Hibernation of Advanced Railroad Trains (ArrT)

Step Three – ...... and Communicate them (aCt)

Unbound Objects (UBOs)

This Hibernation Report is a "Snapshot", it will not be updated again.

The version number has been counted up to 4.x during translation to English language in the course of step 0033.11 of the SrrTrains v0.01 project. Some minor improvements have been done.

1 Use Cases

In the SrrTrains project we are eager to create, modify and delete trains very flexibly from vehicles.

This means, we would like to configure and re-configure trains at the runtime of the simulation – not just pre-configured by config files – by supporting following procedures:

- Creation of a "one-vehicle-train" (a rail vehicle is setup on the tracks at a setup point)
- Deletion of a train
- Concatenation of two trains into one
- Splitting of one train into two

Additionally we would like to take an approach that eases the implementation of Handover at a later stage. This means, that any vehicle might be assigned to any module at any time, even if it is not the same module as the other vehicles of the train are assigned to.

2 The Approach

We try following approach:

Each train is an unbound object (UBO) and each vehicle is a UBO. Each of these UBOs is assigned to some module at any time. If

- the module the train has been assigned to, is loaded and if
- that module is active,

then **the train has got a valid train state** (position and velocity). If

- the train has got a valid train state and if
- the module the vehicle has been assigned to, is loaded and if
- that module is active

then **the vehicle gets into the state positioned** and hence becomes visible.

The vehicles are Unbound Models of the **UOC Base.SrrVehicles**, the trains are Unbound MIDAS Objects of the **UOC Base.SrrTrains** (Train Containers) that are not actually Models. This complies to the idea that models cannot contain models, but there may exists invisible "Model Containers".

Another UOC will exist for Unbound MIDAS Objects that create relations between vehicles and trains, i.e. the **UOC Base.SrrRelations**.

3 Ideas for the Implementation

3.1 The Idea of the UBO Loader

Already in step 0033.10 each programmer of an SSC Extension could use SSC Dispatchers to create UOCs (Universal Object Classes). However, the UOCs were only used to provide SSC Parameters via the console interface.

Now we would like to add functionality to the SSC Dispatchers for the creation, deletion, loading and unloading of Unbound Objects (UBOs).

Therefore the SSC Dispatcher will get a new field "uboLoader" (SFNode), where the programmer of the SSC Extension may add one instance of the **new prototype SscUboLoader.**

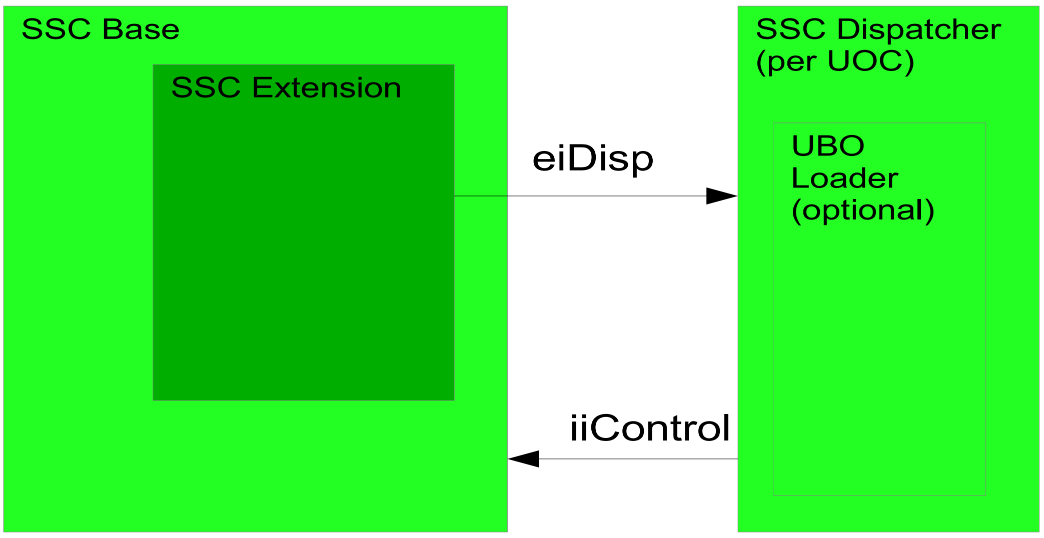

Figure 1: UBO Loader - adding functionality to the SSC Dispatcher

The UBO Loader shall support the basic procedures for the UBOs, such as

- creation and loading of a UBO
- assigning and attaching a UBO to a module
- initializing and setting the state of a UBO
- unloading and deletion of a UBO

The more sophisticated procedures of the SSC Train Manager shall then be combined from the basic proceduresof the UBO Loader. E.g. the creation of a "one-vehicle-train" could be done by

- creation of an SrrTrains UBO
- creation of an SrrVehicles UBO
- creation of an SrrRelations UBO

3.2 The "Object Type List" / "Existence State" – OTL/ES

Each UBO Loader will maintain two global states, which are related to the UOC the UBO Loader stands for:

- …a list of Object Types (i.e. URLs, the UBOs can be loaded from)
- …a list of all UBOs of this UOC