

Simulated Railroad Framework, <http://simulrr.sourceforge.net>

Synopsis: [000_Synopsis](#)

This file valid for step 0033.10

Issue Date: 2017-03-17

The Avatar Container

=====

1 Synopsis

The "Avatar Container" is a special MIDAS Object, that is provided together with the SMUOS Framework. It does not help to animate/simulate models (as other MIDAS Objects do), but it helps to transmit avatars' positions and orientations relative to local coordinate systems, which may be animated.

The Avatar Container is realized as X3D prototype "MoosAvatarContainer" within the file MoosAvatarContainer.x3d.

2 Purpose of the Avatar Container

The avatar container helps to avoid the "bouncing avatars problem", when avatars "enter" animated models.

3 External View

As a general rule, each model/module needs one avatar container per each differently animated coordinate system (and one for the static coordinate systems).

The avatars will only become visible, if the avatar container is initialized. Initialization of MIDAS Objects is described in [301_MidasObjects](#), where it should be noted, that an avatar container can be used outside of all models / modules (directly within the frame), in which case it can be directly initialized with the "commParam" reference (using an input field "commParam" instead of "modParam").

The "isBound" events of all viewpoints of the animated coordinate system should be routed to the "set_bind" field of the avatar container. This is achieved by referencing the viewpoints in the "viewpoints" (MFNode) field of the avatar container. The avatar container maintains dynamic routes internally. No need to use explicit <ROUTE> commands.

4 Internal View

The SSC Base supports the avatar container (it is said "the SSC Base hosts the MASTER avatar container").

To support the avatar container, the SSC Base

- provides some fields at the eiControl interface (refer to section 4.1.1)
- expects some fields at the eiControl interface (refer to section 4.1.2)
- defines a new CSCR to request commState changes (refer to section 4.2)
- uses new parameters in the commState (refer to section 4.3)

4.1 The Interface with the SSC Base (eiControl)

4.1.1 Fields of the SSC Base at eiControl

announceAvaCon (SFNode)

when the avatar container has been initialized (during the initialization of the frame, of the module or of the model), then it tells the SSC Base that it is now able to hold avatars and to report user's position and orientation

disableAvaCon (SFNode)

when the avatar container is being disabled (before a module is unloaded or before a model is unloaded), then it informs the SSC Base that it is no more able to hold avatars nor to report user's position/orientation

bindAvaCon (SFNode)

when the user has sent the "set_bind" event to the avatar container, then it will request from the SSC Base to move the avatar of the user of the current scene instance into this container in all scene instances.

Furthermore the SSC Base will switch the responsibility to report user's position/orientation to this avatar container in this scene instance.

reportOwnPosition (SFVec3f)

each avatar container maintains a dynamic route from the internal <ProximitySensor> ("position_changed" field) to the SSC Base. The proximity sensor that is enabled will report the avatar's position

reportOwnRotation (SFRotation)

each avatar container maintains a dynamic route from the internal <ProximitySensor> ("orientation_changed" field) to the SSC Base. The proximity sensor that is enabled will report the avatar's orientation

4.1.1 Fields of the Avatar Container at eiControl

extObjId (SFString)

the SSC Base needs this field to determine the extended object ID of an avatar container.

activatePosOriReporting (SFString)

the avatar container maintains a dynamic route from commParam.setPosOriReporting to this field. When the SSC Base broadcasts an extended object ID via this field, then all avatar containers will update the "enabled" field of their internal <ProximitySensor> node.

avatarGroup (SFNode)

this field is a reference to the <Group> node, that is contained in the avatar container. The SSC Base will use this field to add/remove avatars to/from the <Group> node.

joinedPosition (SFVec3f)

the avatar container maintains a dynamic route from commParam.joinedPosition to this field. When the SSC Base broadcasts a joined position, then this will be stored for the next "joinAvatar" event.

joinedOrientation (SFRotation)

the avatar container maintains a dynamic route from commParam.joinedOrientation to this field. When the SSC Base broadcasts a joined orientation, then this will be stored for the next "joinAvatar" event.

joinAvatar (SFInt32)

the avatar container maintains a dynamic route from commParam.joinAvatar to this field. When the SSC Base broadcasts the sessionId of a (remote) user to be joined and when the avatar container finds an avatar with this sessionId in his avatarGroup, then the avatar container will bind the viewpoint "viewpoints[0]" and - after some wait time - manually set position and orientation via Browser.setViewpointByValue. This "join user" function is only available in the case of BS Contact.

4.2 Communication State Change Requests (CSCRs)

When an avatar container sends a bind request to the SSC Base (see bindAvaCon), then this will create a CSCR and send it to the central controller

- bindAvatar;<extObjId>;<sessionId>
this means: the avatar container <extObjId> (i.e the module name and the objId of the avatar container) requests to be bound in scene instance <sessionId>. The result shall be that the avatars with sessionId <sessionId> shall be added to the <Group> nodes of this avatar container in ALL scene instances and that this avatar container in scene instance <sessionId> will take over the role to generate position and orientation reports for this avatar.

The central controller will update the "avatar container binding" in the commState (see next chapter) and will distribute a new version of the commState.

Only one "bindAvatar" CSCR can be outstanding at any time in any scene instance, new bind requests will be "collected" until the answer has been received. ("Collecting" means, that only the last one will be kept and sent to the central controller afterwards).

4.3 Parameters in the Communication State (commState)

- Avatar Container Binding Answer Flags
this parameters help to decide, whether a commState contains the answer to an CSCR "bindAvatar" from the present scene instance.
- avaConBinding
an MFString value with the same length as the sessionIds MFInt32.
Contains the extObjId of the bound avatar containers of each scene instance. Elements can be empty strings (''), if a scene instance has no bound avatar container.

The SSC Base keeps binding-less avatars in a temporary (invisible) list and binds those avatars as soon, as the avatar container binding information and the referenced avatar container are available.

5 Additional Info

none