the corresponding federate. Therefore, only the relevant transformations between the reference frame of the received pose from the simulator and those in the facility must be accounted for in the robot controller (under responsibility of the facility to implement).

Platform-Art® uses a proxy software between the controller and federate to de-couple the federate and robot controller. The proxy simplifies the functional and performance requirements of the federate by handling interpolation of the received pose (computed once per 4s of Simulation time) and shall continue commanding the robot arm to the same pose at a constant period of 12ms, even while awaiting a new pose.

GRALS requires use of a Simulink model for interfacing, transformation, command and low-level control of their robot arm, developed by the GNC section at ESA/ESTEC. This model was adapted to include a non-blocking UDP interface to maintain the required constant communication with the robotic arm at period of 4ms, but also receive new poses at a frequency of ~1Hz from the federate. This presented a challenge to define asynchronous logic to process new poses when received, and respond once position and velocity constraints were met (known from telemetry) all from within the Simulink model.

Both implementations shall send an interaction to the RTI once the commanded pose has been reached, and the robot is stationary, signifying it is now possible to capture an image.

6. Conclusions

HRAF3 is the first major application of the SISO SpaceFOM to a European project. A baseline federation for the Harwell Robotic and Autonomy Facility has been successfully developed. It can be reused and forms a starting point for future experiments. The SpaceFOM was helpful in this integration.

Some challenges experienced include integration of reused and complex Matlab/Simulink models in federations and meeting particular timing requirements of existing hardware.

The project will provide feedback to SISO for improving future versions of the SpaceFOM standard, for example around FOM development and Time Management.

Acknowledgment

This work has been funded by the European Space Agency, Contract No 4000124144/18/NL/PA between ESA-ECSAT and GMV Innovating Solutions Ltd.

References

[1] B. Möller, E. Z. Crues, D. E. Dexter, A. Garro, A. Skuratovskiy, A. Vankov. “A First Look at the Upcoming SISO Space Reference FOM”. Proceedings of the SISO 2016 Simulation Innovation Workshop (SIW), Orlando, Florida, USA, September 11-16, 2016.

[2] SISO-STD-018-2020 Space Reference Federation Object Model (SpaceFOM), October 2019, www.sisostds.org

[3] SISO - Simulation Interoperability Standards Organization, www.sisostds.org

[4] IEEE - Institute of Electrical and Electronics Engineers, www.ieee.org

[5] SISO-ADM-003-2011 “Balloted Products Development and Support Process”, November 2011, www.siostds.org

[6] IEEE: "IEEE 1278.1-2012 - Standard for Distributed Interactive Simulation - Application protocols", www.ieee-org, August 2012

[7] IEEE 1516-2010. High Level Architecture (HLA), [online], August 2010, available at www.ieee.org,

[8] Ward, R., et. al. “HRAF PILOT- Prototype Facility for Verification & Validation of Robotics & Autonomy”, DASIA 2014 - DAta Systems In Aerospace, Proceedings of the conference held 3-5 June 2014 in Warsaw, Poland. Edited by L. Ouwehand. ESA SP Vol. 725. ISBN: 978-92-9221-289-6, id.58

[9] Allouis, E., et al. "A FACILITY FOR THE VERIFICATION & VALIDATION OF ROBOTICS & AUTONOMY FOR PLANETARY EXPLORATION." Advanced Space Technologies in Robotics and Automation (ASTRA), Noordwijk, The Netherlands (2013).

[10] Kostavelis, I., et. al. (2014), SPARTAN: Developing a Vision System for Future Autonomous Space Exploration Robots. J. Field Robotics, 31: 107-140. https://doi.org/10.1002/rob.21484

[11] https://www.gmv.com/en/Products/platform/

[12] Strippoli, L., et. al. “MSR RENDEZVOUS AND CAPTURE PHASE: THE GNC SUBSYSTEM SOLUTION AND A SNAPSHOT OVER THE IMAGE PROCESSING ALGORITHMS AND NARROW ANGLE CAMERA ELEGANT BREADBOARD”, 70th International Astronautical Congress, Washington, D.C., USA (2019)
[13] Muirhead, B.K., et. al. “Mars Sample Return Campaign Concept Status”, Acta Astronautica, Volume 176, 2020, Pages 131-138, ISSN 0094-5765, https://doi.org/10.1016/j.actaastro.2020.06.026.
[14] Suatoni, M., et. al. “DEVELOPMENT AND VALIDATION OF GNC FUNCTIONAL MODES FOR ASTEROIDS SAMPLE RETURN MISSION IN THE FRAME OF THE EUROPEAN COMMISSION H2020 NEOSHIELD-2 PROJECT”, 68th International Astronautical Congress, Adelaide, Australia (2017)
[15] Michel, P., et. al. (2012). MarcoPolo-R: Near Earth Asteroid Sample Return Mission candidate as ESA-M3 class mission. Proceedings of the International Astronomical Union, 10(H16), 163-163. doi:10.1017/S1743921314005183
[16] PANGU: https://www.star-dundee.com/products/pangu-planet-and-asteroid-natural-scene-generation-utility/#product_features

Author Biographies

BJÖRN MÖLLER is the President and co-founder of Pitch Technologies. He has more than twenty-five years of experience in high-tech R&D companies, with an international profile 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 is currently serving as the chairman of the SISO RPR FOM Product Development group, chairman of the SISO Space FOM Product Development Group. and the vice chairman of the SISO HLA Evolved Product Development Group.

TOM GRAY is a Senior Software Developer at Pitch with more than ten years of experience in design, development, integration and acceptance of simulation systems for UK and international customers. Tom is currently an active member of SISO’s RPR FOM Product Development Group. He studied B.Eng. Electrical and Electronics Engineering at the University of Surrey, England.

STEVEN KAY is a Robotics Engineer at GMVNSL and is the System Engineer and Technical Lead for the current HRAF Pilot 3 activity. Steven previously contributed to the HRAF Pilot 2 activity while working for STFC RAL Space. He holds an M.Eng. in Robotics & Cybertronics from Heriot-Watt University, Scotland and is currently studying for a PhD at the Surrey Space Centre, University of Surrey, England in advanced autonomy concepts for planetary robotics.

ARON KISDI is currently the Robotics Group Leader at GMVNSL and project manager of the HRAF project. He is also managing commercial robotics projects in non-space application areas of infrastructure monitoring, mining and agriculture. Aron is also heavily involved in RAS strategy and policy with KTN and UK RAS network and has contributed to the white papers on space robotics. Aron has been responsible for logistics on field trials SEEKER and SAFER and was a systems engineer in HRAF Pilot 2 rover validation project. He has extensive experience in how to validate robotic systems and this is key. He holds an MEng in Space Systems Engineering from University of Southampton and a Chartered engineer with Royal Aeronautical Society.

KARL BUCKLEY is a Robotics Engineer at GMVNSL, and was a Software Developer for the HRAF activity. Karl has contributed to the development and integration of robotic platforms for space and terrestrial applications, including HRAF Pilot 2 and the H2020 activity ERGO. He studied MEng Robotics at Plymouth University.

JUAN DELFA is a robotics and autonomy engineer at the Human and Robotic Exploration (HRE) directorate in ESA. He has been working for several years in the field within ESA and European industry, leading the design of several autonomous systems. At present, he shares his time between technology development activities and the role of Surface Operations System Engineer for the Sample Fetch Rover in the Mars Sample Return Campaign. He holds a M. Eng in Computer Science by the Technical University of Ciudad Real (Spain) and a PhD on Automated Planning by the Technical University of Darmstadt (Germany).