

Simulated Railroad Framework, <http://simulrr.sourceforge.net>
Synopsis: [000_Synopsis](#)

This file valid for step 0033.10
Issue Date: 2017-03-17

SRR Framework = SMUOS Framework + Train Manager Extension
=====

1 Synopsis -----

This paper serves as an introduction to the software "SMUOS Framework", which is provided by the sourceforge project "smuos", and to the software "SRR Framework", which is provided by the sourceforge project "simulrr".

This paper should be read as an add on to the paper [001_Glossary](#), which contains the description of the basic concepts of SMS and of SrrTrains.

The present paper displays only those additional concepts that might not become parts of the SMS concept.

2 Purpose -----

The purpose of the SRR Framework is to build an intermediate layer between Web3D Browsers and Railway Simulation applications.

The intermediate layer "SRR Framework" shall implement general purpose functions that are general enough to outsource them from the application layer, but that are not general enough to integrate them into the X3D standard.

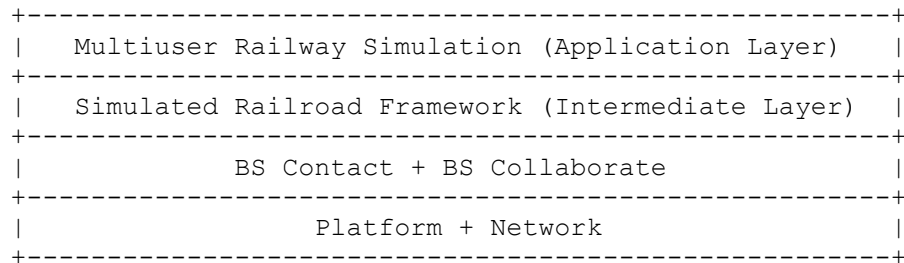


Figure 1: The SRR Framework seen as an intermediate layer

First use case we are aiming at is the "Do-it-yourself-virtual-multiplayer-model-railroad".

3 How to Use the SRR Framework

When using the SRR Framework, following concepts should be understood in addition to the concepts described in paper [001_Glossary](#).

The concepts are first listed and then the concepts with an asterisk (*) are explained in detail:

SMUOS Framework.....* experimental framework for SMS
SRR Framework.....* Simulated Railroad Framework, experimental
NS.....* Network Sensor
CS.....* Collaboration Server
Scene Instance.....* see also Personal Scene Instance (PSI)
Session.....*
Multiuser Session....*
CBSS.....* Client Based Server Software
BIMPF..... Browser Independent Multiplayer Framework
Initialization.....*
commParam..... Common Parameters
modParam..... Module Parameters
SMUOS Extensions.....*
SSCE..... Simple Scene Controller Extension
MCE..... Module Coordinator Extension
SSCEC..... Simple Scene Controller Extension Client
SSCES..... Simple Scene Controller Extension Server
BME.....* "Beamer Manager" Extension
KME.....* "Key Manager" Extension
Keys and Roles.....*
TME.....* "Train Manager" Extension

x) Experimental Framework for SMS (SMUOS Framework) -

<http://smuos.sourceforge.net>

When thinking about interoperability of models and modules, it comes to ones thoughts that standards should be used and developed as much as possible.

The chosen standard is X3D/VRML (<http://www.web3d.org>) plus its network sensor concept.

The gaps that cannot be covered by the standards, should be filled with an open source framework.

These words were true for the SRR Framework, before it was split into SRR Framework = SMUOS Framework + Train Manager Extension, now they are true for the SMUOS Framework, too.

A historic note: the SMUOS Framework was derived from the SRR Framework, by taking the "base module" of the SRR Framework

Precisely spoken, the SMUOS Framework consists of

- Simple Scene Controller (SSC)
- Module Coordinator + Module Wrapper
- MIDAS Base + Model Base
- SMS Base

The SMUOS Framework is closely accompanied by

- Example Basic MIDAS Objects
- Example Basic Extensions
- Example Extension MIDAS Objects
- Hello World Scenes (HWS)

Due to several reasons, the SMUOS Framework is currently an *EXPERIMENTAL* framework.

x) Simulated Railroad Framework - Experimental (SRR Framework) - <http://simulrr.sourceforge.net>

The SRR Framework extends the SMUOS Framework and specializes it to the use for simulated railroads

Precisely spoken, the SRR Framework consists of

- SMUOS Framework
- SRR Controller (Train Manager) *) - an SSC Extension
- Module Coordinator (Train Manager) *) - a Module Coordinator Extension
- *) not yet finished

The SRR Framework is closely accompanied by

- Addenda of SMUOS Framework
- Example SRR Objects *)
- Example Track Geometry and Models *)
- Concepts' Descriptions (the papers you're currently reading)
- *) not yet finished

Due to several reasons, the SRR Framework is currently an *EXPERIMENTAL* framework.

x) Network Sensor (NS)

