Simulation Interoperability Standards Organization

"Simulation Interoperability & Reuse through Standards"

SISO Space Reference FOM 101

Edwin "Zack" Crues

NASA JSC

SpaceFOM Editor

Björn Möller

Pitch

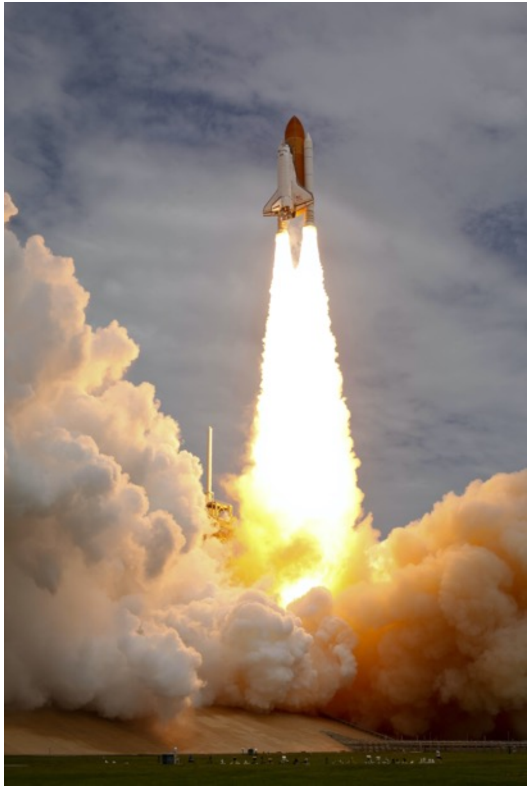
SpaceFOM Chair

Outline

- Why a Space Reference FOM Standard?
- Early Space HLA Federations
- Quick HLA Recap
- Reference Frames and Standard Reference Frames
- Time and Time Management
- Execution Control
- Physical Entities
- Implementing and Extending the Space Reference FOM
- A Practical Example: Simulation Exploration Experience
- Summary

Why a SISO Space Reference FOM – 1

- Spaceflight is difficult, dangerous and expensive; human spaceflight even more so
- In order to mitigate some of the danger and expense, professionals in the space domain have relied, and continue to rely, on computer simulation
- Simulation is used at every level including concept, design, analysis, production, testing, training and ultimately flight
- Distributed simulation provides a base technology for segmenting these complex space systems into smaller, and usually simpler, component systems or subsystems



Why a SISO Space Reference FOM – 2

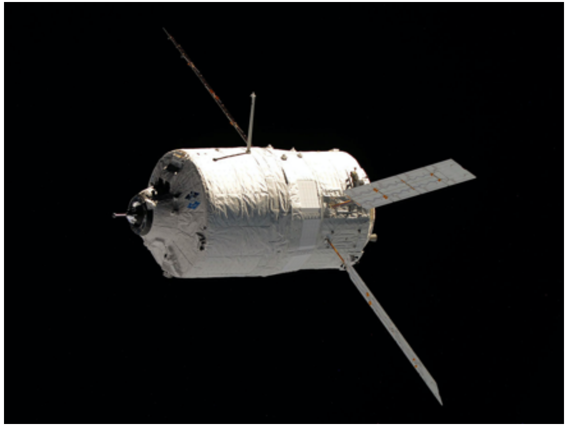
- **Integrating simulations is costly**
  - Need to minimize the integration effort for new and reused systems
- **Open standards offer a more efficient way to combine and reuse systems and tools in new configurations**
  - They also offer a neutral ground that is easier to accept for many participants than proprietary interfaces
  - Open standard can also capture best-practices and help communicate them to new developers
- **HLA is not enough to ensure interoperability, a FOM is also needed; thus, the SISO Space Reference FOM (SpaceFOM)**
- **The SpaceFOM should make collaboration politically, contractually and technically easier**

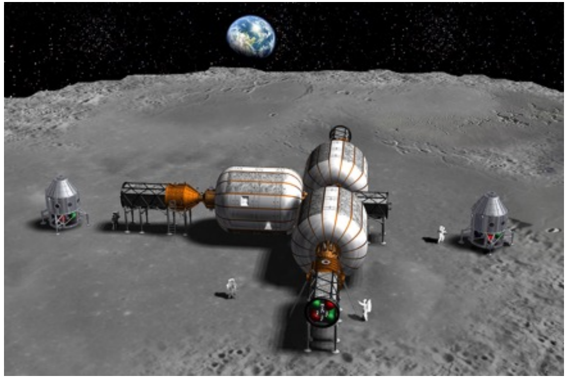
Early Space HLA Federations – 1

- **NASA Space Station Training Facility (SSTF) [circa 2000]**
  - Authorization of space crafts visiting the ISS includes simulation
  - SSTF and JAXA (Japan) HII-A Transfer Vehicle (HTV) federation in support of flight controller and crew training
- **NASA Integrated Mission Simulation (IMSim) in support of the Constellation program [circa 2010]**
  - Evolved into NASA Exploration Systems Simulations (NExSyS)

Early Space HLA Federations – 2

- ESA and Russian Mission Control Center federation for Automated Transfer Vehicle (ATV) to the Russian segment of ISS
- ESA/ESTEC and NASA JSC rendezvous and docking
- ESTEC and Tsukuba (Japan) space center rendezvous
- EDISON European R&D consortium (man-in-the-loop, hardware-in-the-loop)
- SISO/SCS Simulation Exploration Experience (SEE, ex Smackdown) – "Lunar City"





Development of the SpaceFOM

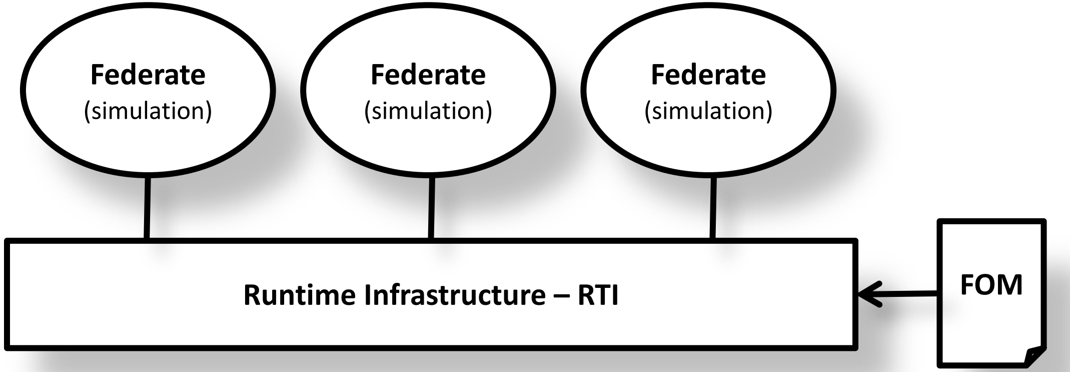
- **SISO provides a process called the Balloted Products Development and Support Process (BPDSP)**
  - Product Nomination
  - Product Development
  - Product Balloting
  - Product Approval
- **The Product Nomination was approved May 2015**
- **The Product Development Group (PDG) developed the standard**
  - Members from government agencies, industry and academia
- **The standard has recently been balloted and approved**
- **It was published in January 2020 and is now publicly available**

