Project SISO Space Reference FOM

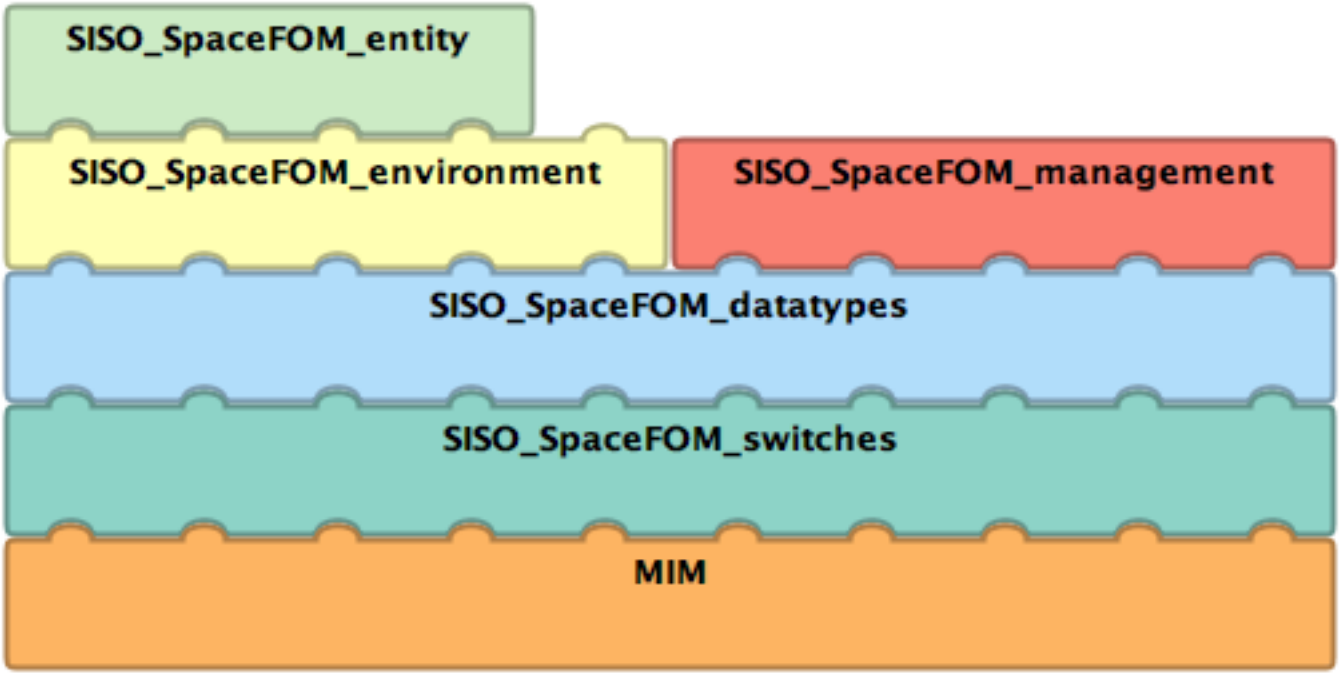


Table of contents

1. Module SISO_SpaceFOM_datatypes

Information

Name:	SISO Space FOM Core
Type:	FOM
Version:	0.3D
Modification Date:	2018-09-12
Security Classification:	Unclassified
Purpose:	SISO Space FOM Data Types
Application Domain:	
Description:	Definitions of low level data types.
Use Limitation:	
Other:	

Primary author Point Of Contact

Name:	Edwin Z. Crues
Organization:	NASA Johnson Space Center, Houston, TX
Telephone:	+1-281-483-2902
Email:	edwin.z.crues@nasa.gov

Contributor Point Of Contact

Name:	Daniel E. Dexter
Organization:	NASA Johnson Space Center, Houston, TX
Telephone:	+1-281-483-1142
Email:	daniel.e.dexter@nasa.gov

Contributor Point Of Contact

Name:	Björn Möller
Organization:	Pitch Technologies, Sweden
Telephone:	

Email:	bjorn.moller@pitch.se

Contributor Point Of Contact

Name:	Alfredo Garro
Organization:	DIMES Department, University of Calabria, Italy
Telephone:	+39-0984-49-4795
Email:	alfredo.garro@unical.it

Dependencies

SISO_SpaceFOM_switches

1.1. Datatypes

1.1.1. Simple Datatypes

Angle

Representation: HLAfloat64LE
Units: radian (r)
Resolution: NA
Accuracy: NA
Semantics: A scalar angular counterclockwise quantity.

Mass

Representation: HLAfloat64LE
Units: kilogram (kg)
Resolution: NA
Accuracy: NA
Semantics: A measurement of mass.

MassRate

Representation: HLAfloat64LE
Units: kilogram per second (kg/s)
Resolution: NA
Accuracy: NA
Semantics: A measurement of the rate of change of mass.

MassMomentOfInertia

Representation: HLAfloat64LE
Units: kilogram meter squared (kg*m^2)
Resolution: NA
Accuracy: NA

Semantics: A scalar moment or coefficient of inertia. There are nine such scalars in a moment of inertia matrix.

Length

Representation: HLAfloat64LE
Units: meter (m)
Resolution: NA
Accuracy: NA
Semantics: A scalar length.

Velocity

Representation: HLAfloat64LE
Units: meter per second (m/s)
Resolution: NA
Accuracy: NA
Semantics: A scalar translational velocity.

Acceleration

Representation: HLAfloat64LE
Units: meter per second squared (m/s^2)
Resolution: NA
Accuracy: NA
Semantics: A scalar translational acceleration.

Scalar

Representation: HLAfloat64LE
Units: NA
Resolution: NA
Accuracy: NA
Semantics: A unitless scalar value.

