

---

Subject: Re: railML 2.3 infrastructure extension proposal switch / crossing

Posted by [Torben Brand](#) on Fri, 24 Feb 2017 14:57:01 GMT

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

---

Christian Rahmig wrote:

This attribute [`@NO:levelOfControl`] is misleading as in the end all switches and crossings are somehow controlled (even if it is manually controlled). Can you please clarify this requirement by providing some more information about the use case behind?

My answer:

True, all switches must be controlled. So the use of the word "control" and "controlled" are misleading. I will use other terms here. What I mean is that we need to define if the switch position is indicated remotely. This usually in the interlocking, but it could also be in simpler none interlocking devices like a switch tableau. Further we need to know if the switch is operated at the switch or remotely. Remotely from a central operator, usually by a dispatcher through the interlocking, but also simpler by a local central operator/ shunting personnel through a switch tableau. The combination of the two states form three possible combinations:

- indicated in interlocking/tableau and remote operated (by same)
- indicated in interlocking but operated at the switch
- not indicated in interlocking and operated at the switch.

The fourth combination, remote operated but not indicated is not logical.

Alternatively, we could just make it clean and use:

`<switch/crossing>@NO:positionRemoteIndicated value:boolean(yes/no)`

`<switch/crossing>@NO:positionRemoteOperated value:boolean(yes/no)`

In fact, all

infrastructure elements that can be incorporated in an interlocking controlling, may have a reference to a controller. Alternatively (from an interlocking point of view), the controller/interlocking shall reference all the infrastructure elements that it controls. We should decide for one direction of referencing, but not implement both.

Agreed. The usual workflow in Norway is: is the switch controlled and how. Not: Which switches does this controller operate? Furthermore, the `controllerRef` is already implemented in railML 2.3 on the switch. So we recommend this direction.

However, instead of a "typical" throw time, which may vary a lot, I prefer defining a `"minimumThrowTime"` and/or a `"maximumThrowTime"`.

Your suggested attributes `@minimumThrowTime` and `@maximumThrowTime` are needed in the interlocking use case and will be modelled in the interlocking schema.

For the capacity use case we need the average throw time. Thus my suggestion for element name was `"averageThrowTime"`. But the interlocking coordinator preferred `"typicalThrowTime"`. Obedient as I am, I bow to the coordinators... ;-) But this maybe indicates that the suggested name is misleading.

Bob writes: "attribute typicalThrowTime for movable elements is now defined: «typical throw time is the average time it takes between the moment the IL receives the call and the element reaches the new position. Point throwing adds a delay to route setting that is of great interest to the use case simulation. For this purpose, we add an attribute typicalThrowTime that allows capacity planners to estimate the influence of slow throwing points on train traffic. Note that this excludes OCS processing time and communication between OCS and IL."

---