Subject: railML 2.3 infrastructure extension proposal - controller
Posted by Torben Brand on Tue, 20 Dec 2016 17:24:29 GMT
View Forum Message <> Reply to Message

Dear railML infrastructure forum,
This posting contains the discussion to an extension towards the
controller
The controller (DE:Stellwerk) needs to be defined on a macroscopic level for what type and system is used. This as to give the capacity planner generic values of some capacity related values of the stations features.
Thus I have added the three new attributes:,@NO:model, @NO:type, @NO:technologyType and @NO:swVersion.
@NO:model: [datatype:string] Defines the model/system used. Examples are: SIMIS-C,Thales L-90, NSB-77, NSI-63,...
@NO:type [datatype:enumeration] Defines the type of controller on a general level. This is predefined with three Norwegian national presets, the value "none" and "other:" The presets are "NO:plussStasjon" (English: full interlocking), "NO:enkeltSikringsanlegg" (English: simplified interlocking) and "NO:enkeltInnkjørsignal" (English:simplified entry signal).
@NO:technologyType [datatype: enumeration] The predefined values are: "electric", "electromechanic", "electronic", "mechanic"
PS. The terms @type, @model, @system, @mode need to be defined more clearly in railML in general to be consistent throughout.

Subject: Re: railML 2.3 infrastructure extension proposal - controller
Posted by christian.rahmig on Mon, 02 Jan 2017 16:28:57 GMT
View Forum Message <> Reply to Message

Dear Torben,

*** This post has been cross-posted in infrastructure and interlocking
forum. Please only reply in the infrastructure forum. Thank you. ***

Am 20.12.2016 um 18:24 schrieb Torben Brand:
> [...]
> controller
> The controller (DE:Stellwerk) needs to be defined on a
> macroscopic level for what type and system is used. This as
> to give the capacity planner generic values of some capacity
> related values of the stations features. Thus I have added the three
> new attributes:,@NO:model,
> @NO:type, @NO:technologyType and @NO:swVersion.
> @NO:model: [datatype:string] Defines the model/system used.
> Examples are: SIMIS-C,Thales L-90, NSB-77, NSI-63,...
> @NO:type [datatype:enumeration] Defines the type of
> controller on a general level. This is predefined with three
> Norwegian national presets, the value "none" and "other:"

> The presets are "NO:plussStasjon" (English: full
> interlocking), "NO:enkeltSikringsanlegg" (English:
> simplified interlocking) and "NO:enkeltInnkjørsignal"
> (English:simplified entry signal).
> @NO:technologyType [datatype: enumeration] The predefined
> values are: "electric", "electromechanic", "electronic",
> "mechanic"

Until railML version 2.3 the <controller> element has been just a placeholder element, which indicates that the railway infrastructure is controlled from some kind of interlocking. All the detailed features of the controller that describe its functionality etc. are part of the upcoming interlocking schema. So, let me comment on your proposal from an infrastructure point of view:

<controller>@NO:model
I agree with putting the product (interlocking) name here. In order to avoid misspelling I prefer implementing an enumeration here or - if there would be too many entries - to use a codelist as it has been done for the TrainProtectionSystem. A codelist - though released and maintained by railML.org - is not an essential part of the schema and may change (new entries) on short notice. Thus, a codelist ist more flexible than an enumeration value. In any case, for railML v3 the attribute @model should be part of the new interlocking schema.

<controller>@NO:type
The idea of this parameter is to provide some classification of interlockings/controllers regarding their complexity or responsibility. I think that this is useful as other countries and railways do the same in order to create some hierarchy of their interlocking network. For a later implementation within the railML schema, I suggest to find a generic classification that is compatible to the different national class structures. Is "none" a useful entry? In any case, for railML v3 the attribute @type should be part of the interlocking schema.

<controller>@NO:technologyType
The current railML version 2.3 already contains an enumeration data type tInterlockingTypes, which is used by the parameter <ocp><propEquipment><summary>@signalBox, and which provides the following values:
* none
* mechanical
* electro-mechanical
* electrical
I suggest to recycle this enumeration data type and to use it for the attribute <controller>@technologyType. In any case, for railML v3 the attribute @technologyType should be part of the interlocking schema.

<controller>@NO:swVersion
Is that needed? Please provide some more explanation.

> PS. The terms @type, @model, @system, @mode need to be
> defined more clearly in railML in general to be consistent
> throughout.

I agree that railML should provide clear definitions for the content of the attributes @type, @model, @system, @kind and @mode. However, we will not change it with railML v2.x, but only with railML v3. In the meantime, we will try to bring more clarity in the documentation of these parameters in the wiki.

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 - controller
Posted by Torben Brand on Fri, 24 Feb 2017 14:03:12 GMT

View Forum Message <> Reply to Message

Christian Rahmig wrote:
Until railML version 2.3 the <controller> element has been just a placeholder element, which indicates that the railway infrastructure is controlled from some kind of interlocking. All the detailed features of the controller that describe its functionality etc. are part of the upcoming interlocking schema.

My reply:
Level of description
My suggestion is to place the <controller> element in-between a placeholder and a full interlocking description in its description level. The purpose is to have a generic macroscopic description of the controller for operational purposes.
Futureproof names
As many suggested element terms will also be used in the upcoming interlocking schema. I have coordinated the name use with the interlocking coordinator, Bob Jansen.