AngularRate

Representation: HLAfloat64LE
Units: radian per second (r/s)
Resolution: NA
Accuracy: NA
Semantics: A scalar angular rate.

AngularAcceleration

Representation: HLAfloat64LE
Units: radian per second squared (r/s^2)
Resolution: NA
Accuracy: NA
Semantics: A scalar angular acceleration.

Time

Representation: HLAfloat64LE
Units: second (s)
Resolution: NA
Accuracy: NA
Semantics: A measurement of time.

Energy

Representation: HLAfloat64LE
Units: Joule (J)
Resolution: NA
Accuracy: NA
Semantics: A measure of energy.

Power

Representation: HLAfloat64LE
Units: Watt (W)
Resolution: NA
Accuracy: NA
Semantics: A measure of power.

SignalStrength

Representation: HLAfloat64LE
Units: Decibel (dB)
Resolution: NA
Accuracy: NA
Semantics: A measure of signal strength.

Temperature

Representation: HLAfloat64LE
Units: Kelvin (K)
Resolution: NA
Accuracy: NA
Semantics: A measure of absolute temperature.

TemperatureRate

Representation: HLAfloat64LE
Units: Kelvin per second (K/s)
Resolution: NA
Accuracy: NA
Semantics: A measure of the time rate of change of temperature.

Force

Representation: HLAfloat64LE
Units: Newton (N)

Resolution: NA
Accuracy: NA
Semantics: A scalar measurement of force.

Torque

Representation: HLAfloat64LE
Units: Newton meter (N*m)
Resolution: NA
Accuracy: NA
Semantics: A scalar measurement of torque.

Density

Representation: HLAfloat64LE
Units: kilograms per cubic meter (kg/m^3)
Resolution: NA
Accuracy: NA
Semantics: A measure of mass denisty.

MassMomentOfInertiaRate

Representation: HLAfloat64LE
Units: kilogram meter squared per second (kg*m^2/s)
Resolution: NA
Accuracy: NA
Semantics: A measure of the time rate of change of a mass moment of inertia parameter.

1.1.2. Array Datatypes

PositionVector

Element Type: Length
Cardinality: 3
Encoding: HLAfixedArray
Semantics: *A 3-vector that specifies the translational position of one point with respect to another. This data type does not specify which points are involved, nor does it specify the coordinate axes onto which the three components of the vector are projected.*

VelocityVector

Element Type: Velocity
Cardinality: 3
Encoding: HLAfixedArray
Semantics: *A 3-vector that specifies the time derivative of the vector position of some point with respect to another as seen by an observer fixed in some reference frame. This data type does not specify which points are involved, nor does it specify the observer frame of reference, nor does it specify the coordinate axes onto which the three components of the vector are projected.*

AccelerationVector

Element Type: Acceleration
Cardinality: 3
Encoding: HLAfixedArray
Semantics: *A 3-vector that specifies the time derivative of a vector velocity of some point with respect to another as seen by an observer fixed in some reference frame. This data type does not specify which points are involved, nor does it specify the observer frame of reference, nor does it specify the coordinate axes onto which the three components of the vector are projected.*

AngularVelocityVector

Element Type: AngularRate
Cardinality: 3
Encoding: HLAfixedArray

Semantics: *A 3-vector that specifies the time derivative of the orientation of one reference frame with respect to another. This data type does not specify the reference frames, nor does it specify the coordinate axes onto which the three components of the vector are projected.*

AngularAccelerationVector

Element Type: AngularAcceleration
Cardinality: 3
Encoding: HLAfixedArray
Semantics: *A 3-vector that specifies the time derivative of an angular velocity vector as seen by an observer fixed in some reference frame. This data type does not specify which angular velocity, nor does it specify the observer frame of reference, nor does it specify the coordinate axes onto which the three components of the vector are projected.*

InertiaMatrix

Element Type: MassMomentOfInertia
Cardinality: 9
Encoding: HLAfixedArray
Semantics: *A 3x3 matrix that specifies the mass inertia matrix of a body about some coordinate axes. The nine elements of the matrix are stored row-wise, namely: Ixx, Ixy, Ixz, Iyx, Iyy, Iyz, Izx, Izy, Izz. The off-diagonal components Iij (i!=j) are the so-called 'negative integrals'. This means that the elements in this inertia matrix satisfy the equation H=Iw. Where H is the angular momentum vector, I is the inertia matrix and w is the angular velocity vector. This data type does not specify the coordinate axes about which the moments are calculated.*

Vector

Element Type: Scalar
Cardinality: 3
Encoding: HLAfixedArray
Semantics: *A unitless 3-vector.*

Matrix

Element Type: Scalar
Cardinality: 9
Encoding: HLAfixedArray

Semantics: *A unitless 3x3 matrix. The nine elements of the matrix are stored row-wise, namely: m11, m12, m13, m21, m22, m23, m31, m32, m33, where the first index is the row index.*

ForceVector

Element Type: Force
Cardinality: 3
Encoding: HLAfixedArray
Semantics: *A 3-vector that specifies the vector force. This data type does not specify which points are involved, nor does it specify the coordinate axes onto which the three components of the vector are projected.*

TorqueVector

Element Type: Torque
Cardinality: 3
Encoding: HLAfixedArray
Semantics: *A 3-vector that specifies the vector torque. This data type does not specify which points are involved, nor does it specify the coordinate axes onto which the three components of the vector are projected.*

InertiaRateMatrix

Element Type: MassMomentOfInertiaRate
Cardinality: 9
Encoding: HLAfixedArray
Semantics: *A 3x3 matrix that specifies the time rate of change of the parameters in the InertiaMatrix. The elements in this matrix correspond directly to the elements in the InteriaMatrix.*

