

SRR History

Object:

Use Case

Basic - Avatars

Description of the Object:

The "bouncing avatars problem" is solved by the help of so-called avatar containers.

When a user binds an animated viewpoint, the positions and orientations of the avatars should be transmitted in values relative to the animated coordinate system.

Changes of the Object:

| | |
|---|---------------------------------|
| Release: | Reason for Change: |
| SRR_0030 | INT_MOD_FINISH_STEP_0030 |
| Description of the Change: | |
| The master avatar container provides the "addAvatar" and "removeAvatar" fields to add and remove avatars to/from the system of avatar containers. | |
| Each avatar container has a network sensor to send a "bindSessionId" request to all scene instances, when a sessionId shall be bound to a specific avatar container (when the user has bound a viewpoint). | |
| Currently some problems are known regarding placement of avatars that want to enter vehicles of an inactive module -> some internal changes are expected for the future, probably NOT changing the external interfaces uiObj and uiAva. | |
| Currently the avatar container concept is used only together with BS Contact in multiplayer mode. | |
| Documentation: | |
| srr/AvatarContainer.x3d | |
| srr/MasterAvatarContainer.x3d | |
| ExampleVehicles/MyFirstWagon.x3d | |
| FrameMain/Main_bscontact.x3d | |
| Release: | Reason for Change: |
| SRR_0032 | INT_MOD_RENAMING_DIRECTORIES_01 |
| Description of the Change: | |
| ExampleVehicles has been splitted into ExperimentalWagons+ExperimentalLocomotives | |
| Documentation: | |
| ExperimentalWagons/MyFirstWagon.x3d | |
| ExperimentalWagons/MySecondWagon.x3d | |
| ExperimentalLocomotives/MyFirstLocomotive.x3d | |

SRR History

Object:

Use Case

Basic - Chat

Description of the Object:

The demo layout contains a chat HUD and the HUD controller feeds the data to the <BSCollaborate> node to realize a 3D Chat functionality. This functionality is only relevant in multi-user-mode.

Changes of the Object:

| | |
|--|--------------------------|
| Release: | Reason for Change: |
| SRR_0030 | INT_MOD_FINISH_STEP_0030 |
| Description of the Change: | |
| The chat is implemented in the BS Contact versions of the main files, it is documented in the "How to play with the demo layout" papers on the release area. | |
| Documentation: | |
| Main_bscontact.x3d TestMainFirst_bscontact.x3d TestMainSecond_bscontact.x3d | |
| Release: | Reason for Change: |
| SRR_0032 | EXT_MOD_DEMO2012 |
| Description of the Change: | |
| the main files MainE.x3d and Main_trbs.x3d were adapted to work with the new chat of the test frame. | |
| Documentation: | |

SRR History

Object:

Use Case

Basic - Console Interface

Description of the Object:

Each SRR object can provide parameters, that can be changed and read out by the console interface (a command line interface, that is provided via uiControl to the frame).

Changes of the Object:

| | |
|---|---------------------------|
| Release: | Reason for Change: |
| SRR_0030 | INT_MOD_FINISH_STEP_0030 |
| Description of the Change: | |
| SrrConsolePlugin.x3d supports SRR objects when providing parameters for the console interface. | |
| The user (frame author) can send a "consoleCommand" to the SRR Controller, who will distribute the commands to the module coordinators and hence to the SRR objects. | |
| The SRR objects will answer and the answers will be collected into an MFString "consoleResponse", that is forwarded to the user (frame author) again. | |
| Supported commands: | |
| help/? | |
| options | |
| read | |
| set | |
| A "Console HUD" has been implemented as a part of the demo layout to use the console interface of the SRR Controller. | |
| Documentation: | |
| FrameMain/Main*.x3d | |
| FrameMain/ConsoleHud.x3d | |
| srr/SrrSwitchA.x3d | |
| srr/SrrDriveA.x3d | |
| srr/SrrKeyContainer.x3d | |
| Release: | Reason for Change: |
| SRR_0031 | EXT_ERR_CATCHING_VEHICLES |
| Description of the Change: | |
| The console interface is now basically available for SRR objects in vehicles. The SrrCabA has already been adapted to use the new facilities, but the other SRR objects still need some adaptation. | |
| Documentation: | |

SRR History

Object:

Use Case

Basic - Console Interface

| | |
|---|---------------------------------|
| Release: | Reason for Change: |
| SRR_0032 | INT_MOD_RENAMING_DIRECTORIES_01 |
| Description of the Change: | |
| The HUDs were moved to directory ExampleFrame | |
| Documentation: | |
| ExampleFrame/ConsoleHud.x3d | |

SRR History

Object:

Use Case

Basic - Take/Put Keys

Description of the Object:

Keys can be created within key containers and used to unlock locks.

They can be "carried" by an avatar or they can be "contained" in a key container or lock.

Changes of the Object:

Release:

SRR_0030

Reason for Change:

INT_MOD_FINISH_STEP_0030

Description of the Change:

SrrKeyContainer.x3d uses a network sensor to synchronize the state of the list of contained keys and to assure that only one user can take a key.

SrrLock.x3d is a rather simple prototype, cause it uses an embedded SrrKeyContainer.x3d prototype.

The SRR Controller and the Module Coordinator support the concept of keys by providing the functionality of binding a key container, taking keys into the list of "carried keys" and putting keys into the "bound key container".

A "Carried Keys HUD" has been implemented as a part of the demo layout to display the list of carried keys and to click on one key to put it into the bound key container.

SrrSwitchA and SrrSwitchB can contain an SrrLock, the example track geometry doesn't make use of this possibility currently.

Documentation:

srr/SrrKeyContainer.x3d

srr/SrrLock.x3d

srr/SrrControl.x3d

SRR History

Object:

Use Case

Interlocking - Central Interlocking

Description of the Object:

| |
|------------------|
| not yet realized |
|------------------|

Changes of the Object:

SRR History

Object:

Use Case

Interlocking - Locks for Points

Description of the Object:

| |
|------------------|
| not yet realized |
|------------------|

Changes of the Object:

SRR History

Object:

Use Case

Interlocking - Signals

Description of the Object:

| |
|------------------|
| not yet realized |
|------------------|

Changes of the Object:

SRR History

Object:

Use Case

Modeling - Add Models Dynamically

Description of the Object:

Vehicles can be created dynamically (at runtime) by the use of so-called "Setup Points".

