

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

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

Moving Modules (\*not yet implemented\* - cannot happen before 2020)  
=====

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

## 1 Synopsis

-----  
Moving Modules (originally called "Train Movers") are an idea to implement moving tracks, so that the following use cases can be realized.

- Trains being moved with a ferry ship
- turntables
- Rail cars carrying other rail cars (e.g. narrow gauge rail car on a standard gauge rail car)
- Rail cars carrying complete "model railroad within the model railroad"
- Model railroad within the model railroad in a house/garage

The first idea is to implement a MIDAS Object "ModuleContainer", so that one or more modules can be contained within a model.

This would enhance the simple MMF paradigm of an SMS:

(I) frame ---1:N--- module ---1:N--- model

to become more sophisticated

(II) frame ---1:N--- module ---1:N--- model ---1:N--- module ---1:N--- model  
ad infinitum

Architecture (I) supports what we call "static" modules and "dynamic" modules (now renamed to "top level" modules).

The additional modules of architecture (II) - the modules contained in a model - would be denoted "moving" modules or just "dependent" modules.

The moving modules would be contained in the model in a static way, but the model itself could be intrinsic, bound or unbound, contained in a static, dynamic or moving module.

Hence the handover of modules could be reduced to the handover of unbound models.

## 2 First Ideas

-----

E.g.

```
<!-- this code snippet is a part of a model -->
<!-- here is the first module as a part of the model -->
<Transform>
  <ProtoInstance DEF="M1" name="Module1"/>
</Transform>
<!-- here is the second module as a part of the model -->
<Transform>
  <ProtoInstance DEF="M2" name="Module2"/>
</Transform>
<!-- here follows the MIDAS Object "ModuleContainer" -->
<ProtoInstance name="ModuleContainer">
  <fieldValue name="modules">
    <Script USE="M1"/>
    <Script USE="M2"/>
  </fieldValue>
</ProtoInstance>
```

The MIDAS Object "ModuleContainer" would care for the initialization of the contained modules in dependence of the initialization of the containing model.

The MIDAS Object would have a configuration option to switch between "forward module activity from parent module to children modules" and "set module activity of children modules explicitly".

It should be possible to induce forces to the models on the children modules, depending on the parent models movements.

## 3 Naming Rules

-----

Moving Modules are always instantiated as part of a model. The model must contain a "ModuleContainer" MIDAS Object, that organizes the initialization, disabling, activation and deactivation of the moving modules.

The moving modules get the module name

```
<moduleNameMovingModule> =
  <moduleNameParent>.<objIdModuleContainer>.<localModuleNameMovingModule>
```

The tracks on the moving module get extended objIds as usual

```
<extObjIdMovingTrack> = <moduleNameMovingModule>-<objIdMovingTrack>
```

The vehicles on the moving tracks stay in the "Vehicles" UOC

```
<extObjIdVehicle> = Vehicles-<objIdVehicle>
```

DRAFT END DRAFT END DRAFT END DRAFT END DRAFT END DRAFT END DRAFT END DRAFT END