1.1.3. Fixed Record Datatypes

ReferenceFrameTranslation

Encoding: HLAfixedRecord

Semantics: *This is the translational state of a subject reference frame with respect to a 'referent' frame. This data type does not specify the two reference frames.*

Name	Type	Semantic
position	PositionVector	*Position of the subject frame origin with respect to the referent origin with components expressed in the referent coordinate axes.*
velocity	VelocityVector	*Velocity of the subject frame origin with respect to its referent origin with components expressed in the referent coordinate axes.*

ReferenceFrameRotation

Encoding: HLAfixedRecord

Semantics: *This is the rotational state of a reference frame with respect to a 'referent' frame.*

Name	Type	Semantic
attitude_quaternion	AttitudeQuaternion	*Attitude quaternion that specifies the orientation of the subject frame with respect to the referent.*
angular_velocity	AngularVelocityVector	*Angular velocity of the subject frame with respect to the referent with components resolved onto the subject coordinate axes.*

AttitudeQuaternion

Encoding: HLAfixedRecord

Semantics: *This is a quaternion quantifying the orientation of a 'subject' reference frame with respect to some other 'referent' frame. Quaternions consist of one scalar component and a 3-element vector component and can be denoted Q = (s, V), where s is a scalar and V is a vector (x, y, z). Confusion often arises regarding the meaning of the quaternion. (There are several similar but incompatible conventions.) This data type defines a so-called 'left unit transformation quaternion'. It may be used to transform the elements of a vector V_R = (xR, yR, zR) resolved in the referent frame's coordinate axes into the corresponding elements of a vector V_S = (xS, yS, zS) resolved in the subject frame's coordinate axes (i.e. determine the components of the same vector in another coordinate system). The quaternion transformation formula is (0, V_A) = Q . (0, V_B) . Q*, where '.' denotes quaternion multiplication and * denotes quaternion conjugation. The formula for multiplication of the quaternion Q1 = (s1, V1) where V1 is the vector (x1, y1, z1) by the quaternion Q2 = (s2, V2) where V2 is the vector (x2, y2, z2) is Q3 = Q1 . Q2 = (s3, V3) where s3 = s1 s2 - (x1 x2 + y1 y2 + z1 z2) and V3 = (x3, y3, z3) with x3 = s1 x2 + s2 x1 + (y1 z2 - z1 y2), y3 = s1 y2 + s2 y1 + (z1 x2 - x1 z2) and z3 = s1 z2 + s2 z1 + (x1 y2 - y1 x2). The formula for conjugation of the quaternion Q = (s, V), where V is the vector (x, y, z) is Q* = (s, -V) = (s, (-x, -y, -z)).*

Name	Type	Semantic
scalar	Scalar	*The scalar component of the quaternion.*
vector	Vector	*The vector component of the quaternion.*

SpaceTimeCoordinateState

Encoding: HLAfixedRecord

Semantics: *A multi-dimensional representation of an observational coordinate frame and associated state. There are three spatial dimensions, three attitude dimensions and one time dimension. The spatial and attitude components define a right-handed orthogonal set of coordinate axes that constitute a reference frame. The time dimension specifies the 'position' of the coordinate with respect to the physical time scale (TT).*

Name	Type	Semantic
translational_state	ReferenceFrameTranslation	*This is the reference frame's translational state with respect to its parent frame. If this frame has no parent, this attribute is meaningless.*
rotational_state	ReferenceFrameRotation	*This is the reference frame's rotational state with respect to its parent frame. If this frame has no parent, this attribute is meaningless.*
time	Time	*This specifies the simulated physical time (TT), which represents the time dimension associated with a reference frame state. It is the fourth component along with the three spatial dimensions that define a reference frame coordinate state.*

1.2. Time Representation

Timestamp

Datatype: HLAinteger64Time
Semantics: *Microseconds since the beginning of the federation execution.*

Lookahead

Datatype: HLAinteger64Time
Semantics: *Microseconds*

2. Module SISO_SpaceFOM_environment

Information

Name:	SISO Space Environment FOM
Type:	FOM
Version:	0.3D
Modification Date:	2018-09-12
Security Classification:	Unclassified
Purpose:	SISO Space Environment definitions
Application Domain:	
Description:	Definitions of environment related object classes.
Use Limitation:	
Other:	

Primary author Point Of Contact

Name:	Edwin Z. Crues
Organization:	NASA Johnson Space Center, Houston, TX
Telephone:	+1-281-483-2902
Email:	edwin.z.crues@nasa.gov

Contributor Point Of Contact

Name:	Daniel E. Dexter
Organization:	NASA Johnson Space Center, Houston, TX
Telephone:	+1-281-483-1142
Email:	daniel.e.dexter@nasa.gov

Contributor Point Of Contact

Name:	Björn Möller
Organization:	Pitch Technologies, Sweden
Telephone:	

Email:	bjorn.moller@pitch.se

Contributor Point Of Contact

Name:	Alfredo Garro
Organization:	DIMES Department, University of Calabria, Italy
Telephone:	+39-0984-49-4795
Email:	alfredo.garro@unical.it

Dependencies

SISO_SpaceFOM_datatypes
SISO_SpaceFOM_switches

2.1. Object Classes

2.1.1. **ReferenceFrame**

Full Name: HLAobjectRoot.ReferenceFrame

Sharing: Publish/Subscribe

Semantics: *This is an observational reference frame along with a companion right-handed orthogonal set of coordinate axes that are fixed in the frame.*