SISO Space Reference FOM 101

A Quick Recap of the High Level Architecture

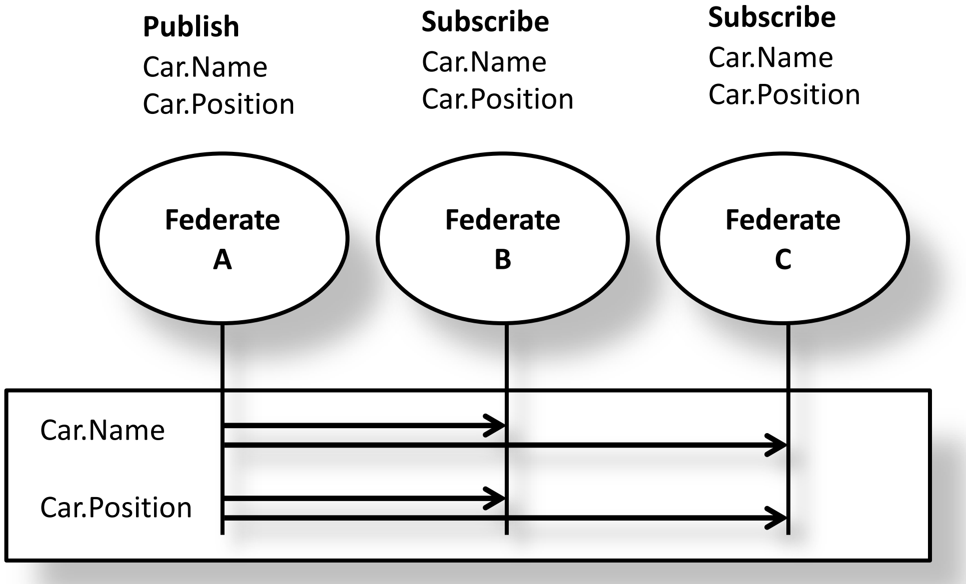
Selected features

HLA Terminology



- Each system is called a federate
- Federates connect to a Runtime Infrastructure that provides services
- Together they form a Federation
- Data is exchanged based on a Federation Object Model (FOM)
- A session is called a Federation Execution

Publishing and Subscribing Data



- Federates publish and subscribe to data according to the FOM
- The RTI makes sure that data is correctly routed between publishers and subscribers

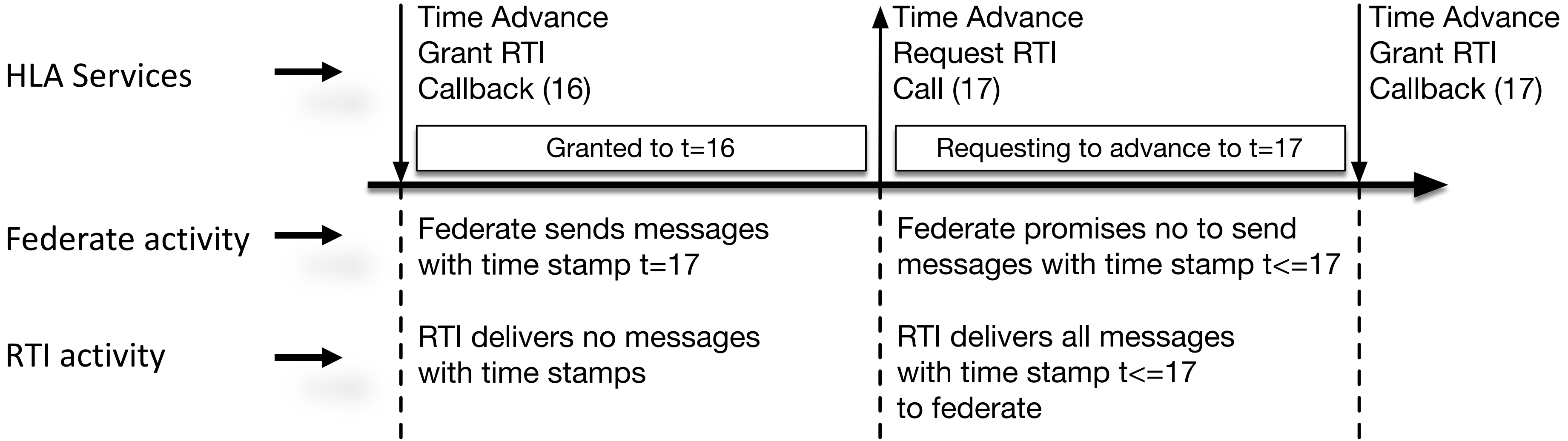
Synchronization Points

- **Synchronization Points are used to make sure that the entire federation has achieved a certain state, before it can move on**
  - “Are you ready?”
- **Sample use case: load a scenario**
- **Typical sequence:**
  - One federate registers a synch point, which is then announced to all federates in the federation by the RTI
  - Each federate performs necessary actions (like loading a scenario)
  - Each federate tells the RTI that it has achieved the synch point and waits
  - When all federates have achieved the synch point, the RTI announces that the federation is synchronized to all federates. Execution can now continue.

HLA Time Management

- **HLA Time Management enables federations to run as-fast-as-possible and facilitates faster/slower than real-time execution , while guaranteeing correct and repeatable results. Key mechanisms:**
  - Reliable, time-stamped ordered delivery of messages and
  - Coordinated time advance, where federates are guaranteed not to get messages in their past logical (scenario) time.
- **Simplified description:**
  - To move forward in time, each federate requests to advance to a particular time
  - The RTI grants the right to advance to that time, once all messages with a time-stamp up to that time has been delivered to the federate

