

---

Subject: Re: railML 2.3 infrastructure extension proposal - locks

Posted by [Torben Brand](#) on Sat, 04 Mar 2017 13:24:08 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Dear All,

As Christian has some more questions I will describe the relative simple case for locks on a generic (norwegian) perspective.

Locks have a daughter relationship (from the lpocks perspective) to one or more identical keys and one or more controllers. They have a mother relationship to one or more detailers and/or switches. Other mother relationships can also be present like to an electrical switch or a gate or any other object to be unlocked.

The key unlocks the lock by local personel that have to present at the lock to be able to fysically unlock it. A controller can or must release the lock before it can be unlocked. The key is located at a certain location. This can be adequately described by referring to the <ocp>. The controller can be on different levels. Either a CTC,interlocking or a local dispatcher. This can be described by referring to the <controller> controlling the lock with @type describing the type of controller. The lock type differs also according to how the derailer/switch it controls is indicated and operated (see extention suggstion towards switches).

With these additional attributes, I think, we can describe the lock in a generic manner for international purposes. The type does still have to be described as locks are very specific to national operational rules. I suggest if the lock becomes part of the official railML to give the lock types a UUID. This possibly through a value list. The easiest implementation for this would be to combine the ISO national code with the national type (wichs has to be unique). For instance: "NO:A" for the norwegian lock type A.

As an example the norwegian lock types will be generic described like this (note, this is a laymanns attempt to structure):

@Type - <ocp>@type - @keyAtRef - @releasedByControllerRef - ControllerRef@type -

Switch@remoteIndicated@remoteOperated - TVD infront of switch

NO:A - siding - same siding - the neighbour station the siding is under or CTC on remote controlled path - station - no - no - yes

NO:B - siding - the neighbour station the siding is under - released by key - NA - no - no - no

NO:C - station - same station - same station - local dispatcher without interlocking - no - no - no

NO:D - siding - both neighbour stations - released by key - NA - no -no - no

NO:E - siding/station - Key is in the lock - same controller as ocp or CTC on remote controlled path - interlocking - yes - no - yes

NO:S - station - Key is in the lock - same controller as ocp - interlocking - yes/no - Yes/no - maybe

This seems complex. But most of the information is already given. The only thing that needs to be described additionally to describe a unique lock type is where the key is located at and what controller releases the key/lock.

To the question of number of relationships.

A key is usually at one place. But in one type there is a key at both neighbour stations. So n @KeyRef.

A lock is only controlled by one controller (I think) So 1 @releaseAtControllerRef

A lock can control one or more detailers/switches. Mostly, but not always through a key collection lock. I suggest to omit this and ref directly to the switches/derailles on the level of description.

Further details are probably to be availed in the interlocking schema.

---