
Subject: railML 2.3 infrastructure extension proposal - locks
Posted by [Torben Brand](#) on Tue, 20 Dec 2016 17:29:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear railML infrastructure forum,

This posting contains the discussion to an extension towards the
ocsElements

Locks are simple physical lineside elements that can only have two states: locked or unlocked.
Locks reference to which objects they lock. Their relationship will later be defined in the
interlocking scheme.

The element <ocsElements> is extended with the new element <locks> with sub element <lock>
with attributes @objectRef and @type [datatype: enumeration] with 5 Norwegian presets, the
value "ERTMS" and "other:".

Subject: Re: railML 2.3 infrastructure extension proposal - locks
Posted by [christian.rahmig](#) on Mon, 02 Jan 2017 16:30:40 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Torben,

Am 20.12.2016 um 18:29 schrieb Torben Brand:

> [...]
> ocsElements
> Locks are simple physical lineside elements that can only
> have two states: locked or unlocked. Locks reference to
> which objects they lock. Their relationship will later be
> defined in the interlocking scheme. The element <ocsElements> is
> extended with the new element
> <locks> with sub element <lock> with attributes @objectRef
> and @type [datatype: enumeration] with 5 Norwegian presets,
> the value "ERTMS" and "other:".

The lock (de: "Schlüsselsperre") is a device that is relevant for
interlocking purposes. At the same time, it is a physical element and
thus should be part of the railway infrastructure model. Currently,
modelling a lock with railML 2.3 would look like this:

```
<track ...>
...
<ocsElements>
  <trainProtectionElements>
    <trainProtectionElement id="tpe01" pos="123.4" description="lock">
    </trainProtectionElement>
  </trainProtectionElements>
</ocsElements>
</track>
```

Your approach to introduce a new element <lock> is more straightforward and it fits better considering that a lock is not a train protection element like an Indusi magnet. If there are use cases requiring the specific definition of locks, I appreciate having them included in the railML infrastructure model. Which objects are referenced by <lock>@objectRef? What are the 5 Norwegian presets for the attribute <lock>@type?

Best regards
Christian

--

Christian Rahmig - Infrastructure scheme coordinator
railML.org (Registry of Associations: VR 5750)
Phone Coordinator: +49 173 2714509; railML.org: +49 351 47582911
Altplauen 19h; 01187 Dresden; Germany www.railml.org

Subject: Re: railML 2.3 infrastructure extension proposal - locks
Posted by [Torben Brand](#) on Fri, 24 Feb 2017 14:35:47 GMT
[View Forum Message](#) <> [Reply to Message](#)

My reply:

The use case is for signal drawings. Locks are physical logical elements that can lock switches and derailleurs in certain positions. These elements need to be referenced. The derailleurs are obviously train protection elements. Switches can be if they act as protective/deflective switches. But it's not the lock that forms the train protection element. Thus I suggest creating them as separate elements.

Correction there are 6 types of locks in Norway. The 6 Norwegian lock types are: type: A,B,C,D, E and S. For more information, see:

[http://orv.jbv.no/orv/doku.php?id=tjn:kap_12:betjening_av_kontrollaser_og_samlaser&s\[\]=l%C3%A5s](http://orv.jbv.no/orv/doku.php?id=tjn:kap_12:betjening_av_kontrollaser_og_samlaser&s[]=l%C3%A5s)

Subject: Re: railML 2.3 infrastructure extension proposal - locks
Posted by [christian.rahmig](#) on Fri, 03 Mar 2017 14:48:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Torben,

thank you very much for your feedback and answers. So, I created a Trac ticket for this issue, which can be found in [1]. If you think, that there are still some points missing, please let us know.

Am 24.02.2017 um 15:35 schrieb Torben Brand:

- > My reply:
- > The use case is for signal drawings. Locks are physical
- > logical elements that can lock switches and derailleurs in
- > certain positions. These elements need to be referenced. The
- > derailleurs are obviously train protection elements. Switches
- > can be if they act as protective/deflective switches. But
- > it's not the lock that forms the train protection element.
- > Thus I suggest creating them as separate elements. Correction there are
- > 6 types of locks in Norway. The 6
- > Norwegian lock types are: type: A,B,C,D, E and S.

Two short questions:

* Are there always 2 elements linked together via a lock or do you know examples with more elements?

* Are the Norwegian lock types very country-specific or are they known in other countries, too?

[1] <https://trac.railml.org/ticket/305>

Best regards
Christian

--

Christian Rahmig - Infrastructure scheme coordinator

railML.org (Registry of Associations: VR 5750)

Phone Coordinator: +49 173 2714509; railML.org: +49 351 47582911

Altplauen 19h; 01187 Dresden; Germany www.railml.org

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 llocks 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 controlls 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 (which 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 in front 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/derails on the level of description.

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

Subject: Re: railML 2.3 infrastructure extension proposal - locks

Posted by [christian.rahmig](#) on Fri, 31 Aug 2018 08:16:50 GMT

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

Dear all,

Am 03.03.2017 um 15:48 schrieb Christian Rahmig:

> [...]

> thank you very much for your feedback and answers. So, I created a Trac

> ticket for this issue, which can be found in [1]. If you think, that

> there are still some points missing, please let us know.

> [...]

>

> [1] <https://trac.railml.org/ticket/305>

following our latest telephone discussion on that topic, I adapted the Trac ticket #305 [1]. For railML 2.4, a very reduced first step solution of <lock> is planned. A <lock> element shall have a <geoCoord>, attributes @pos and @absPos and a generic "any" (element and attribute) anchor for further specific extensions.

Best regards
Christian

--

Christian Rahmig - Infrastructure scheme coordinator
railML.org (Registry of Associations: VR 5750)
Phone Coordinator: +49 173 2714509; railML.org: +49 351 47582911
Altplauen 19h; 01187 Dresden; Germany www.railml.org