Implementation of Time Management Services in a Federate

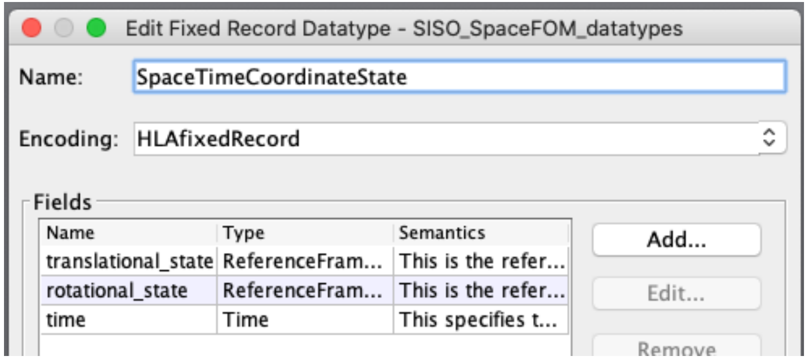


SISO Space Reference FOM 101

Reference Frames and Standard Reference Frames

Introduction to Spacetime Coordinates

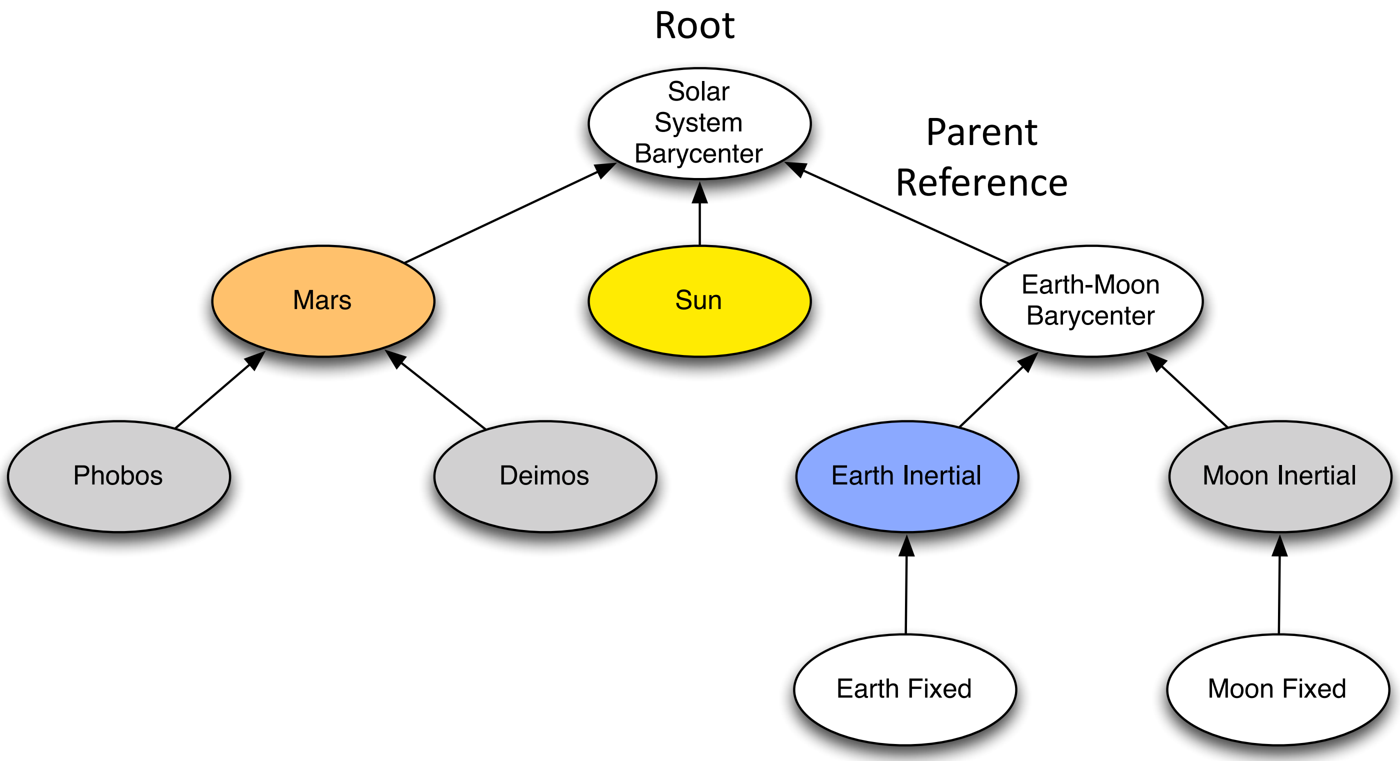
- **We need to express where and when a physical entity exists.**
- **This is done with respect to some time scale and some referent coordinate system.**
- **This combination of time and coordinate system is referred to as a spacetime coordinate.**
- **For an entity in the SpaceFOM, a parent reference frame and a spacetime coordinate state is used**
  - Translational State (Position)
  - Rotation State (Attitude)
  - Time



Reference Frames

- **Reference frames are the basis for expressing the state of a physical entity.**
- **A reference frame may be more or less computationally convenient for different purposes**
  - Performing a mining operation on the Moon using an Earth centric inertial reference frame would be inconvenient
- **Reference Frames are provided as HLA object instances**
  - Several reference frames can be used
  - A federation can use the reference frames that meets its needs
  - Reference frames are related to each other so that a spacetime state in one reference frame can be represented in another reference frame

Sample Reference Frame Tree

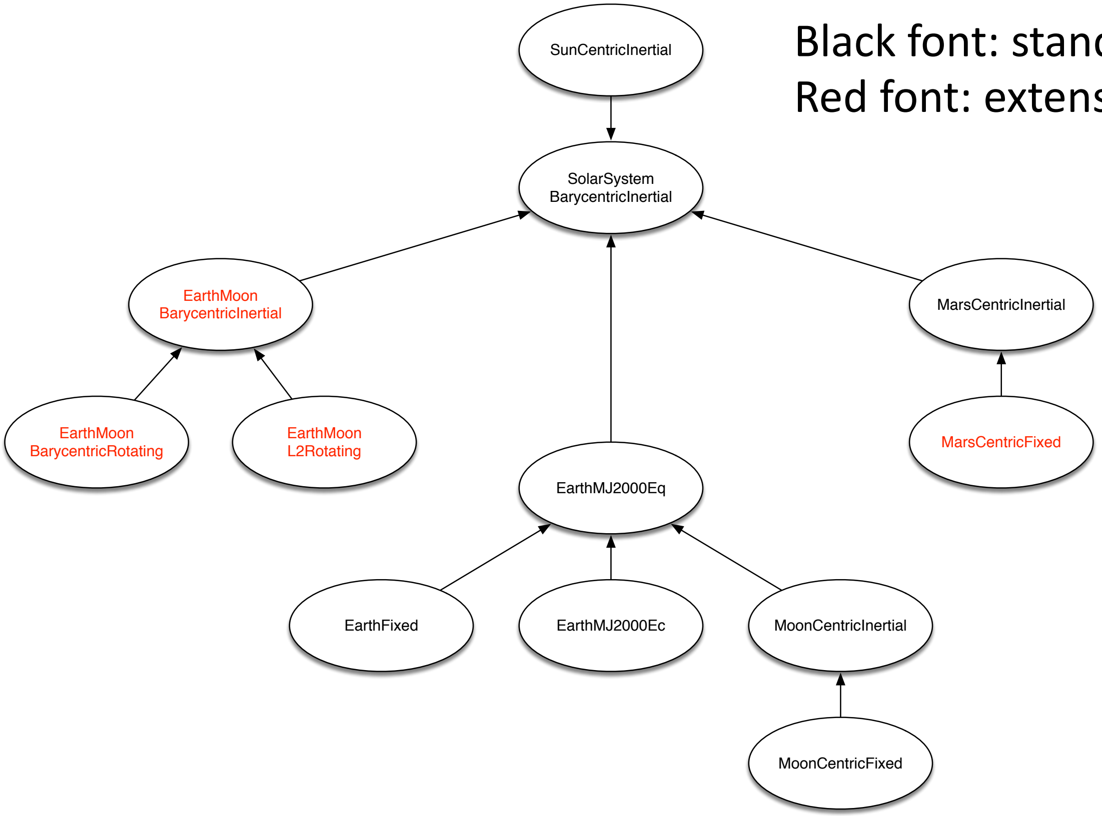


Reference Naming Conventions

- **The SpaceFOM provides a convention for naming reference frames based on <u>frame origin</u> and <u>frame orientation</u>**
  - Example: SunCentricInertial
- **Frame Origin (simplified)**
  - Celestial body, Lagrange point, Human Made Body or User defined
  - Celestial bodies can be Solar System Bodies (Sun, Planet, Dwarf planet, Natural Satellite or other body) or Extra Solar System Bodies
  - Celestial body name are based on International Astronomical Union DB
- **Frame Orientation (simplified)**
  - Fixed, Inertial, Rotational, Rotating
  - Axes types may be specified

Some Standard Reference Frames and Extensions

Black font: standard reference frames
Red font: extensions



SISO Space Reference FOM 101

Time and Time Management

Time – Basics

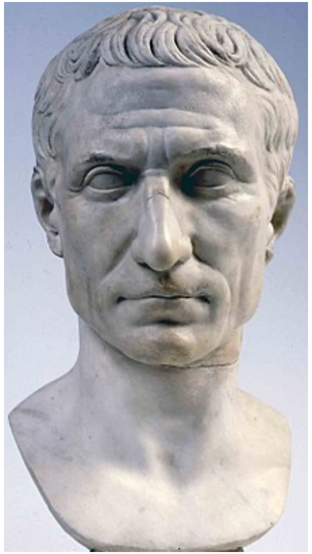
- **The SpaceFOM uses the Newtonian approximation to propagating time**
  - Relativistic time dilatation is on the order of microseconds per day using current space transportation technology. This is insignificant for shorter scenarios.
- **Some important time lines**
  - Physical time
  - Computer clock time
  - Simulation elapsed time
  - Simulation scenario time
  - HLA logical time
  - Federation scenario time

Time – Time Line Definitions

- **Physical Time**
  - The time that ”we” experience. The non-spatial dimension associated with our spacetime continuum in which events are ordered in irreversible succession from the past to the present to the future.
- **Computer Clock Time**
  - The count of oscillation in a ”clock” on the computer that is translated into a representation of the current Physical Time.
- **Simulation Elapsed Time**
  - The time measure associated with an individual simulation starting at zero and advancing monotonically in quantifiable steps.
- **Simulation Scenario Time**
  - A model within a simulation that associates the Simulation Elapsed Time with a representation of the problem’s Physical Time.
- **HLA logical time**
  - The time line used by HLA to order messages, regulate execution time advance (TAR & TAG), and enable deterministic behavior in distributed simulation.
- **Federation scenario time**
  - A conceptual time associated with the physical systems being modeled in the participating federates in the federation execution.