Attributes:

name	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	HLAunicodeString	PS		TS	HLAreliable	
	Update type	**Update Condition**				
	Static	*At initialization*				
	Semantics					
	A unique name for this reference frame instance. Reference frame names are essential in forming 'links' between parent/child reference frames.					
parent_name	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	HLAunicodeString	PS		TS	HLAreliable	
	Update type	**Update Condition**				
	Static	*At initialization*				
	Semantics					
	The name of this frame's parent reference frame. If this frame has no parent (i.e., is a 'root' reference frame), then this string must be empty, otherwise the non-empty string must correspond to the name attribute of some other ReferenceFrame object instance in the simulation.					

state	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	SpaceTimeCoordinateState	PS		TS	HLAreliable	
	Update type	**Update Condition**				
	Not applicable	*NA*				
	Semantics					
	A four dimensional representation of the reference frame with respect to its parent reference frame. If the parent fame is an empty string, then only the time dimension has meaning.					
HLAprivilegeToDeleteObject *Inherited from HLAobjectRoot in MIM*	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	HLAtoken	PS	DA	TS	HLAreliable	
	Update type	**Update Condition**				
	Static	*NA*				
	Semantics					

3. Module SISO_SpaceFOM_entity

Information

Name:	SISO Space FOM
Type:	FOM
Version:	0.3D
Modification Date:	2018-09-12
Security Classification:	Unclassified
Purpose:	SISO Space Entity definitions
Application Domain:	
Description:	Definitions of entity and interface related object classes.
Use Limitation:	
Other:	

Primary author Point Of Contact

Name:	Edwin Zack Crues
Organization:	NASA Johnson Space Center
Telephone:	1-281-483-2902
Email:	edwin.z.crues@nasa.gov

Contributor Point Of Contact

Name:	Dan Dexter
Organization:	NASA Johnson Space Center
Telephone:	1-281-483-1142
Email:	dan.e.dexter@nasa.gov

Contributor Point Of Contact

Name:	Björn Möller
Organization:	Pitch Technologies, Sweden
Telephone:	

Email:	bjorn.moller@pitch.se

Contributor Point Of Contact

Name:	Alfredo Garro
Organization:	DIMES Department, University of Calabria, Italy
Telephone:	+39-0984-49-4795
Email:	alfredo.garro@unical.it

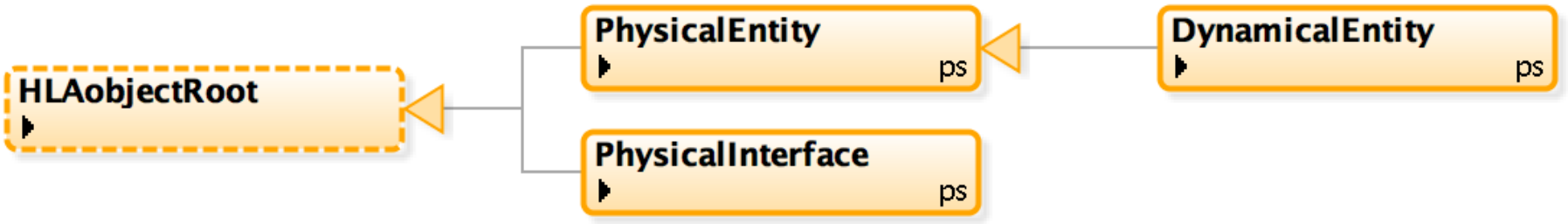
Contributor Point Of Contact

Name:	Daniel Verret
Organization:	ForwardSim Inc.
Telephone:	418-650-3493 ext 6000
Email:	dverret@forwardsim.com

Dependencies

SISO_SpaceFOM_datatypes
SISO_SpaceFOM_environment
SISO_SpaceFOM_switches

3.1. Object Classes



3.1.1. **PhysicalEntity**

Full Name: HLAobjectRoot.PhysicalEntity

Sharing: Publish/Subscribe

Semantics: *A PhysicalEntity is the highest-level object class in the Space FOM entity hierarchy. This object class provides attributes to describe an entity's location in time and space. It also contains attributes to uniquely identify it individually from all other physical entities in the federation execution.*

Physical entities have two intrinsically associated reference frames: a 'structural frame' and a 'body frame'. These are not registered in the Federation Execution's Reference Frame tree (see 6.2.2) but are used to place and orient the entity in space with respect to a reference frame in that tree. The origin of the structural frame is located at some arbitrary but known point on the entity. The body frame origin is at the entity's center of mass. The body frame is located with respect to the entity's structural reference frame by a vector from the origin of the structural reference frame to the center of mass of the entity. This vector is expressed in the entity's structural reference frame. The orientation of the entity's body frame with respect to the entity's structural reference frame is defined by an attitude quaternion.

The position and attitude of an entity is therefore defined by the position and attitude of the entity's body frame with respect to the entity's parent_reference_frame, which must be a reference frame instance in the Federation Execution's Reference Frame Tree. This, along with time, the center_of_mass vector and body_wrt_structural attitude quaternion, can be used to unambiguously locate the entity in time and space

Attributes:

name	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	HLAunicodeString	PS		TS	HLAreliable	
	Update type	**Update Condition**				
	Static	*during initialization*				
	Semantics					
	A non-empty string that identifies the entity. Each entity instance in the federation must have a unique name.					
type	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	HLAunicodeString	PS		TS	HLAreliable	
	Update type	**Update Condition**				
	Static	*during initialization*				
	Semantics					
	A non-empty string that identifies the entity type. It is not a mandatory field but it can be used to differentiate from a fuel tank and a space vehicle for example.					
status	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	HLAunicodeString	PS	DA	TS	HLAreliable	
	Update type	**Update Condition**				
	Periodic	*when changes*				
	Semantics					
	An informative string that documents the current status of the entity (whatever that might be).					
parent_reference_frame	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	HLAunicodeString	PS	DA	TS	HLAreliable	
	Update type	**Update Condition**				
	Periodic	*when changes*				
	Semantics					
	The non-empty string that identifies the reference frame with respect to which the kinematic state attributes of this entity are calculated. This string must exactly match the name of some ReferenceFrame instance in the federation.					
state	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	SpaceTimeCoordinateState	PS	DA	TS	HLAreliable	
	Update type	**Update Condition**				
	Periodic	*when changes*				
	Semantics					
	A four dimensional representation of the entity's translational and rotational state with respect to its parent reference frame.					