Changes of the Object:

| | |
|---|---------------------------------|
| Release: | Reason for Change: |
| SRR_0030 | INT_MOD_FINISH_STEP_0030 |
| Description of the Change: | |
| <p>A vehicle model is registered via uiControl or uiMod.</p> <p>The vehicle model provides the miMod(Model) interface to the train manager.</p> <p>The user triggers the creation of a "1-vehicle-train"</p> <ul style="list-style-type: none">a) via uiControl - at SRR Controller-TMM,b) via uiObj(SrrSetupPoint) - at a setup point. <p>A change request "create" is sent to the "central train controller".</p> <p>A new "Train/Vehicle-Status" (TV-Status) is distributed to all scene instances.</p> <p>If the module is active, then the train manager of the scene instance loads the vehicle, the containing train part and the containing train and initializes them.</p> <p>The demo layout: contains an example static model of a setup point - MySetupPoint - that makes use of the SRR object SrrSetupPoint. The example modules SecondModule and ThirdModule use this model to create their setup points.</p> <p>Documentation: srr/SrrSetupPoint.x3d srr/SrrModCoordTm.x3d srr/SrrControlTm.x3d ExampleStaticModels/MySetupPoint.x3d</p> | |
| Release: | Reason for Change: |
| SRR_0032 | INT_MOD_RENAMING_DIRECTORIES_01 |
| Description of the Change: | |
| ExampleStaticModels was splitted into ExampleSmallProps+ExampleHouses | |
| Documentation: ExampleSmallProps/MySetupPoint.x3d | |

SRR History

Object:

Use Case

Modeling - Add Models Dynamically

Release:

SRR_0032

Reason for Change:

INT_ERR_NO_OF_GLOBAL_MODELS

Description of the Change:

The SrrControlTm.x3d file has been modified to allow

- 30 trains

- 30 train parts

- 30 vehicles

in the TV Status

Documentation:

SRR History

Object:

Use Case

Modeling - Add Models Statically

Description of the Object:

A module can add static models/models statically.

A module can add SRR objects directly (intrinsic models).

Changes of the Object:

| | |
|---|--|
| Release: | Reason for Change: |
| <input type="text" value="SRR_0030"/> | <input type="text" value="INT_MOD_FINISH_STEP_0030"/> |
| Description of the Change: | |
| <div>The demo layout provides several static models: ExampleStaticModels/StationHouse.x3d ExampleStaticModels/MySetupPoint.x3d ExampleStaticModels/Windmill.x3d where the windmill is not strictly an SrrTrains model, because it does not contain any SRR objects (and needs not provide the interface miMod(Model) therefore). The carousel in the module FirstModule is an intrinsic model, for example. It cannot be used in another module. The vehicle models can be used as static models, too, see MyFirstWagon in FirstModule.</div> | |
| Documentation: ExampleStaticModels/* FirstModule/* SecondModule/* ThirdModule/* | |
| Release: | Reason for Change: |
| <input type="text" value="SRR_0032"/> | <input type="text" value="INT_MOD_RENAMING_DIRECTORIES_01"/> |
| Description of the Change: | |
| <div>ExampleStaticModels was splitted into ExampleSmallProps+ExampleHouses</div> | |
| Documentation: ExampleSmallProps/* ExampleHouses/* | |

SRR History

Object:

Use Case

Modeling - Build a Frame

Description of the Object:

The user (frame author) can take the frame of the demo layout as an example for an own frame.

Changes of the Object:

| | |
|--|--------------------------|
| Release: | Reason for Change: |
| SRR_0030 | INT_MOD_FINISH_STEP_0030 |
| Description of the Change: | |
| The frame of the demo layout has been produced purely by X3D-Edit. | |
| The user can take the frame of the demo layout as an example for an own frame. | |
| Documentation: | |
| FrameMain/* ExampleFrame/* | |

| | |
|---|------------------------|
| Release: | Reason for Change: |
| SRR_0031 | INT_MOD_COMMENTS_FRAME |
| Description of the Change: | |
| Following files' inline comments have been improved: <ul style="list-style-type: none">- ExampleFrame.x3d- ChatHud.x3d- ConsoleHud.x3d- ControlHud.x3d- KeyHooks.x3d- TracerHud.x3d- VehicleHud.x3d | |
| Documentation: | |

| | |
|--|--------------------|
| Release: | Reason for Change: |
| SRR_0032 | EXT_MOD_DEMO2012 |
| Description of the Change: | |
| the file ExampleFrame.x3d has been changed: <ul style="list-style-type: none">- removed the "brown frame" around the layout and increased the size of the floor, windmill got an own pedestal- introduced a DirectionalLight 0 -1 0 | |
| Documentation: | |

SRR History

Object:

Use Case

Modeling - Build a Model - Carousel

Description of the Object:

The user (module/model author) can use the carousel of the demo layout as an example to build an own carousel.

Changes of the Object:

| | |
|--|--------------------------|
| Release: | Reason for Change: |
| SRR_0030 | INT_MOD_FINISH_STEP_0030 |
| Description of the Change: | |
| The module FirstModule of the demo layout contains an intrinsic model "Carousel". | |
| This intrinsic model uses the SRR objects SrrLock SrrSwitchA SrrDriveA to instrument the carousel. | |
| Documentation: FirstModule/FirstModule.x3d | |

SRR History

Object:

Use Case

Modeling - Build a Model - House

Description of the Object:

The user (module/model author) can use the station house of the demo layout as an example to build an own model of a house.

