

Use Case Overview (SRR)

Basic - Avatars

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.

Basic - Chat

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.

Basic - Console Interface

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).

Basic - Take/Put Keys

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.

Interlocking - Central Interlocking

not yet realized

Interlocking - Locks for Points

not yet realized

Interlocking - Signals

not yet realized

Modeling - Add Models Dynamically

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

Use Case Overview (SRR)

Modeling - Add Models Statically

A module can add static models/models statically.

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

Modeling - Build a Frame

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

Modeling - Build a Model - Carousel

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

Modeling - Build a Model - House

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.

Modeling - Build a Model - Locomotive

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

Modeling - Build a Model - Setup Point

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.

Modeling - Build a Model - Tracks

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.

Modeling - Build a Model - Turnouts

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.

Modeling - Build a Model - Wagon

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

Use Case Overview (SRR)

Modeling - Build a Module

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

Modeling - Register Module Staticly

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.

Programming - Build a Specific Vehicle Prototype

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

Programming - Build an own Track Geometry

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

Programming - Build an SRR Object

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

Trains - Basic User Interface (doze, vConst)

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

Trains - Couple by Gentle Collision

not yet realized

Trains - Crash by Hard Collision

not yet realized

Trains - Create Vehicle

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

Use Case Overview (SRR)

Trains - Decouple Manually

not yet realized

Trains - Decoupling Track

not yet realized

Trains - Delete Train

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

Trains - Derail in Curves (Speeding)

not yet realized

Trains - Derail on Points

not yet realized

Trains - Derailed Models

not yet realized

Trains - Side Collision on Points

not yet realized

Trains - Switch the Points Manually

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

Trains - Train Changes Module

not yet realized

Trains - Train is Moving

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

Use Case Overview (SRR)

Trains - Train Movers

not yet realized

Trains - User Interface (Cabs)

Vehicles can contain cabs containing cab controls in turn