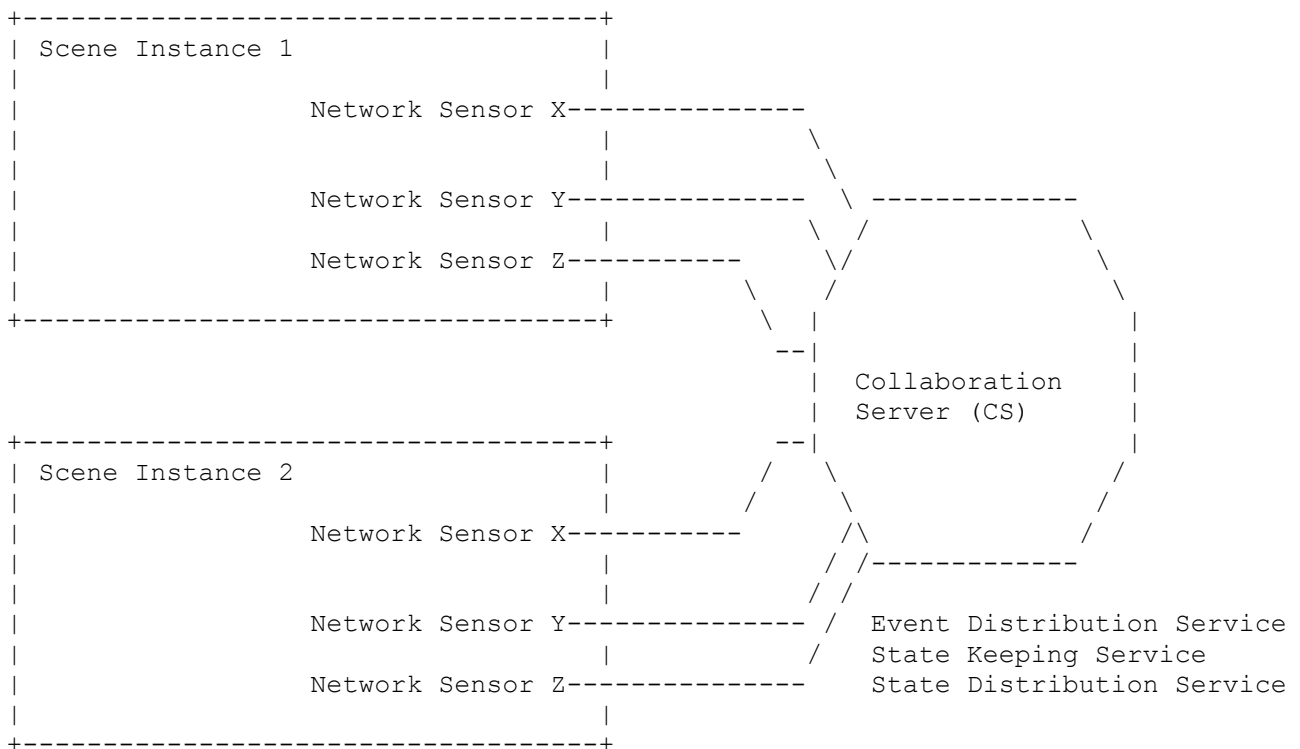
The Network Sensor is a conceptual name of an VRML/X3D node, which describes the interface of an VRML/X3D scene to the services of the collaboration server:

http://www.web3d.org/pipermail/x3d-public_web3d.org/2011-February/001423.html.

One could compare the Network Sensor with a web socket interface. However, the protocol, which is used between Scene Instances and Collaboration Server, is not yet standardized, according to our knowledge.

We will describe some terms related to network sensors, based on following figure.

The figure depicts a multiuser session with two scene instances that are connected via a collaboration server.



x) SMUOS Extensions

As well the Simple Scene Controller as the Module Coordinator may be extended by your own software (kind of plugin).
The SSC extension and Module Coordinator extension together are called "SMUOS extension".

MIDAS Objects may depend on specific SMUOS extensions, i.e. some MIDAS objects can only function, when a specific SMUOS extension is present. SMUOS Extensions may serve to handle classes of unbound models, e.g. the "Train Manager" extension will probably serve to handle the universal object classes "RailVehicles", "TrainParts" and "Trains".

x) "Beamer Manager" Extension (BME)

The SMUOS Framework (taken from <http://smuos.sourceforge.net>) is accompanied by two Example Basic Extensions and by the according Example Extension MIDAS Objects.

One of those extensions is the "Beamer Manager" Extension.

x) "Key Manager" Extension (KME)

The SMUOS Framework (taken from <http://smuos.sourceforge.net>) is accompanied by two Example Basic Extensions and by the according Example Extension MIDAS Objects.

One of those extensions is the "Key Manager" Extension.

x) Keys and Roles

SrrTrains supports assuming "roles" (e.g. engineer, shunter, ...). This is realized by so-called "keys". A key is a token, that can be owned by an avatar (user) in one scene instance or - alternatively - be contained in a so-called key container. Users can take keys from a key container and put keys into a key container.

Locks can react on carried keys (a user is e.g. a shunter, if he carries a "shunter key" and can hence access the basic user interface of a vehicle) or they can react on contained keys.

Keys and Roles are provided by the "Key Manager" Extension.

x) "Train Manager" Extension (TME) - not yet finished

The SRR Framework consists of the SMUOS Framework (taken from <http://smuos.sourceforge.net>) and of the "Train Manager" Extension. Additionally, the SRR Framework is accompanied by Example SRR Objects (i.e., by MIDAS Objects for the "Train Manager" Extension) and by the Example Track Geometry with Models.

4 How to Develop the SRR Framework

Tbd.

5 Additional Info about the SRR Framework

Tbd.