<controller>@NO:model
I agree with putting the product (interlocking) name here. In order to avoid misspelling I prefer implementing an enumeration here or - if there would be too many entries - to use a codelist as it has been done

for the TrainProtectionSystem. A codelist - though released and maintained by railML.org - is not an essential part of the schema and may change (new entries) on short notice. Thus, a codelist ist more flexible than an enumeration value. In any case, for railML v3 the attribute @model should be part of the new interlocking schema.

Codelists
I agree that using a codelist is wise to avoid misspelling and increase efficiency. But it also complicates the use. So I agree that it should be used in RailML3, but for railML 2 a free text data type should suffice.
I suggest to publish a list in wiki.railml.org that lists and links to all codelists used in railML.

<controller>@NO:type
The idea of this parameter is to provide some classification of interlockings/controllers regarding their complexity or responsibility.
I think that this is useful as other countries and railways do the same in order to create some hierarchy of their interlocking network. For a later implementation within the railML schema, I suggest to find a generic classification that is compatible to the different national class structures. Is "none" a useful entry? In any case, for railML v3 the attribute @type should be part of the interlocking schema.

<controller>@NO:type
I agree that in the future the interlocking schema group should find generic values for Controller:Type. But I am uncertain that this is possible. This as the meaning for <controller>@NO:type is which type of controller is used from an operational perspective. We refer to the operational rules [in Norway http://orv.jbv.no/orv/doku.php?id=tjn:start]. These differ according to the controller type. As the operational rules differ on national level, we suggest to just use the Norwegian values (in Norwegian) for now. If no common usage can be found we should maybe keep national values in the upcoming standard. For instance, with a country code first following the type. Maybe also with a reference to the operational rule.

Value "none"
There should be a general discussion towards the use of the value "none". Today not writing a value indicates that you do not have that functionality or that you just have not mapped it. Placing a "none" value indicates that you have mapped the value and it does not exist.

<controller>@NO:technologyType
The current railML version 2.3 already contains an enumeration data type tInterlockingTypes, which is used by the parameter <ocp><propEquipment><summary>@signalBox, and which provides the following values:
* none
* mechanical
* electro-mechanical
* electrical
I suggest to recycle this enumeration data type and to use it for the attribute <controller>@technologyType. In any case, for railML v3 the

attribute @technologyType should be part of the interlocking schema.

<controller>@NO:technologyType
I agree to recycle previous enumeration values.

<controller>@NO:swVersion
Is that needed? Please provide some more explanation.

<controller>@NO:swVersion
This element was requested by Bob Jansen. It makes sense for me to have it her on the operational description level as the controller's software version is important for interoperability issues.

I agree that railML should provide clear definitions for the content of
the attributes @type, @model, @system, @kind and @mode. However, we will
not change it with railML v2.x, but only with railML v3. In the
meantime, we will try to bring more clarity in the documentation of
these parameters in the wiki.

Clarity
I applaud more documentation in railML 2 and (spring) cleaning in railML3.

New issue: The documentation would have to be clear about the interface towards the existing element <locallyControlledArea>. For instance, one locally controlled area can have one or more controllers. Track should not be referenced in both at the same time.

Subject: Re: railML 2.3 infrastructure extension proposal - controller
Posted by christian.rahmig on Mon, 27 Feb 2017 11:10:08 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 me know.

[1] https://trac.railml.org/ticket/304

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 - controller
Posted by Torben Brand on Sat, 14 Jul 2018 15:12:09 GMT
View Forum Message <> Reply to Message

controller@nor:ocpRef
We need to map the ocp to the controller from the controller's side. So we will add a Norwegian extension optional attribute nor:ocpRef to the <controller>. In Norway the (interlocking) controller is under the ocp. An ocp can have multiple controllers, but a controller (at least in Norway) can have only one ocp.

Subject: Re: railML 2.3 infrastructure extension proposal - controller
Posted by christian.rahmig on Thu, 16 Aug 2018 11:03:01 GMT
View Forum Message <> Reply to Message

Dear Torben,
dear all,

Am 14.07.2018 um 17:12 schrieb Torben Brand:
> controller@nor:ocpRef We need to map the ocp to the controller from the
> controller's side. So we will add a Norwegian extension
> optional attribute nor:ocpRef to the <controller>. In Norway
> the (interlocking) controller is under the ocp. An ocp can
> have multiple controllers, but a controller (at least in
> Norway) can have only one ocp.

I adapted the ticket #304 accordingly (adding reference from OCP to Controller). The changes may be implemented with railML 2.4.

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 - controller
Posted by christian.rahmig on Thu, 16 Aug 2018 13:52:48 GMT
View Forum Message <> Reply to Message

Dear Torben,
dear all,

Am 27.02.2017 um 12:10 schrieb Christian Rahmig:

> [...] 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 me know.
>
> [1] https://trac.railml.org/ticket/304

I just found out that current proposal misses the possibility to define a hierarchy of controllers via a new attribute @parentControllerRef. Therefore, I adapted the Trac ticket description [1]. Any further issues missing?

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