Time Representation and Interpretation

- **Julian Dates**
  - A Julian day number is the number of days that have elapsed since the Julian Epoch, i.e. noon of January 1, 4713 BC in the Julian calendar
  - Truncated Julian Date (TJD) starts at 00:00:00 May 24, 1968
- **A SpaceFOM federation shall use TJD to specify the epoch (starting point) of a scenario**
- **The Terrestrial Time (TT) time scale shall be used**
  - Uses the SI second unit. No leap seconds (unlike UTC).
- **Time shall be represented using a 64 bit integer that shall be interpreted as microseconds.**



Julius Caesar

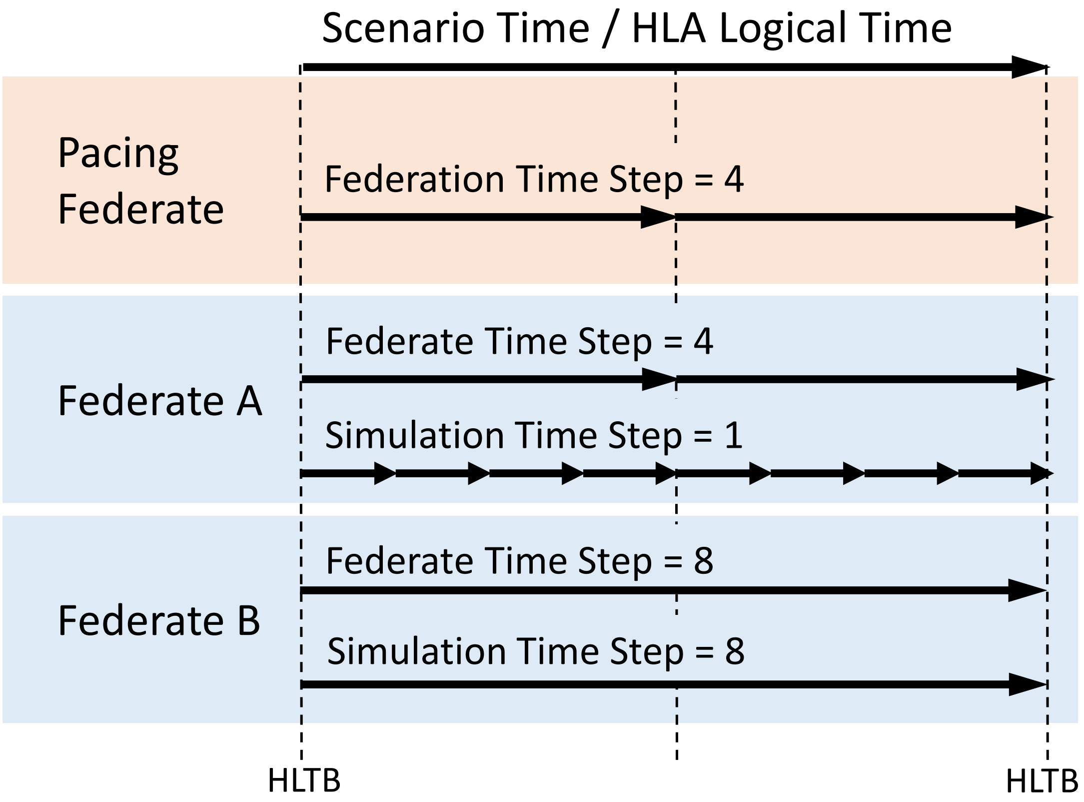
Types of Time Management Supported

- **No Pacing (as-fast-as-possible)**
- **Pacing (regulated by a clock)**
  - Scaled Pacing
  - Real-time Pacing
    - *Strict/Conservative Real-time (no frame overruns)*
    - *Elastic Real-time (catch-up on overruns) with Limited Overruns*
    - *Elastic Real-time (catch-up on overruns) with Unlimited Overruns*
- **Fixed time step is required**
- **As-fast-as-possible is useful for Monte Carlo simulation**
- **Real-time is useful for “man in the loop” or “hardware in the loop”**

Time Steps in a Federation

- **Simulation Time Step**
  - Native time step used inside a simulation
  - Complete simulation states are only available at simulation time steps
- **Federate Time Step**
  - Time step used by a federate for Time Advance Request
- **Federation Time Step**
  - Federate time step of the Pacing federate
- **Least Common Time Step**
  - The smallest time step that all federates can support
- **HLA Logical Time Boundary (HLTB)**
  - HLA logical time (position) that is common to all Time Managed federates
  - Used for late joiners and going to freeze

Putting Time Steps Together

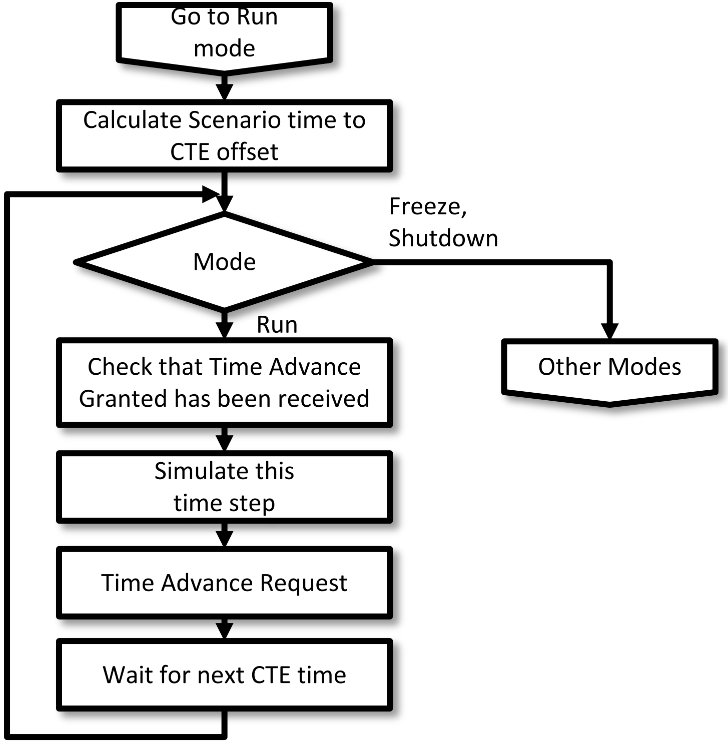


LCTS = 8

Central Timing Equipment (Hard Real-Time)