acceleration	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	AccelerationVector	PS	DA	TS	HLAreliable	
	Update type	**Update Condition**				
	Periodic	*when changes*				
	Semantics					
	A 3-vector that specifies the acceleration of the entity body frame origin (i.e., the entity's center of mass) with respect to the parent reference frame. This is the time derivative of the velocity vector as seen by an observer fixed in the parent frame. The components of this vector are resolved onto the coordinate axes of the parent frame.					
rotational_acceleration	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	AngularAccelerationVector	PS	DA	TS	HLAreliable	
	Update type	**Update Condition**				
	Periodic	*when changes*				
	Semantics					
	A 3-vector that specifies the angular acceleration of the entity body frame with respect to the parent reference frame. This is the time derivative of the angular velocity vector as seen by an observer fixed in the parent frame. The components of this vector are resolved onto the coordinate axes of the entity body frame.					
center_of_mass	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	PositionVector	PS	DA	TS	HLAreliable	
	Update type	**Update Condition**				
	Periodic	*when changes*				
	Semantics					
	A 3-vector that specifies the position of the entity center of mass (the body frame origin) with respect to the origin of the entity's structural frame. The components of this vector are resolved onto the coordinate axes of the structural frame.					
body_wrt_structural	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	AttitudeQuaternion	PS	DA	TS	HLAreliable	
	Update type	**Update Condition**				
	Static	*Not updated*				
	Semantics					
	An attitude quaternion that specifies the orientation of an entity's body frame with respect to the entity's structural frame. This attitude quaternion should never change. If not specified, an identity quaternion is assumed.					

HLAprivilegeToDeleteObject *Inherited from HLAobjectRoot in MIM*	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	HLAtoken	PS	DA	TS	HLAreliable	
	Update type	**Update Condition**				
	Static	*NA*				
	Semantics					

3.1.2. **DynamicalEntity**

Full Name: HLAobjectRoot.PhysicalEntity.DynamicalEntity

Sharing: Publish/Subscribe

Semantics: *The DynamicalEntity object class extends the PhysicalEntity object class to provide additional attributes associated with an object subject to non-conservative dynamic forces and/or torques. Specifically, the DynamicalEntity provides additional force, torque and mass property related parameters. These are usually associated with environmental effects and vehicle effector systems. These can be used for both visualization and to improve state propagation between updates.*

Attributes:

force	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	ForceVector	PS		TS	HLAreliable	
	Update type	**Update Condition**				
	Periodic	*when changes*				
	Semantics					
	A 3-vector that specifies the total external force on the entity. Force is expressed and applied in the entity's structural reference frame.					
torque	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	TorqueVector	PS		TS	HLAreliable	
	Update type	**Update Condition**				
	Periodic	*when changes*				
	Semantics					
	A 3-vector that specifies the total external torque on the entity. It is expressed in the entity's structural reference frame.					

mass	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	Mass	PS	DA	TS	HLAreliable	
	Update type	**Update Condition**				
	Periodic	*when changes*				
	Semantics					
	The mass of the DynamicalEntity.					
mass_rate	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	MassRate	PS	DA	TS	HLAreliable	
	Update type	**Update Condition**				
	Periodic	*when changes*				
	Semantics					
	The time rate of change of the DynamicalEntity's mass.					
inertia	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	InertiaMatrix	PS	DA	TS	HLAreliable	
	Update type	**Update Condition**				
	Periodic	*when changes*				
	Semantics					
	A 3x3 matrix that specifies the centroid moments and coefficients of inertia with respect to the coordinate axes of the DynamicalEntity's body frame					
inertia_rate	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	InertiaRateMatrix	PS	DA	TS	HLAreliable	
	Update type	**Update Condition**				
	Periodic	*when changes*				
	Semantics					
	A 3x3 matrix that specifies the time rate of change of the parameters in the InertiaMatrix. The elements in this matrix correspond directly to the elements in the InteriaMatrix.					
name *Inherited from PhysicalEntity in SISO_SpaceFOM_entity*	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	HLAunicodeString	PS		TS	HLAreliable	
	Update type	**Update Condition**				
	Static	*during initialization*				
	Semantics					
	A non-empty string that identifies the entity. Each entity instance in the federation must have a unique name.					

type *Inherited from PhysicalEntity in SISO_SpaceFOM_entity*	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	HLAunicodeString	PS		TS	HLAreliable	
	Update type	**Update Condition**				
	Static	*during initialization*				
	Semantics					
	A non-empty string that identifies the entity type. It is not a mandatory field but it can be used to differentiate from a fuel tank and a space vehicle for example.					
status *Inherited from PhysicalEntity in SISO_SpaceFOM_entity*	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	HLAunicodeString	PS	DA	TS	HLAreliable	
	Update type	**Update Condition**				
	Periodic	*when changes*				
	Semantics					
	An informative string that documents the current status of the entity (whatever that might be).					
parent_reference_frame *Inherited from PhysicalEntity in SISO_SpaceFOM_entity*	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	HLAunicodeString	PS	DA	TS	HLAreliable	
	Update type	**Update Condition**				
	Periodic	*when changes*				
	Semantics					
	The non-empty string that identifies the reference frame with respect to which the kinematic state attributes of this entity are calculated. This string must exactly match the name of some ReferenceFrame instance in the federation.					
state *Inherited from PhysicalEntity in SISO_SpaceFOM_entity*	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	SpaceTimeCoordinateState	PS	DA	TS	HLAreliable	
	Update type	**Update Condition**				
	Periodic	*when changes*				
	Semantics					
	A four dimensional representation of the entity's translational and rotational state with respect to its parent reference frame.					