Changes of the Object:

| | |
|--|---------------------------------|
| Release: | Reason for Change: |
| SRR_0030 | INT_MOD_FINISH_STEP_0030 |
| Description of the Change: | |
| The module SecondModule of the demo layout contains a static model "StationHouse". | |
| This static model uses the SRR objects SrrLock SrrSwitchA SrrKeyContainer to instrument the station house. | |
| Documentation: SecondModule/SecondModule.x3d ExampleStaticModels/StationHouse.x3d | |
| Release: | Reason for Change: |
| SRR_0031 | INT_MOD_COMMENTS_MODELS |
| Description of the Change: | |
| The inline comments in the files - StationHouse.x3d - Windmill.x3d have been improved | |
| Documentation: | |
| Release: | Reason for Change: |
| SRR_0032 | INT_MOD_RENAMING_DIRECTORIES_01 |
| Description of the Change: | |
| ExampleStaticModels was splitted into ExampleHouses+ExampleSmallProps | |
| Documentation: ExampleHouses/* | |

SRR History

Object:

Use Case

Modeling - Build a Model - Locomotive

Description of the Object:

The user (model author) can take the locomotive of the demo layout as an example to create an own model of a locomotive.

Changes of the Object:

| | |
|--|--------------------------|
| Release: | Reason for Change: |
| SRR_0030 | INT_MOD_FINISH_STEP_0030 |
| Description of the Change: | |
| The example vehicle model MyFirstLocomotive is contained in the directory ExampleVehicles. | |
| It uses the SRR objects SrrAvatarContainer SrrA1plus2 SrrAxle SrrTransformationA SrrControlFloatB SrrCabA SrrControlIntA SrrControlBoolA SrrDriveA to instrument the locomotive. | |
| Documentation: ExampleVehicles/MyFirstLocomotive.x3d | |

| | |
|--|--------------------------------|
| Release: | Reason for Change: |
| SRR_0031 | INT_MOD_NEW_SRR_OBJECT_TRIGGER |
| Description of the Change: | |
| The example locomotive MyFirstLocomotive uses the SRR object SrrTrigger additionally, to instrument a whistle. | |
| Documentation: | |

| | |
|--|-----------------------------------|
| Release: | Reason for Change: |
| SRR_0031 | EXT_MOD_PREPARE_FOR_3RDPARTY_LOCO |
| Description of the Change: | |
| The inline documentation of the example locomotive has been enhanced | |
| Documentation: ExampleVehicles/MyFirstLocomotive.x3d | |

SRR History

Object:

Use Case

Modeling - Build a Model - Locomotive

| | |
|--|---------------------------------|
| Release: | Reason for Change: |
| SRR_0032 | INT_MOD_RENAMING_DIRECTORIES_01 |
| Description of the Change: | |
| ExampleVehicles was splitted into ExperimentalWagons+ExperimentalLocomotives | |
| Documentation: | |
| ExperimentalLocomotives/* | |

SRR History

Object:

Use Case

Modeling - Build a Model - Setup Point

Description of the Object:

The user (model author/module author) can take the example setup point of the demo layout as an example to build an own setup point.

Changes of the Object:

| | |
|--|-------------------------------------|
| Release: | Reason for Change: |
| SRR_0031 | INT_MOD_CREATION_PROGRESS_INDICATOR |
| Description of the Change: | |
| It's now possible to model a progress indicator in a setup point. The example setup point MySetupPoint.x3d has been enhanced to display the progress of the creation process of a vehicle. | |
| Documentation: | |
| | |
| Release: | Reason for Change: |
| SRR_0031 | INT_MOD_COMMENTS_MODELS |
| Description of the Change: | |
| The inline comments in the file - MySetupPoint.x3d have been improved | |
| Documentation: | |
| ExampleStaticModels\MySetupPoint.x3d | |
| Release: | Reason for Change: |
| SRR_0032 | INT_MOD_RENAMING_DIRECTORIES_01 |
| Description of the Change: | |
| ExampleStaticModels was splitted into ExampleSmallProps+ExampleHouses | |
| Documentation: | |
| ExampleSmallProps/MySetupPoint.x3d | |

SRR History

Object:

Use Case

Modeling - Build a Model - Tracks

Description of the Object:

The user (model author) can use the static models from the example track geometry as a template to create own track models. A "Howto" paper exists to support him/her.

Changes of the Object:

Release:

SRR_0030

Reason for Change:

INT_MOD_FINISH_STEP_0030

Description of the Change:

The example track geometry contains the static models SrrTrackSectionA and SrrTrackSectionB.

They use the SRR objects

SrrBasicTrackSection

SrrTrackNode

SrrTrackEdge

to instrument the track section.

A "Howto" paper exists to guide the model author when creating a track model.

Documentation:

How to build my own tracks and turnouts (English and German) - <http://simulrr.wordpress.com/download-area#docu>

The description of the lower layer interface of the example track geometry -

<http://members.chello.at/christoph.valentin/tginterface.htm>

tg/SrrTrackSectionA.x3d

tg/SrrTrackSectionB.x3d

SRR History

Object:

Use Case

Modeling - Build a Model - Turnouts

Description of the Object:

The user (model author) can use the static models from the example track geometry as a template to create own turnout models. A "Howto" paper exists to support him/her.

Changes of the Object:

Release:

SRR_0030

Reason for Change:

INT_MOD_FINISH_STEP_0030

Description of the Change:

The example track geometry contains the static models SrrTurnoutLeftA and SrrTurnoutRightA.

They use the SRR objects

SrrBasicTurnout2Way

SrrSwitchB

SrrTrackNode

SrrTrackEdge

to instrument the turnout.

A "Howto" paper exists to guide the model author when creating a turnout model.

Documentation:

How to build my own tracks and turnouts (English and German) - <http://simulrr.wordpress.com/download-area#docu>

The description of the lower layer interface of the example track geometry -

<http://members.chello.at/christoph.valentin/tginterface.htm>

tg/SrrTurnoutLeftA.x3d

tg/SrrTurnoutRightA.x3d

SRR History

Object:

Use Case

Modeling - Build a Model - Wagon

Description of the Object:

The user (model author) can take the wagons of the demo layout as an example to create an own model of a wagon.