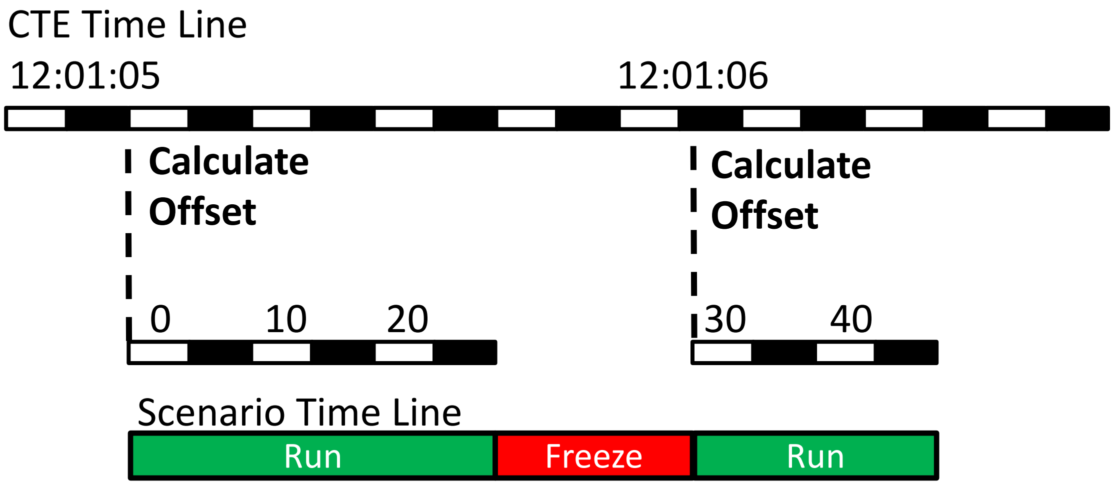
- An HLA federation can accommodate both simulations running in soft real-time and simulators that use central timing equipment (CTE) (e.g., a GPS timing board) for hard real-time synchronization
- The HLA federation is capable of going to freeze, and later resume
- The simulations that synchronize using the CTE, must also be able to handle these mode transitions

Main Loop with CTE (Hard Realtime)



Run and Freeze with CTE

- **The HLA Logical time line and the CTE physical time line are connected during Run mode**
  - An offset is calculated when entering Run mode
- **When entering Freeze mode they are disconnected**



SISO Space Reference FOM 101

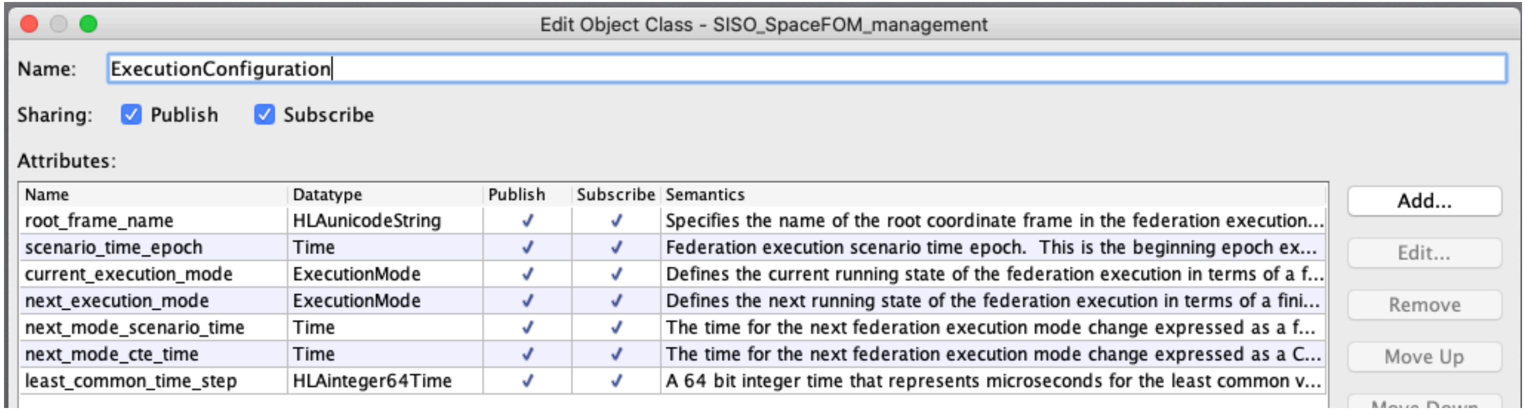
Execution Control

Federate Roles

- **The SpaceFOM defines three required federate roles that must be present in a federation**
- **Master Federate**
  - Manages the startup, initialization and execution control which includes maintaining the Execution Control Object and registering synchronization points.
- **Pacer Federate**
  - Manages the advance of the HLA Logical Time in the federation
- **Root Reference Publisher Federate**
  - Publishes the root reference frame.
- **A particular federate may implement more than one role**

The Execution Control Object (ExCO)

- **A singleton object instance with shared information for Execution Control**
  - Root reference frame name
  - Federation execution scenario time epoch
  - Mode and time information



Edit Object Class - SISO_SpaceFOM_management

Name: ExecutionConfiguration

Sharing: ☑ Publish ☑ Subscribe

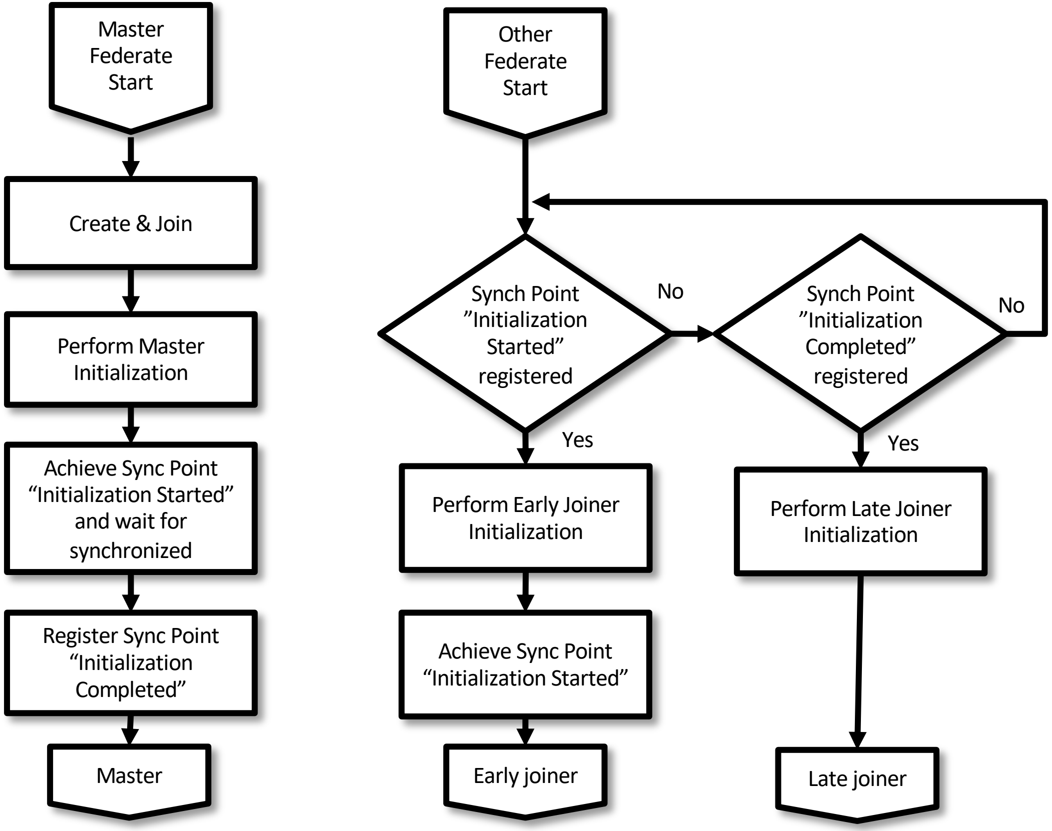
Attributes:

Name	Datatype	Publish	Subscribe	Semantics
root_frame_name	HLAunicodeString	✓	✓	Specifies the name of the root coordinate frame in the federation execution...
scenario_time_epoch	Time	✓	✓	Federation execution scenario time epoch. This is the beginning epoch ex...
current_execution_mode	ExecutionMode	✓	✓	Defines the current running state of the federation execution in terms of a f...
next_execution_mode	ExecutionMode	✓	✓	Defines the next running state of the federation execution in terms of a fini...
next_mode_scenario_time	Time	✓	✓	The time for the next federation execution mode change expressed as a f...
next_mode_cte_time	Time	✓	✓	The time for the next federation execution mode change expressed as a C...
least_common_time_step	HLAinteger64Time	✓	✓	A 64 bit integer time that represents microseconds for the least common v...