acceleration *Inherited from PhysicalEntity in SISO_SpaceFOM_entity*	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	AccelerationVector	PS	DA	TS	HLAreliable	
	Update type	**Update Condition**				
	Periodic	*when changes*				
	Semantics					
	A 3-vector that specifies the acceleration of the entity body frame origin (i.e., the entity's center of mass) with respect to the parent reference frame. This is the time derivative of the velocity vector as seen by an observer fixed in the parent frame. The components of this vector are resolved onto the coordinate axes of the parent frame.					
rotational_acceleration *Inherited from PhysicalEntity in SISO_SpaceFOM_entity*	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	AngularAccelerationVector	PS	DA	TS	HLAreliable	
	Update type	**Update Condition**				
	Periodic	*when changes*				
	Semantics					
	A 3-vector that specifies the angular acceleration of the entity body frame with respect to the parent reference frame. This is the time derivative of the angular velocity vector as seen by an observer fixed in the parent frame. The components of this vector are resolved onto the coordinate axes of the entity body frame.					
center_of_mass *Inherited from PhysicalEntity in SISO_SpaceFOM_entity*	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	PositionVector	PS	DA	TS	HLAreliable	
	Update type	**Update Condition**				
	Periodic	*when changes*				
	Semantics					
	A 3-vector that specifies the position of the entity center of mass (the body frame origin) with respect to the origin of the entity's structural frame. The components of this vector are resolved onto the coordinate axes of the structural frame.					
body_wrt_structural *Inherited from PhysicalEntity in SISO_SpaceFOM_entity*	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	AttitudeQuaternion	PS	DA	TS	HLAreliable	
	Update type	**Update Condition**				
	Static	*Not updated*				
	Semantics					
	An attitude quaternion that specifies the orientation of an entity's body frame with respect to the entity's structural frame. This attitude quaternion should never change. If not specified, an identity quaternion is assumed.					

HLAprivilegeToDeleteObject *Inherited from HLAobjectRoot in MIM*	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	HLAtoken	PS	DA	TS	HLAreliable	
	Update type	**Update Condition**				
	Static	*NA*				
	Semantics					

3.1.3. **PhysicalInterface**

Full Name: HLAobjectRoot.PhysicalInterface

Sharing: Publish/Subscribe

Semantics: *Represents a location and orientation with respect to another frame. It is intended to act as a base representation for the position and orientation of an interface associated with either another PhysicalInterface instance or a PhysicalEntity instance. In either case, the position and orientation of the interface are specified with respect to the structural reference frame of the entity to which it is attached. This Object Class can be used as a common base for derived interfaces like grapple fixtures, docking ports, berthing interfaces, etc.*

Attributes:

name	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	HLAunicodeString	PS		TS	HLAreliable	
	Update type	**Update Condition**				
	Static	*during initialization*				
	Semantics					
	A non-empty string that identifies the interface. Each PhysicalInterface instance in the federation must have a unique name.					
parent_name	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	HLAunicodeString	PS		TS	HLAreliable	
	Update type	**Update Condition**				
	Static	*during initialization*				
	Semantics					
	The HLA Object Instance Name of the PhysicalEntity or PhysicalInterface to which this interface is attached.					

position	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	PositionVector	PS	DA	TS	HLAreliable	
	Update type	**Update Condition**				
	Conditional	*when changes*				
	Semantics					
	A 3-vector that specifies the position of the interface reference frame origin with respect to the parent structural reference frame. The components of this vector are resolved onto the coordinate axes of the parent frame.					
attitude	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	AttitudeQuaternion	PS	DA	TS	HLAreliable	
	Update type	**Update Condition**				
	Periodic	*when changes*				
	Semantics					
	An attitude quaternion of the interfaces reference frame ('subject frame') with respect to its parent structural reference frame ('referent frame').					
HLAprivilegeToDeleteObject *Inherited from HLAobjectRoot in MIM*	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	HLAtoken	PS	DA	TS	HLAreliable	
	Update type	**Update Condition**				
	Static	*NA*				
	Semantics					

4. Module SISO_SpaceFOM_switches

Information

Name:	SISO Space FOM Switches
Type:	FOM
Version:	0.3D
Modification Date:	2018-09-12
Security Classification:	Unclassified
Purpose:	SISO Space FOM Switches Table
Application Domain:	
Description:	Required HLA Switches table definitions.
Use Limitation:	
Other:	

Primary author Point Of Contact

Name:	Edwin Zack Crues
Organization:	NASA Johnson Space Center
Telephone:	1-281-483-2902
Email:	edwin.z.crues@nasa.gov

Primary author Point Of Contact

Name:	Dan Dexter
Organization:	NASA Johnson Space Center
Telephone:	1-281-483-1142
Email:	dan.e.dexter@nasa.gov

Contributor Point Of Contact

Name:	Björn Möller
Organization:	Pitch Technologies, Sweden
Telephone:	

Email:	bjorn.moller@pitch.se

Contributor Point Of Contact

Name:	Alfredo Garro
Organization:	DIMES Department, University of Calabria, Italy
Telephone:	+39-0984-49-4795
Email:	alfredo.garro@unical.it