Changes of the Object:

| | |
|---|--------------------------|
| Release: | Reason for Change: |
| SRR_0030 | INT_MOD_FINISH_STEP_0030 |
| Description of the Change: | |
| The example vehicle models MyFirstWagon.x3d and MySecondWagon.x3d are contained in the directory ExampleVehicles. | |
| They use the SRR objects SrrAvatarContainer SrrWagon2axA SrrAxle SrrTransformationA SrrControlBoolA SrrControlFloatA SrrControlFloatB to instrument the wagons. | |
| Documentation: ExampleVehicles/MyFirstWagon.x3d ExampleVehicles/MySecondWagon.x3d | |

| | |
|---|-----------------------------------|
| Release: | Reason for Change: |
| SRR_0031 | EXT_MOD_PREPARE_FOR_3RDPARTY_LOCO |
| Description of the Change: | |
| The inline documentation of the example wagons has been enhanced | |
| Documentation: ExampleVehicles/MyFirstWagon.x3d ExampleVehicles/MySecondWagon.x3d | |

| | |
|---|---------------------------------|
| Release: | Reason for Change: |
| SRR_0032 | INT_MOD_RENAMING_DIRECTORIES_01 |
| Description of the Change: | |
| ExampleVehicles was split into ExperimentalWagons+ExperimentalLocomotives | |
| Documentation: ExperimentalWagons/* | |

SRR History

Object:

Use Case

Modeling - Build a Module

Description of the Object:

The user (module author) can take the modules of the demo layout as examples to create own modules. A "Howto" paper supports him/her.

Changes of the Object:

| | |
|---|---|
| Release: | Reason for Change: |
| <input type="text" value="SRR_0030"/> | <input type="text" value="INT_MOD_FINISH_STEP_0030"/> |
| Description of the Change: | |
| <input ("narrowgauge")="" ("standardgauge")."="" and="" hill"),="" secondmodule="" thirdmodule="" type="text" value="The demo layout contains three example modules - FirstModule ("/> <input a="" author="" creating="" exists="" guide="" howto"="" module="" module."="" paper="" the="" to="" type="text" value="A " when=""/> | |
| Documentation: | |
| <input download-area#docu"="" http:="" simulrr.wordpress.com="" type="text" value="How to build my own module (German) - http://simulrr.wordpress.com/download-area#docu FirstModule/FirstModule.x3d SecondModule/SecondModule.x3d ThirdModule/ThirdModule.x3d"/> | |
| Release: | Reason for Change: |
| <input type="text" value="SRR_0031"/> | <input type="text" value="INT_MOD_COMMENTS_MODULES"/> |
| Description of the Change: | |
| <input type="text" value="The inline comments of the files - FirstModule.x3d - SecondModule.x3d - ThirdModule.x3d have been enhanced."/> | |
| Documentation: | |

SRR History

Object:

Use Case

Modeling - Register Module Statically

Description of the Object:

The user (frame author) can register a module statically in his frame. The frame of the demo layout can be taken as example for doing so.

Changes of the Object:

| | |
|---|--------------------------|
| Release: | Reason for Change: |
| SRR_0030 | INT_MOD_FINISH_STEP_0030 |
| Description of the Change: | |
| The demo layout contains the example frame. | |
| The example frame loads and initializes three modules statically via the miMod(Module) interface. | |
| Documentation: | |
| ExampleFrame/ExampleFrame.x3d | |

SRR History

Object:

Use Case

Programming - Build a Specific Vehicle Prototype

Description of the Object:

The user (programmer) can build a so-called "specific vehicle prototype". A "Howto" paper exists to support him/her.

Changes of the Object:

Release:

SRR_0030

Reason for Change:

INT_MOD_FINISH_STEP_0030

Description of the Change:

The SRR object SrrVehicle provides services to instrument an SrrTrains vehicle.

However, those services tend to be too abstract. Therefore it is possible to put a so-called "specific vehicle prototype" between SrrVehicle and the vehicle model to enable "easy-to-use" modeling of vehicle models.

Currently two specific vehicle prototypes exist:

SrrWagon2axA: a 2-axled wagon without brake, but providing the basic user interface for v=const and doze control.

SrrA1plus2: a "Rocket like" locomotive (axle arrangement A1 plus a 2-axled tender), providing doze control additionally.

The user of the SRR framework can build own specific vehicle prototypes, a "Howto" paper exists to help with this.

Documentation:

How to build my own specific vehicle prototype (German) - <http://simulrr.wordpress.com/download-area#docu>

srr/SrrWagon2axA.x3d

srr/SrrA1plus2.x3d

SRR History

Object:

Use Case

Programming - Build an own Track Geometry

Description of the Object:

The user (programmer) can build an own "track geometry". A "Howto" paper exists to support him/her.

Changes of the Object:

Release:

SRR_0030

Reason for Change:

INT_MOD_FINISH_STEP_0030

Description of the Change:

The SRR Framework is published along with an example track geometry.

This track geometry calculates the geometric properties of each track edge by 5 vectors and 1 scalar and returns three MFVec3f arrays to the user (model author). Those return values can be used to create a graphical representation of the track section/turnout.

It's now possible to replace this track geometry by an own track geometry, a "Howto" paper exists to help with this.

Documentation:

How to build my own tracks and turnouts (English and German) - http://simulrr.wordpress.com/download-area#docutg/*

SRR History

Object:

Use Case

Programming - Build an SRR Object

Description of the Object:

The user (programmer) can build an own SRR object. A "Howto" paper exists to support him/her.

Changes of the Object:

Release:

SRR_0030

Reason for Change:

INT_MOD_FINISH_STEP_0030

Description of the Change:

SRR objects build the major services of the SRR framework. They are used by model/module authors to "instrument" models and modules.

It's now possible to build your own SRR object, a "Howto" paper exists to help with this.

Documentation:

How to build my own SRR object (German) - http://simulrr.wordpress.com/download-area#docusrr/*

SRR History

Object:

Use Case

Trains - Basic User Interface (doze, vConst)

Description of the Object:

Each vehicle can impose a "dozing force" into the train.
Each vehicle can request to drive with constant velocity (vConst).

Changes of the Object:

Release:

SRR_0030

Reason for Change:

INT_MOD_FINISH_STEP_0030

Description of the Change:

The SrrVehicle object supports not only the "cabs and controls" concept, but also three additional SFNode fields to hold controls.