Add...
Edit...
Remove
Move Up

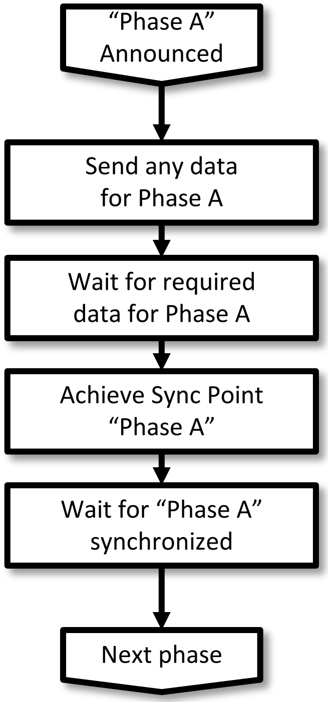
Early and Late Joiners

- The Master checks that all required federates have joined
- It then triggers Early Joiner initialization
- Federates joining after that will be put on hold and perform a late join.



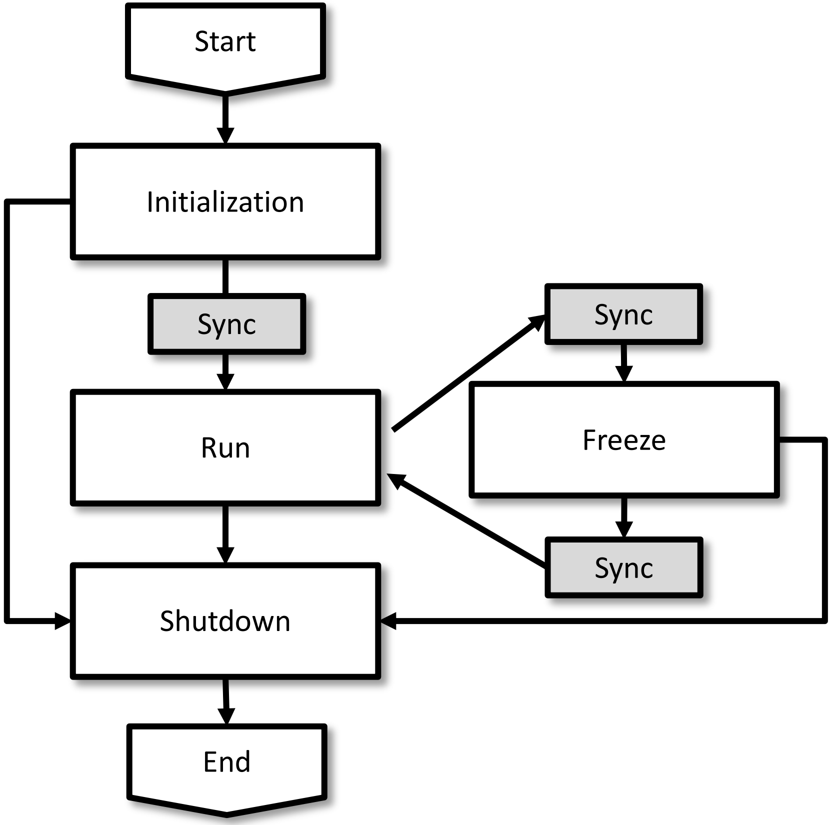
Multi Phase Initialization

- Initialization of a federation may often need to be performed in several discrete steps
- Lift-off simulation example:
  1. Initialize launch pad objects
  2. Initialize first stage objects
  3. Initialize second stage objects
- This is supported in the SpaceFOM using synch points
- The SpaceFOM allows for a configurable set of phases



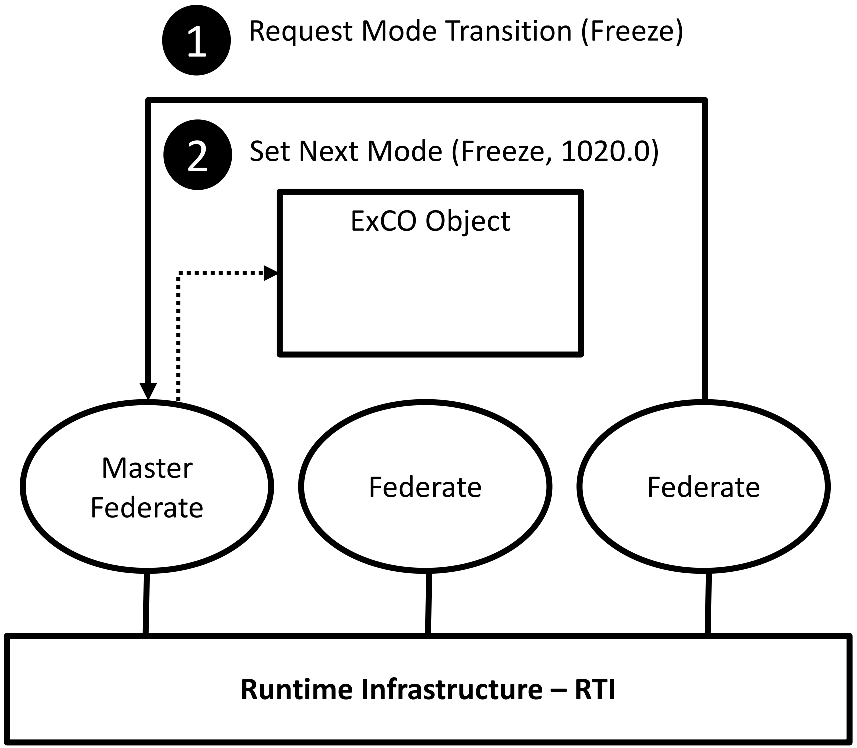
Central Execution Control Flowchart (simplified)

- **Four phases exist:**
  - Initialization
  - Run
  - Freeze
  - Shutdown
- **Going to Run, Freeze and Shutdown is controlled by the Master, using the ExCO**
- **Synch Points are used when entering new phase, except Shutdown**



Requesting a Mode Transition (simplified)

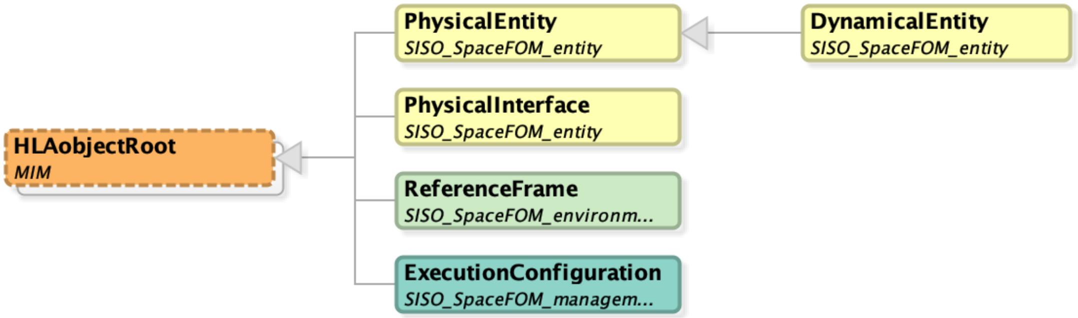
- Any federate can request a mode transition using an interaction
- The Master decides if and when a mode transition shall take place
- Can only transition to Freeze or Run on an HLA Logical Time Boundary (HLTB)