4.2. Switches

Auto Provide	Disabled
Convey Region Designator Sets	Disabled
Convey Producing Federate	Disabled
Attribute Scope Advisory	Disabled
Attribute Relevance Advisory	Disabled
Object Class Relevance Advisory	Disabled
Interaction Relevance Advisory	Disabled
Service Reporting	Disabled
Exception Reporting	Disabled
Delay Subscription Evaluation	Disabled
Automatic Resign Action	CancelThenDeleteThenDivest

5. Module SISO_SpaceFOM_management

Information

Name:	SISO_SpaceFOM_configuration
Type:	FOM
Version:	0.3D
Modification Date:	2018-09-12
Security Classification:	unclassified
Purpose:	Define Space FOM compliant federation execution configuration.
Application Domain:	
Description:	Contains the base set of parameters parameters that define a Space Reference FOM compliant federation execution.
Use Limitation:	
Other:	

Keywords

Keyword	**Taxonomy**
Space	-
FOM	-
Configuration	-
Execution	-
Federation	-

Primary author Point Of Contact

Name:	Edwin Z. Crues
Organization:	NASA Johnson Space Center, Houston, TX
Telephone:	+1-281-483-2902
Email:	edwin.z.crues@nasa.gov

Contributor Point Of Contact

Name:	Daniel E. Dexter
Organization:	NASA Johnson Space Center, Houston, TX
Telephone:	+1-281-483-1142
Email:	daniel.e.dexter@nasa.gov

Contributor Point Of Contact

Name:	Björn Möller
Organization:	Pitch Technologies, Sweden
Telephone:	
Email:	bjorn.moller@pitch.se

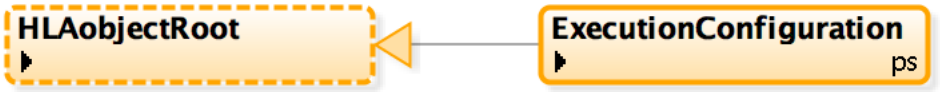
Contributor Point Of Contact

Name:	Alfredo Garro
Organization:	DIMES Department, University of Calabria, Italy
Telephone:	+39-0984-49-4795
Email:	alfredo.garro@unical.it

Dependencies

SISO_SpaceFOM_datatypes
SISO_SpaceFOM_switches

5.1. Object Classes



5.1.1. **ExecutionConfiguration**

Full Name: HLAobjectRoot.ExecutionConfiguration

Sharing: Publish/Subscribe

Semantics: *This is the federation Execution Configuration Object (ExCO). This object defines the base set of parameters necessary to coordinate federation and federate execution time lines and execution mode transitions in a SISO Space Reference FOM compliant federation execution.*

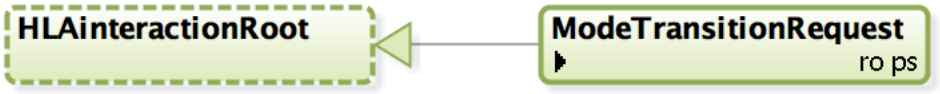
Attributes:

root_frame_name	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	HLAunicodeString	PS		TS	HLAreliable	
	Update type	**Update Condition**				
	Static					
	Semantics					
	Specifies the name of the root coordinate frame in the federation execution's reference frame tree. This frame shall remain fixed throughout the federation execution.					
scenario_time_epoch	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	Time	PS		TS	HLAreliable	
	Update type	**Update Condition**				
	Conditional					
	Semantics					
	Federation execution scenario time epoch. This is the beginning epoch expressed in Terrestrial Time (TT), using as starting epoch that of the Truncated Julian Date (TJD)- 1968-05-24 00:00:00 UTC, that corresponds to HLA logical time 0. All joining federates shall use this time to coordinate the offset between their local simulation scenario times, their local simulation execution times and the HLA logical time.					

current_execution_mode	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	ExecutionMode	PS		TS	HLAreliable	
	Update type	**Update Condition**				
	Conditional					
	Semantics					
	Defines the current running state of the federation execution in terms of a finite set of states expressed in the ExecutionMode enumeration.					
next_execution_mode	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	ExecutionMode	PS		TS	HLAreliable	
	Update type	**Update Condition**				
	Not applicable					
	Semantics					
	Defines the next running state of the federation execution in terms of a finite set of states expressed in the ExecutionMode enumeration. This is used in conjunction with the cte_mode_time, sim_mode_time and associated sync point mechanisms to coordinate federation execution mode transitions.					
next_mode_scenario_time	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	Time	PS		TS	HLAreliable	
	Update type	**Update Condition**				
	Conditional					
	Semantics					
	The time for the next federation execution mode change expressed as a federation scenario time reference. Note: this value is only meaningful for going into freeze; exiting freeze is coordinated through a sync point mechanism.					
next_mode_cte_time	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	Time	PS		TS	HLAreliable	
	Update type	**Update Condition**				
	Conditional					
	Semantics					
	The time for the next federation execution mode change expressed as a Central Timing Equipment (CTE) time reference. The standard for this reference shall be defined in the federation agreement when CTE is used.					

least_common_time_step	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	HLAinteger64Time	PS		TS	HLAreliable	
	Update type	**Update Condition**				
	Static					
	Semantics					
	*A 64 bit integer time that represents microseconds for the least common value of all the time step values in the federation execution (LCTS). This value is set by the Master Federate and does not change during the federation execution. This is used in the computation to find the next HLA Logical Time Boundary (HLTB) available to all federates in the federation execution. The basic equation is HLTB = (floor(GALT/LCTS) + 1) * LCTS, where GALT is the greatest available logical time. This is used to synchronize the federates in a federation execution to be on a common logical time boundary.*					
HLAprivilegeToDeleteObject *Inherited from HLAobjectRoot in MIM*	**Datatype**	**Sharing**	**Ownership**	**Order**	**Transportation**	**Dimensions**
	HLAtoken	PS	DA	TS	HLAreliable	
	Update type	**Update Condition**				
	Static	*NA*				
	Semantics					