"vConstControl" - holds optionally a floating point control to set the constant velocity of the train
"vConstEnableControl" - holds optionally a boolean control to enable constant velocity of the train
"dozeControl" - holds optionally a floating point control to impose a "dozing force"

currently following controls exist:

SrrControlBoolA: simple boolean control
SrrControlIntA: simple integer control
SrrControlFloatA: simple floating point control
SrrControlFloatB: floating point control for a plane sensor

Documentation:

srr/SrrVehicle.x3d
srr/SrrTrain.x3d
srr/SrrVehicleDriveBasic.x3d

SRR History

Object:

Use Case

Trains - Couple by Gentle Collision

Description of the Object:

| |
|------------------|
| not yet realized |
|------------------|

Changes of the Object:

SRR History

Object:

Use Case

Trains - Crash by Hard Collision

Description of the Object:

| |
|------------------|
| not yet realized |
|------------------|

Changes of the Object:

SRR History

Object:

Use Case

Trains - Create Vehicle

Description of the Object:

If the user has created a "1-vehicle-train" at a setup point, the train will be initialized and start to move.

Changes of the Object:

Release:

SRR_0030

Reason for Change:

INT_MOD_FINISH_STEP_0030

Description of the Change:

If a vehicle model has been registered, the train manager can load a "1-vehicle-train", calculate the "state" of the train by the properties of the setup point and afterwards initialize the train.

First the vehicle will be initialized, still invisible.

Then the train part will be initialized.

Afterwards the train will be initialized: the train will fetch it's "state" (from the MOC) and will set the initial position of all axles. Then it will set the vehicles' visibility to "ON".

After initialization, the SRR object SrrTrain.x3d will start to calculate a deltaEss event each frame and send the event to the axles of the train.

Documentation:

srr/SrrTrain.x3d
srr/SrrTrainPart.x3d
srr/SrrVehicle.x3d
srr/SrrAxle.x3d

Release:

SRR_0031

Reason for Change:

INT_MOD_CREATION_PROGRESS_INDICATOR

Description of the Change:

Additionally, the progress of the creation process is indicated via the setup point.

Documentation:

SRR History

Object:

Use Case

Trains - Decouple Manually

Description of the Object:

| |
|------------------|
| not yet realized |
|------------------|

Changes of the Object:

SRR History

Object:

Use Case

Trains - Decoupling Track

Description of the Object:

| |
|------------------|
| not yet realized |
|------------------|

Changes of the Object:

SRR History

Object:

Use Case

Trains - Delete Train

Description of the Object:

A train and all it's vehicles can be removed from the simulation globally (in all scene instances).

Changes of the Object:

Release:

SRR_0030

Reason for Change:

INT_MOD_FINISH_STEP_0030

Description of the Change:

The train manager provides the possibility to send a change request "delete" to the central train controller. The TV-Status (Train/Vehicle-Status) will be updated and all scene instances will delete the train and all it's vehicles.

Currently a change request "delete" is sent at several occasions automatically (when a train cannot be created and the creation has to be undone),

and the SRR object SrrVehicle provides an interface to all vehicle models to request the deletion of the whole train.

Documentation:

srr/SrrModCoordTm.x3d

srr/SrrControlTm.x3d

srr/SrrVehicle.x3d

SRR History

Object:

Use Case

Trains - Derail in Curves (Speeding)

Description of the Object:

| |
|------------------|
| not yet realized |
|------------------|

Changes of the Object:

SRR History

Object:

Use Case

Trains - Derail on Points

Description of the Object:

| |
|------------------|
| not yet realized |
|------------------|

Changes of the Object:

SRR History

Object:

Use Case

Trains - Derailed Models

Description of the Object:

| |
|------------------|
| not yet realized |
|------------------|

Changes of the Object:

SRR History

Object:

Use Case

Trains - Side Collision on Points

Description of the Object:

| |
|------------------|
| not yet realized |
|------------------|

Changes of the Object:

SRR History

Object:

Use Case

Trains - Switch the Points Manually

Description of the Object:

A model of a turnout can contain a "switch" SRR object to switch the points.

Changes of the Object:

Release:

SRR_0030

Reason for Change:

INT_MOD_FINISH_STEP_0030

Description of the Change:

The SRR object SrrTrackNode expects an SrrSwitchB SRR object to be used for switching the points.

The turnout models in the example track geometry, SrrTurnoutLeftA and SrrTurnoutRightA, implement a "lever with a red knob" to provide input to and to display out of the SrrSwitchB object.

Documentation:

srr/SrrSwitchB.x3d

srr/SrrTrackNode.x3d

srr/SrrAxle.x3d

tg/SrrTurnoutLeftA.x3d

tg/SrrTurnoutRightA.x3d

SRR History

Object:

Use Case

Trains - Train Changes Module

Description of the Object:

| |
|------------------|
| not yet realized |
|------------------|

Changes of the Object:

SRR History

Object:

Use Case

Trains - Train is Moving

Description of the Object:

Trains move over the track layout either with constant velocity or according to Newton's laws.

Changes of the Object:

Release:

SRR_0030

Reason for Change:

INT_MOD_FINISH_STEP_0030

Description of the Change:

As soon, as a train is initialized, the SRR object SrrTrain will start to read parameters from the SrrVehicle objects and from the drives of the train.

It will calculate a new velocity and optionally a new acceleration of the train and tell the new velocity to all drives of the train.

Furthermore it will calculate a "deltaEss" value for all axles of the train, send this value to the axles and hence display the vehicles and axles at a new position.

Documentation:

srr/SrrTrain.x3d

srr/SrrTrainPart.x3d

srr/SrrVehicle.x3d

srr/SrrVehicleDriveA.x3d

srr/srrVehicleDriveBasic.x3d

SRR History

Object:

Use Case

Trains - Train Movers

Description of the Object:

| |
|------------------|
| not yet realized |
|------------------|

Changes of the Object:

SRR History

Object:

Use Case

Trains - User Interface (Cabs)

Description of the Object:

Vehicles can contain cabs containing cab controls in turn

Changes of the Object:

Release:

SRR_0030

Reason for Change:

INT_MOD_FINISH_STEP_0030

Description of the Change:

The simple cab SrrCabA can optionally contain two controls
"regulatorControl": an integer control to set the "regulating notch"
"directionControl": a boolean control to set the "direction" of the train

The output of the cab must be routed to the drive(s) of the vehicle. SrrVehicleDriveA supports
"regulatingNotch" and "direction" fields for input.

