Subject: [railML3] Specific commands and indications of signal box user interface
Posted by Heidrun Jost on Thu, 04 Jun 2020 18:18:39 GMT
View Forum Message <> Reply to Message

Dear all,

in railML 3.2 we need an extension for market specific commands and indications to define the catalog of commands and indications.

Commands and Indications control and show the interlocking functionality from the user interface of a signal box (today a screen only) point of view. It is a standard interface of interlockings.

Therefore we propose to have a new element in <specificIMs> in parallel to the existing element <usesType> and <ownsSetsOfAssets>. The proposal for the name is <usesOperations>. That contains all elements of the new defined elements <commands> and <indications>.

Here is a XML example for the usage of the proposed commands and indications.

```
<usesOperations>
   <commands>
    <command id="CMD1" acronym="HTV">
       <designator register="commandList:HMI" entry="SetRoute"/>
       <name name="Set main route" language="en"/>
       <name name="Sikre hovedtogvei" language="no"/>
       <scope scope="interlockingType" validFor="IL1_Type" alias="0x11"/>
       <scope scope="interlockingType" validFor="IL2_Type"/>
       <classification class="securityLevel" value="DEFAULT"/>
     </command>
   </commands>
   <indications>
     <indication id="IND1" acronym="OCCUPIED">
       <designator register="il1_Type:indicationList:TRACK"
entry="OccupiedTrack"/>
       <designator register="il2_Type:indicationList:TRAIL"
entry="OccupiedTrail"/>
       <name name="Track occupied" language="en"/>
       <scope scope="interlockingType" validFor="il1_Type "
alias="00100111"/>
      <scope scope="interlockingType" validFor="il2_Type "/>
       <classification class="..." value="..."/>
     </indication>
   </indications>
</usesOperations>
```

The <designator> will be used for identification in different contexts.
The <name> will be used for external identification (e.g. HMI).
The <scope> element supports the assignment of commands/indications to different interface types.
The attribute @scope defines the interface classification, dependent e.g. from interlocking or other

adjacent system and is an arbitrary string.
The attribute @validFor contains the name of the interface type and is an arbitrary string.
The attribute @alias allows the definition of an interface type specific alias of the command/indication and is an arbitrary string.

In most cases we have to handle in our project the same commands for different interface types (e.g. interlocking type). But there are exceptions. Some commands are valid only for one interlocking type or adjacent system.

The <classification> will be used for classification in the common way (e.g. for security levels of commands).

The defined commands and indications should be usable as references for each element, that is provided under <assetsForIL> element and also for <implementsElementGroup>.

To use the defined commands and indications we propose to add new elements <hasCommands> and <hasIndications> for each element as mentioned above, see this XML example:

```
<signalIL id="IL_sig1" isVirtual="false" function="main">
    <designator register="il1_Type:signalList" entry="sig1"/>
    <hasCommand ref="CMD1" entityCode="8.7"/>
    <hasCommand ref="CMD2" entityCode="14.2"/>
    <hasIndication ref="IND1" entityCode="13.1"/>
    <hasindication ref="IND2" entityCode="13.2"/>
    <refersTo ref="IS_sig1"/>
</signalIL>
```

Any comments or suggestion for railML 3.2?

Best regards,
--
Heidrun Jost
Data Manager TMS Norway
Transportation Systems
Thales Deutschland GmbH

Phone: +49 (0) 30 688306 423
Schützenstr. 25 10117 Berlin Germany

Subject: Re: [railML3] Specific commands and indications of signal box user interface
Posted by Joerg von Lingen on Sat, 13 Jun 2020 12:52:29 GMT
View Forum Message <> Reply to Message

Thanks Heidrun for your input.

I support the idea of having the possibility to define the commands and indications used for control of interlocking. However, I am sceptic on the use of arbitrary strings for specific definitions. This may make data interchange more complex and may cause failures due to different writings.

Best regards,
Joerg v. Lingen - Rollingstock Coordinator