5.2. Interaction Classes



5.2.1. **HLAinteractionRoot**

Full Name: HLAinteractionRoot
Sharing:
Transportation type: HLAreliable
Order: Receive
Dimensions:
Semantics:
Parameters: -

5.2.2. **ModeTransitionRequest**

Full Name: HLAinteractionRoot.ModeTransitionRequest
Sharing: Publish/Subscribe
Transportation type: HLAreliable
Order: Receive
Dimensions:
Semantics: *The ModeTransitionRequest (MTR) interaction is used by participating federates, that are not the Master Federate, to request a federation execution mode transition. An MTR can be sent at anytime during initialization or execution but only certain MTR requests are valid at certain times.*
Parameters:

Name	Datatype	Semantics
execution_mode	MTRMode	*The run mode requested. There are only 3 valid Mode Transition Request (MTR) mode values: MTR_GOTO_RUN, MTR_GOTO_FREEZE, MTR_GOTO_SHUTDOWN. Of these three valid mode requests, only 7 combinations of current execution mode and requested mode are valid:* *1. EXEC_MODE_UNINITIALIZED -> EXEC_MODE_SHUTDOWN* *2. EXEC_MODE_INITIALIZED -> EXEC_MODE_FREEZE* *3. EXEC_MODE_INITIALIZED -> EXEC_MODE_SHUTDOWN* *4. EXEC_MODE_RUNNING -> EXEC_MODE_FREEZE* *5. EXEC_MODE_RUNNING -> EXEC_MODE_SHUTDOWN* *6. EXEC_MODE_FREEZE -> EXEC_MODE_RUNNING* *7. EXEC_MODE_FREEZE -> EXEC_MODE_SHUTDOWN*

5.3. Datatypes

5.3.1. Enumerated Datatypes

ExecutionMode

Representation: HLAinteger16LE

Semantics: *Defines the mode for the running federation execution. This enumeration type is used for coordinating transitions between federation execution run states.*

Enumerator	Value
EXEC_MODE_UNINITIALIZED	0
EXEC_MODE_INITIALIZING	1
EXEC_MODE_RUNNING	2
EXEC_MODE_FREEZE	3
EXEC_MODE_SHUTDOWN	4

MTRMode

Representation: HLAinteger16LE

Semantics: *Mode Transition Request (MTR) transition values. This enumeration is used to request a specific mode transition. However, not all mode transition requests are accepted for any given Run Mode. See mode transition validation table in the Space Reference FOM documentation.*

Enumerator	Value
MTR_GOTO_RUN	2
MTR_GOTO_FREEZE	3
MTR_GOTO_SHUTDOWN	4

5.4. Synchronization Points

initialization_started

Tag Datatype: NA

Capability: Register And Achieve

Semantics: *Used to indicate that the initialization phase of a Space FOM compliant federation execution has been started. This synchronization point (sync-point) is not created until all federates required by the Master Federate have joined the federation execution. Once this occurs, the Master Federate announces this sync-point along with the "startup" sync-point for any federates that have already joined the federation execution. All federates in the sync-point group must achieve this sync-point prior to proceeding with federate and federation execution initialization.*

initialization_completed

Tag Datatype: NA

Capability: Register

Semantics: *This synchronization point (sync-point) is registered by the federation execution Master Federate after all the early joining federates have achieved the "initialization_started" sync-point. This signals to any late joining federates that they can now proceed to the current run mode of the federation execution. This sync-point will never be achieved.*

objects_discovered

Tag Datatype: NA

Capability: Register And Achieve

Semantics: *This synchronization point (sync-point) is used to mark the point at which all required objects have been discovered by all the federates taking part in the initialization process. This sync-point is used to insure that all the necessary objects have been discovered prior to proceeding with the root reference frame discovery process and then multi-phase initialization.*

mtr_run

Tag Datatype: NA

Capability: Register And Achieve

Semantics: *This is used to synchronize the mode transition to EXEC_MODE_RUNNING. This synchronization point (sync-point) is registered by the federation execution Master Federate upon receipt of a valid MTR interaction after sending out the associated ExCO update. Upon receiving the ExCO for the mode transition and at the associated transition time, all federates must achieve this sync-point prior to going into mode EXEC_MODE_RUNNING.*

mtr_freeze

Tag Datatype: NA

Capability: Register And Achieve

Semantics: *This is used to synchronize the mode transition to EXEC_MODE_FREEZE. This synchronization point (sync-point) is registered by the federation execution Master Federate upon receipt of a valid MTR interaction after sending out the associated ExCO update. Upon receiving the ExCO for the mode transition and at the associated transition time, all federates must achieve this sync-point prior to going into mode EXEC_MODE_FREEZE.*

mtr_shutdown

Tag Datatype: NA

Capability: Register

Semantics: *This synchronization point (sync-point) is used as a marker for the mode transition to EXEC_MODE_SHUTDOWN. This sync-point is registered by the federation execution's Master Federate to "mark" the federation execution as shutting down. This marker sync-point is used in addition to the ExCO. This sync-point is never achieved and will remain for the life of the federation execution to inform any late joining federates of shutdown and that the federates should proceed directly to their shutdown processes.*

root_frame_discovered

Tag Datatype: NA

Capability: Register And Achieve

Semantics: *This synchronization point (sync-point) is used to mark the point at which the root reference frame for this federation execution has been discovered by the Master Federate and all other federates participating in the initialization process. This is necessary prior to moving into the multi-phase initialization process.*