Routing of the parameters is done by the "parameterLinkage" field of SrrVehicle.

currently following controls exist:

SrrControlBoolA: simple boolean control

SrrControlIntA: simple integer control

SrrControlFloatA: simple floating point control

SrrControlFloatB: floating point control for a plane sensor

Documentation:

srr/SrrCabA.x3d

srr/SrrVehicle.x3d

SRR History

Object:

Subsystem

Central Controller

Description of the Object:

Physically the central controller is a part of the SRR Controller, but logically it contains own instances "CommControl" and "TrainControl" with separate logic, responsible for the Communication State and the Train/Vehicle-Status, respectively. The Central Controller runs on one dedicated scene instance.

Changes of the Object:

| | |
|---|-------------------------------------|
| Release: | Reason for Change: |
| SRR_0030 | INT_MOD_FINISH_STEP_0030 |
| Description of the Change: | |
| The Central Controller instances "CommControl" and "TrainControl" run in only one of the scene instances (in that instance, that has the "central controller role"). | |
| CommControl receives "Registration Request" messages and "Access Request" messages from the SRR Controllers (Base) of all scene instances and creates therefore a new value for the Communication State "commState". | |
| The commState is distributed and the SRR Controllers (Base) of all scene instances inform their users (frames and module coordinators) via the interfaces uiControl and iiControl, respectively. The module coordinators in turn inform their SRR objects about relevant changes in the commState. | |
| TrainControl receives "Vehicle Registration" messages and "Change Request" messages from the SRR Controllers (TMM) of all scene instances and creates therefore a new value for the Train/Vehicle-Status. The TV-Status is distributed and the SRR Controllers (TMM) of all scene instances trigger their train management functionality to create/initialize/delete trains and vehicles accordingly. | |
| Documentation: | |
| srr/SrrControl.x3d srr/SrrControlTm.x3d | |
| Release: | Reason for Change: |
| SRR_0032 | INT_ERR_UNIQUE_OBJID_GLOBAL_OBJECTS |
| Description of the Change: | |
| The TMM of the central controller sets the prefix G.TM. at the beginning of the vehicle's objId. | |
| Documentation: | |
| | |
| Release: | Reason for Change: |
| SRR_0032 | INT_MOD_TAKEMOC_UICONTROL |
| Description of the Change: | |
| The handling of the MOC requests was already prepared in the central controller -> no change | |
| Documentation: | |
| SrrControl.x3d | |

SRR History

Object:

Subsystem

Dummy - Subsystem

Description of the Object:

Changes of the Object:

| | |
|--|---------------------------------|
| Release: | Reason for Change: |
| SRR_0032 | INT_MOD_DYN_MODULE_REGISTRATION |
| Description of the Change: | |
| In a first step, no changes were done in the SRR Framework. | |
| The Testframe and the Empty Frame provide a function to add modules statically to a frame, but it is done automatically. Modules can be removed manually from the empty frame, but this is not documented. | |
| Documentation: | |
| no documentation | |

SRR History

Object:

Subsystem

Example Track Geometry

Description of the Object:

The SRR Framework is accompanied by an example track geometry, that contains not only a track geometry prototype, but also a small set of example models of tracks and turnouts.

Changes of the Object:

Release:

SRR_0030

Reason for Change:

INT_MOD_FINISH_STEP_0030

Description of the Change:

Currently the example track geometry comprises

- the track geometry node SrrTrackGeometryABI
- two example track section models, that are based on the "ABI" track geometry
- two example turnout models, that are based on the "ABI" track geometry

The geometry of each track section/turnout has to be defined by the user (module author), using geometric parameters.

The track/turnout models will take the result from the track geometry node (calculated during initialization) and display the graphical representation of the model.

The turnout models contain an "SrrSwitchB" SRR object that will use the input from a lever to switch the points. The output of the SrrSwitchB is displayed as a movement of the lever.

Documentation:

How to build my own tracks and turnouts (English and German) - <http://simulrr.wordpress.com/download-area#docutg/>

SRR History

Object:

Subsystem

SRR Framework

Description of the Object:

The SRR Framework comprises the SRR Controller and the Module Coordinator.

Changes of the Object:

| | |
|--|--------------------------|
| Release: | Reason for Change: |
| SRR_0030 | INT_MOD_FINISH_STEP_0030 |
| <p>Description of the Change:</p> <p>The SRR Framework is split into two modules</p> <ul style="list-style-type: none">- the base module (SrrControl and SrrModCoord)- the train manager module (SrrControlTm and SrrModCoordTm) <p>The services of the SRR Framework are used</p> <ul style="list-style-type: none">- by the frame (via the user interface uiControl)- by the modules (via the user interface uiMod)- by the SRR objects (via the internal interface iiObj) <p>The services of the SRR Framework can be categorized as follows</p> <ul style="list-style-type: none">- Initialization, Session Handling- Tracer- Communication State (Central Controller, Roles, Module Activation)- Module Management (Announcement/Initialization, Module Registration)- Console Interface- Keys, Key Container and Locks- Avatars and Moving Objects- Modularity of the SRR Framework- Track Management- Vehicle and Train Management (Vehicle Registration/Setup Points, Train/Vehicle Status)- Dynamic Model Support (activation and deactivation of modules) <p>Documentation:</p> <p>A general introduction can be found in the concept papers (English and German)</p> <ul style="list-style-type: none">- http://simulrr.wordpress.com/simul-rr- http://simulrr.wordpress.com/simul-rr-german <p>The user interfaces uiControl and uiMod are described in the documentation of the lower layer interface of the SRR Framework</p> <ul style="list-style-type: none">- http://members.chello.at/christoph.valentin/lowlevelinterface.htm <p>The internal interface iiObj is described in the paper "How to build my own SRR object"</p> <ul style="list-style-type: none">- http://simulrr.wordpress.com/download-area#docu <p>srr/SrrControl.x3d srr/SrrModCoord.x3d srr/SrrControlTm.x3d srr/SrrModCoordTm.x3d</p> | |

SRR History

Object:

Subsystem

SRR Framework

Release:

SRR_0031

Reason for Change:

INT_MOD_CREATION_PROGRESS_INDICATOR

Description of the Change:

The SRR Controller - TMM has been enhanced:
The modification scenarios logic has been extended to set the progress of the creation process in the setup point

Documentation:

Release:

SRR_0031

Reason for Change:

EXT_ERR_CATCHING_VEHICLES

Description of the Change:

SRR Module Coordinator: A new field was introduced at iiObj interface,
to de-announce objects from the console interface.