SISO Space Reference FOM 101

Physical Entities

SpaceFOM classes for Entities



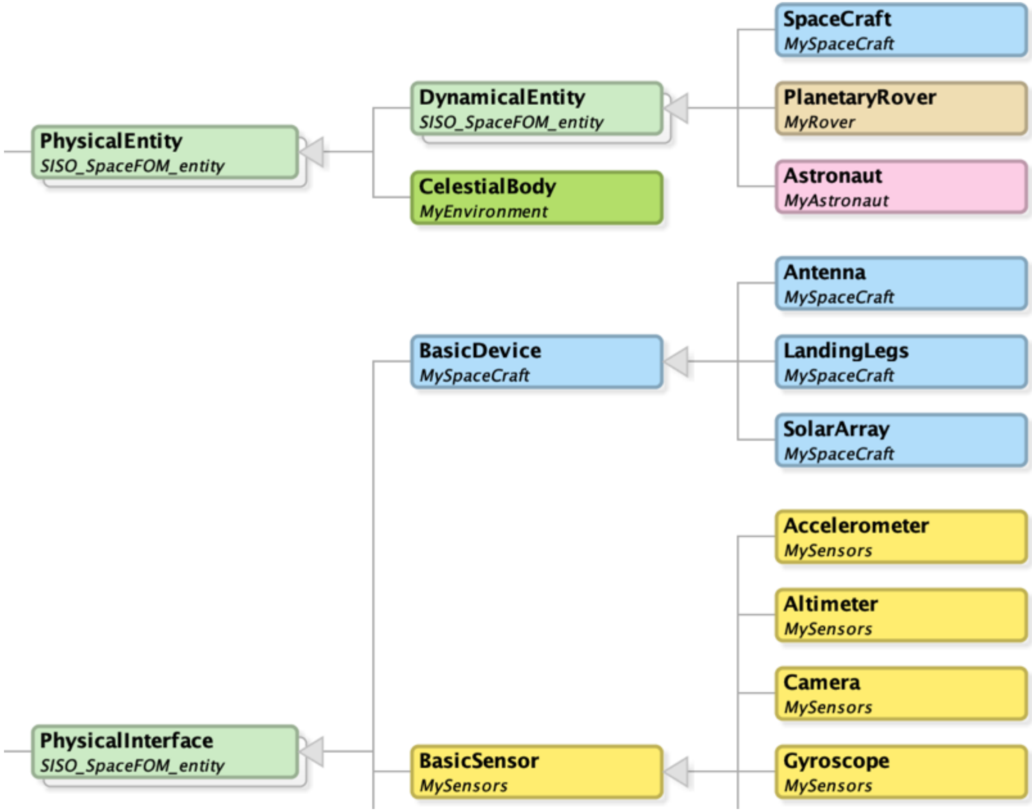
- **Three key classes that can be used or subclassed:**
  - PhysicalEntity – any physical entity
  - DynamicalEntity – a physical entity with dynamic forces and/or torques, such as a space craft.
  - PhysicalInterface – interface of object mounted on an entity, such as a docking port
- **It is also possible to create subclasses of the HLAobjectRoot**

A Sample Dynamical Entity Instance

Attribute	Datatype	Sample Value
name	HLAunicodeString	“DemoRover1945”
type	HLAunicodeString	“Lunar Rover”
status	HLAunicodeString	“Parked”
parent_reference_frame	HLAunicodeString	“MoonCentricInertial”
state	SpaceTimeCoordinateState	(((3, 4, 5), (0, 0, 0)), ((Q*), (0, 0, 0)), 25343225)
acceleration	AccelerationVector	(0, 0, 0)
rotational_acceleration	AngularAccelerationVector	(0, 0, 0)
center_of_mass	PositionVector	(0, 0, 0)
body_wrt_structure	AttitudeQuaternion	IDENTITY_QUATERNION

Q* = Quaternion not expanded here

Sample Extended Object Class Tree (partial)



SISO Space Reference FOM 101

Implementing and Extending the Space Reference FOM

Common Project Considerations

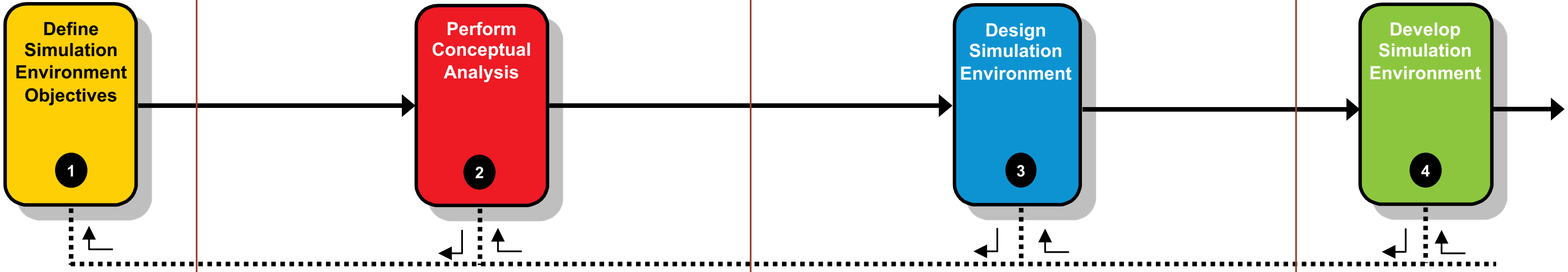
- Purpose of the simulation, time, budget, other project constraints
- Scenarios
- Conceptual model – entities, events and processes
- Object instances, naming, states
- Datasets, databases, maps, terrain, …
- Reference frames
- Time: epoch, resolution/time step, hard/soft real-time, scaling
- Federates – existing or to be developed
- Allocation of responsibility to federates
- And more…

FESFA and FCD

- **The SpaceFOM is a baseline Federation Agreement to build upon**
- **The Federation Execution Specific Federation Agreement (FESFA) provides additional agreements for your particular federation**
  - Entities, reference frames used, time step used, etc
  - A FESFA template is available in an appendix of the SpaceFOM
- **The Federate Compliance Declaration (FCD) describes capabilities of candidate federates. Useful for assessing suitability.**
  - Entities, reference frames supported, time steps supported, etc
  - An FCD template is available in an appendix of the SpaceFOM

DSEEP (IEEE 1730)

Distributed Simulation Engineering and Execution Process



Objectives → **Scenarios** → **Conceptual Model** → **Allocate Responsibilities to Federates** → **Develop Federation Agreement** → **Develop/ Reuse Federates**

Objectives

Sponsor:
"Develop a high-fidelity engineering federation, …"

Scenarios

"Space Vehicle enters the Mars atmosphere and then…"

Conceptual Model

- Space Vehicle
- Planet
- Atmosphere
- Astronaut

Allocate Responsibilities to Federates

Federate A:
- Solar System
- Atmospheres

Federate B:
- Space Vehicles

Check out FCDs

Develop Federation Agreement

- Produce Federation Agreement
- Fill in FESFA
- Extend Space Reference FOM

Develop/ Reuse Federates

- Adapt existing federate A
- Develop new federate B

Also check out Federation Engineering Agreements Template (SISO-STD-012)

FESFA and FCD contents

FESFA

1. Identification
2. Federation Composition
3. Time Management
4. Reference Frames
5. Object Management
6. Initialization
7. Additional Technical Information