Documentation:

Release:

SRR_0031

Reason for Change:

EXT_MOD_PREPARE_FOR_3RDPARTY_LOCO

Description of the Change:

The SRR Controller TM has been changed to set quasiModule='Vehicles' immediately after having loaded a vehicle.

Documentation:

Release:

SRR_0032

Reason for Change:

INT_ERR_NO_OF_GLOBAL_MODELS

Description of the Change:

The SrrControlTm.x3d file has been modified to allow
30 trains
30 train parts
30 vehicles
in the TV Status

Documentation:

SRR History

Object:

Subsystem

SRR Framework

| | |
|--|-------------------------------------|
| Release: | Reason for Change: |
| SRR_0032 | INT_ERR_UNIQUE_OBJID_GLOBAL_OBJECTS |
| Description of the Change: | |
| The SRR Controller (TMM) sets the prefix G.TM. In front of the train's and of the train part's objId. | |
| Documentation: | |
| | |
| Release: | Reason for Change: |
| SRR_0032 | INT_MOD_TAKEMOC_UICONTROL |
| Description of the Change: | |
| The SRR Controller was changed to handle MOC requests and send them to the central communication controller. | |
| Documentation: | |
| SrrControl.x3d | |

SRR History

Object:

Subsystem

SRR Objects

Description of the Object:

The SRR Framework is accompanied by a basic set of SRR objects.

Changes of the Object:

Release:

SRR_0030

Reason for Change:

INT_MOD_FINISH_STEP_0030

Description of the Change:

There are several SRR objects, which are documented and can be used directly therefore.

Some SRR objects are used only internally and need not be documented therefore.

The following SRR objects are documented (their user interface uiObj/uiAva is defined)

Basic SRR Objects (no TMM necessary)

- The binary switch (SrrSwitchA)
- The carousel drive (SrrDriveA)
- The key container (SrrKeyContainer)
- The lock (SrrLock)
- The avatar containers (SrrAvatarContainer and SrrMasterAvatarContainer)

SRR Objects for Tracks and Turnouts (TMM necessary)

- The basic track section (SrrBasicTrackSection)
- The basic 2-way turnout (SrrBasicTurnout2Way)
- The n-way switch (SrrSwitchB)
- The track edge (SrrTrackEdge)
- The track node (SrrTrackNode)
- The setup point (SrrSetupPoint)

SRR Objects for Vehicles (TMM necessary)

- The 2-axled wagon type A (SrrWagon2axA)
- The "Rocket" style locomotive (SrrA1plus2)
- The axle (SrrAxle)
- The transformation type A (SrrTransformationA)
- The simple cab type A (SrrCabA)
- The boolean control type A (SrrControlBoolA)
- The integer control type A (SrrControlIntA)
- The floating point control type A (SrrControlFloatA)
- The floating point control type B for a plane sensor (SrrControlFloatB)
- The simple vehicle drive type A (SrrVehicleDriveA)

Documentation:

The user interfaces uiObj/uiAva are described in the description of the lower layer interface of the SRR Framework

- <http://members.chello.at/christoph.valentin/lowlevelinterface.htm>

SRR History

Object:

Subsystem

SRR Objects

Release:

SRR_0031

Reason for Change:

INT_MOD_CREATION_PROGRESS_INDICATOR

Description of the Change:

SrrSetupPoint:

The setup point has been extended to indicate the progress of the creation process

Documentation:

Release:

SRR_0031

Reason for Change:

EXT_ERR_CATCHING_VEHICLES

Description of the Change:

SrrCabA:

Now the SrrCabA provides a console interface for

- regulating notch
- direction

This can be used to "catch" locomotives.

Documentation:

Release:

SRR_0031

Reason for Change:

INT_MOD_NEW_SRR_OBJECT_TRIGGER

Description of the Change:

A new SRR object SrrTrigger has been implemented, that can be used in static/intrinsic and dynamic vehicle models.

An SFTIME event is taken from the user and delivered to all (active) instances of the SRR object.

Documentation:

Release:

SRR_0031

Reason for Change:

INT_ERR_HARMONIZE_SRR_OBJECTS_01

Description of the Change:

SrrControlBoolA/SrrControlFloatA|B/SrrControlIntA:

- Update Request was obsoleted
- general streamlining
- network sensor considerations: done but nothing implemented

Documentation:

SRR History

Object:

Subsystem

SRR Objects

Release:

SRR_0031

Reason for Change:

INT_ERR_HARMONIZE_SRR_OBJECTS_01

Description of the Change:

SrrDriveA:

- quasiModule was provided
- enabled was implemented
- general streamlining
- network sensor considerations: done but nothing implemented

Documentation:

Release:

SRR_0031

Reason for Change:

INT_ERR_HARMONIZE_SRR_OBJECTS_01

Description of the Change:

SrrKeyContainer:

- Update Request was obsoleted
- quasiModule was provided
- enabled was implemented
- general streamlining
- network sensor considerations: done but nothing implemented

Documentation:

Release:

SRR_0031

Reason for Change:

INT_ERR_HARMONIZE_SRR_OBJECTS_01

Description of the Change:

SrrSwitchA/SrrSwitchB:

- Update Request was obsoleted
- quasiModule was provided
- enabled was implemented
- general streamlining
- network sensor considerations: done but nothing implemented

Documentation:

Release:

SRR_0032

Reason for Change:

INT_ERR_CONSOLE_02

Description of the Change:

All SRR objects were checked to ensure following behaviour:

If an external request is received via the commParam/modParam, this request will be ignored, if "enabled" is false.

Documentation:

SRR History

Object:

Internal Interface

iiObj(SrrObject)

Description of the Object:

The SRR Framework provides an interface to the SRR objects.

Changes of the Object:

| | |
|--|-------------------------------------|
| Release: | Reason for Change: |
| SRR_0030 | INT_MOD_FINISH_STEP_0030 |
| Description of the Change: | |
| The interface iiObj is described in an "Howto" paper. | |
| Documentation: | |
| How to build my own SRR object - http://simulrr.wordpress.com/download-area#docu | |
| Release: | Reason for Change: |
| SRR_0031 | INT_MOD_CREATION_PROGRESS_INDICATOR |
| Description of the Change: | |
| A new field "setProgressIndicator" is provided by the setup point. The SRR Controller - TMM uses this field directly to indicate the status of the creation of the vehicle. | |
| Documentation: | |
| Release: | Reason for Change: |
| SRR_0031 | EXT_ERR_CATCHING_VEHICLES |
| Description of the Change: | |
| The new field deannounceObject has been introduced at the module coordinator, to de-announce objects from the console interface | |
| Documentation: | |

SRR History

Object:

Minimum Interface miMod(Model)

Description of the Object:

SRR requires a minimum interface (a minimum set of fields and their behaviour), that must be

- provided by an SrrTrains static model to an SrrTrains module,
- provided by an SrrTrains vehicle model (dynamic/global model) to the Train Manager or to an SrrTrains module.

SRR doesn't require a minimum interface from "non-SrrTrains" models nor from intrinsic models.

Changes of the Object:

| | |
|--|--|
| Release: | Reason for Change: |
| <input type="text" value="SRR_0030"/> | <input type="text" value="INT_MOD_FINISH_STEP_0030"/> |
| Description of the Change: | |
| <input type="text" value="The miMod(Model) interface is defined and described."/> | |
| Documentation: | |
| description of the lower layer interface of the SRR Framework - http://members.chello.at/christoph.valentin/lowlevelinterface.htm | |
| Release: | Reason for Change: |
| <input type="text" value="SRR_0031"/> | <input type="text" value="EXT_MOD_PREPARE_FOR_3RDPARTY_LOCO"/> |
| Description of the Change: | |
| <input type="text" value="The description is extended by the parameter quasiModule."/> | |
| Documentation: | |
| description of the lower layer interface of the SRR Framework - http://members.chello.at/christoph.valentin/lowlevelinterface.htm | |

SRR History

Object:

Minimum Interface miMod(Module)

Description of the Object:

SRR requires a minimum interface (a minimum set of fields and their behaviour), that must be
- provided by an SrrTrains module to an SrrTrains frame.

Changes of the Object:

| | |
|--|--------------------------|
| Release: | Reason for Change: |
| SRR_0030 | INT_MOD_FINISH_STEP_0030 |
| Description of the Change: | |
| The miMod(Module) interface is defined and described. | |
| Documentation: | |
| description of the lower layer interface of the SRR Framework - http://members.chello.at/christoph.valentin/lowlevelinterface.htm | |

SRR History

Object:

User Interface uiControl

Description of the Object:

The SRR Controller provides an interface to the user (frame author).

Changes of the Object:

| | |
|--|--------------------------------|
| Release: | Reason for Change: |
| SRR_0030 | INT_MOD_FINISH_STEP_0030 |
| Description of the Change: | |
| uiControl is described. | |
| Documentation: | |
| description of the lower layer interface of the SRR Framework - http://members.chello.at/christoph.valentin/lowlevelinterface.htm | |
| Release: | Reason for Change: |
| SRR_0032 | INT_MOD_TAKEMOC_UICONTROL |
| Description of the Change: | |
| Two new fields "mocRequest (MFString)" and "mocxsRequest (MFInt32)" allow to request MOC roles for a scene instance. | |
| Documentation: | |
| SrrControl.x3d | |
| Release: | Reason for Change: |
| SRR_0032 | INT_MOD_EXTERNAL_TRACER_OUTPUT |
| Description of the Change: | |
| The SRR Controller got a new field in the user interface: "traceOutput". All the tracer's output is not only printed to the Browsers console, but additionally output via this field. | |
| Documentation: | |

SRR History

Object:

User Interface

uiMod

Description of the Object:

The Module Coordinator provides an interface to the user (module author).

Changes of the Object:

| | |
|--|---|
| Release: | Reason for Change: |
| <input type="text" value="SRR_0030"/> | <input type="text" value="INT_MOD_FINISH_STEP_0030"/> |
| Description of the Change: | |
| <input type="text" value="uiMod is described."/> | |
| Documentation: | |
| description of the lower layer interface of the SRR Framework - http://members.chello.at/christoph.valentin/lowlevelinterface.htm | |

SRR History

Object:

User Interface

uiObj(SrrObject)

Description of the Object:

The SRR objects, that accompany the SRR Framework, provide user interfaces to their users (module authors, model authors, frame authors).

Changes of the Object:

| | |
|--|-------------------------------------|
| Release: | Reason for Change: |
| SRR_0030 | INT_MOD_FINISH_STEP_0030 |
| Description of the Change: | |
| uiObj(SrrObject) is described. | |
| Documentation: | |
| description of the lower layer interface of the SRR Framework - http://members.chello.at/christoph.valentin/lowlevelinterface.htm | |
| Release: | Reason for Change: |
| SRR_0031 | INT_MOD_CREATION_PROGRESS_INDICATOR |
| Description of the Change: | |
| SrrSetupPoint: The setup point provides a new output field "progressIndicator". | |
| Documentation: | |
| Release: | Reason for Change: |
| SRR_0031 | INT_MOD_NEW_SRR_OBJECT_TRIGGER |
| Description of the Change: | |
| The new SRR object SrrTrigger has been (has to be) described. | |
| Documentation: | |
| Release: | Reason for Change: |
| SRR_0031 | INT_ERR_HARMONIZE_SRR_OBJECTS_01 |
| Description of the Change: | |
| SrrDriveA: - new fields in uiObj: enabled/quasiModule | |
| Documentation: | |

SRR History

Object:

User Interface

uiObj(SrrObject)

| | |
|--|----------------------------------|
| Release: | Reason for Change: |
| SRR_0031 | INT_ERR_HARMONIZE_SRR_OBJECTS_01 |
| Description of the Change: | |
| SrrKeyContainer: - new fields in uiObj: enabled/quasiModule | |
| Documentation: | |

| | |
|--|----------------------------------|
| Release: | Reason for Change: |
| SRR_0031 | INT_ERR_HARMONIZE_SRR_OBJECTS_01 |
| Description of the Change: | |
| SrrSwitchA/SrrSwitchB: - new fields in uiObj: enabled/quasiModule | |
| Documentation: | |