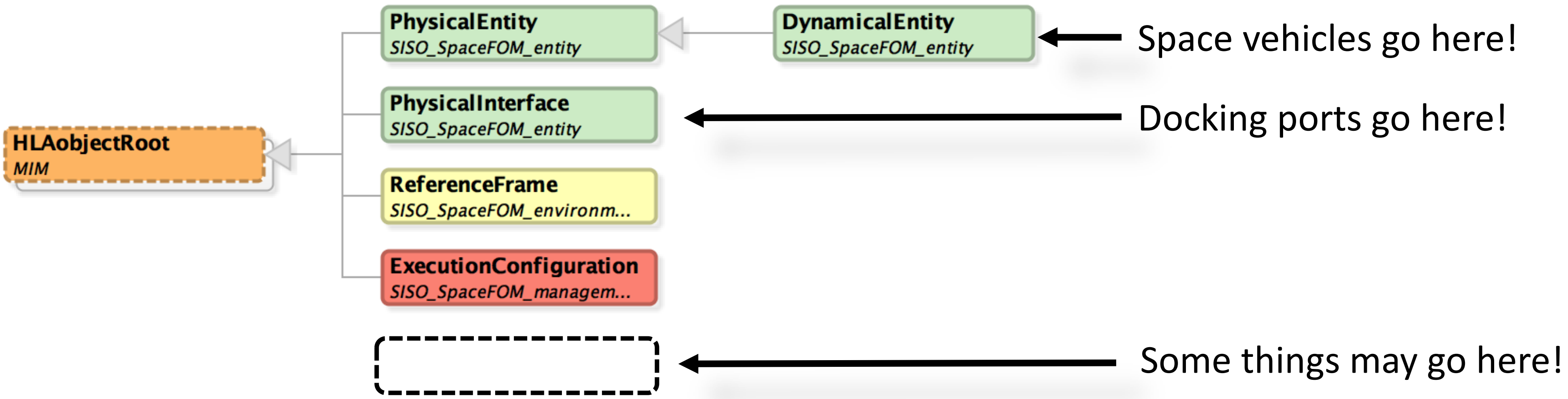
FCD

1. Identification
2. Roles supported
3. Time Management support
4. Reference Frames supported
5. Object Management
6. Initialization
7. Additional Technical Information
8. Compliance statement

More about the FESFA and FCD

- **Reference Frames**
  - FESFA: Root Reference Frame, Additional reference frames
  - FCD: Root Reference Frames it can publish. Other published or required reference frames
- **Object Management**
  - FESFA: FOM modules, Key object instances. Naming convention. Type and Status tags.
  - FCD: FOM modules, publishes or required object instances, Naming convention, Type and Status tags
- **Initialization**
  - FESFA: Use of Multi-phase Initialization. specification document
  - FCD: Use of Multi-phase Initialization. specification document
- **Other**
  - FESFA: Non-standard RTI switches. Common data and databases, etc
  - FCD: Common data and data bases. etc

Extending the FOM – Sample



Tools for SpaceFOM federations

Tool	Description	Developer
Trick	Simulation environment, simulation executive, logging	NASA
TrickHLA	Middleware between Trick and HLA	NASA
HLA Development Kit	Framework for HLA federate development	University of Calabria
SEE HLA Development Kit	Specialized version with SpaceFOM features	University of Calabria
Java Space Dynamics Library	Space dynamics library	University of Calabria
Pitch Visual OMT	FOM development	Pitch
Pitch Developer Studio	Middleware generator (C++, Java, C#)	Pitch
Pitch Recorder	Record, play back, inspect, export data	Pitch
Space Master	Generic Master and Pacer for SpaceFOM federations	Pitch
Space Monitor	Generic monitor of entities and reference frames	Pitch

SISO Space Reference FOM 101

A Practical Example: Simulation Exploration Experience

Simulation Exloration Experience

- **Provides international inter-university teams with hands on M&S experience with**
  - Standards & interoperability
  - Distributed HLA and SpaceFOM simulation testing and execution
  - Participation among far-flung international teams
  - Government and industry tutors and mentors

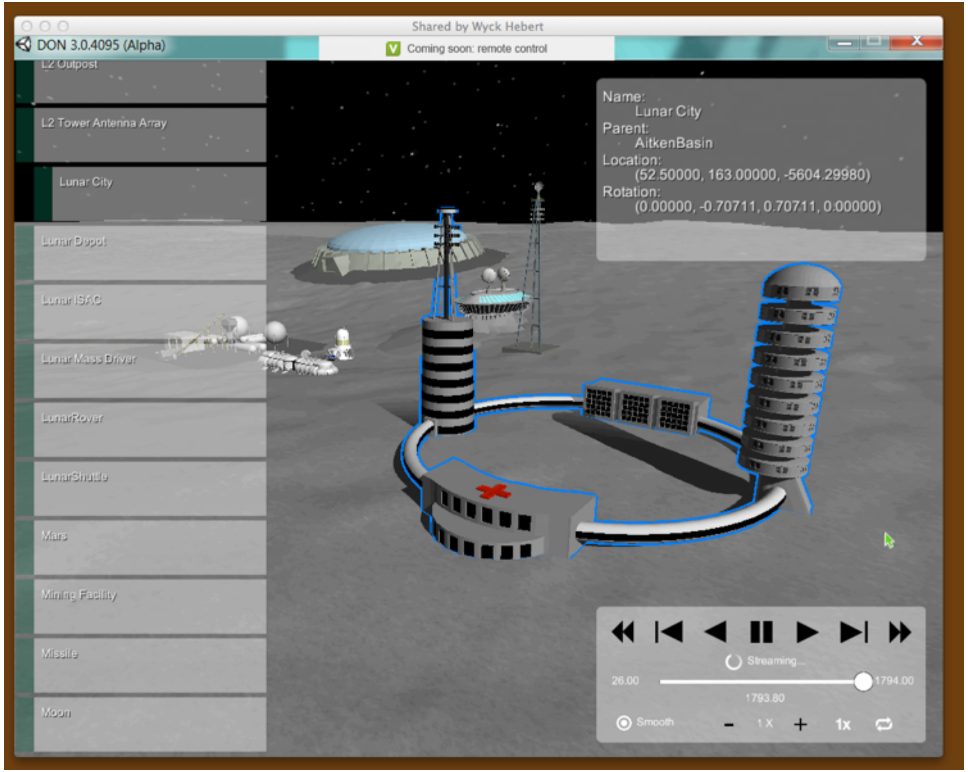
> *“As an experienced M&S educator, SEE allows me to give students practical and educational experience in a very complex area using a uniquely inspiring approach. There is nothing else out there that comes anywhere close to SEE.”*

Simon Taylor, Ph.D., faculty Advisor of the Brunel University team

SEE Participants (2019)

Some SEE Vignetttes

- Operations in Aitken Basin on the moon
- Scanning for mining resources
- Constructing a waste management facility
- Astronaut health monitoring
- Operating an asteroid protection systems
- Launching a laser thrust space craft



Additional SpaceFOM Reading

- SISO-STD-018-2020 Standard for Space Reference Federation Object Model
- “A First Look at the Upcoming SISO Space Reference FOM” (SISO 2016-SIW-017)
- “Design and Principles Enabling the Space Reference FOM” (SISO 2018-SIW-038)
- “Developing Space FOM Federation Agreements” (SISO 2019—SIW-024)
- “SISO Space Reference FOM - Tools and Testing” (SISO 2020-SIW-027)

SISO

Simulation Interoperability Standards Organization

"Simulation Interoperability & Reuse through Standards"

QUESTIONS