Subject: Re: [railML3]: keyLocksIL and ownsLogicalDevice
Posted by Joerg von Lingen on Sun, 01 Nov 2020 06:18:03 GMT
View Forum Message <> Reply to Message

Dear all,

the element <keylockIL> was until this summer not in the focus of any user. Thus
we didn't discover the failure in the schema file before. This was corrected in
file https://www.railml.org/schemas/3.1/interlocking3.xsd.

During the refactoring to change the way of polymorphism probably one line did
remain. It was line 2017:
<xs:element name="ownsLogicalDevice" type="rail3:LogicalDevice" minOccurs="1"
maxOccurs="unbounded"/>

With this line in the schema an element of an abstract data type is required,
which is not really possible.

In case you are not validating against the online schema you shall check to have
the correct schema (railML-3.1.zip) downloaded from railML.org.

Best regards,
Joerg v. Lingen - Rollingstock Coordinator

Am 20.10.2020 um 13:28 schrieb Dominik Looser:
> Dear all,
>
> in RailML3.1, the interlocking complexType keyLocksIL not
> only has a child keyLockIL, but also another child
> ownsLogicalDevice:
> <xs:element name="ownsLogicalDevice"
> type="rail3:LogicalDevice" minOccurs="1"
> maxOccurs="unbounded"/>
>
> We have tried several possibilites on what to fill into this
> ownsLogicalDevice child, e.g.:
>
>
> <keyLocksIL>
>   <keyLockIL id="kloIL8">
>   <refersTo ref="klo8"/>
>   </keyLockIL>
>   <ownsLogicalDevice id="old_temp_id"/>
> </keyLocksIL>
>
>
> Since all other attributes and subelements of

> ownsLogicalDevice are optional, this should be sufficient
> from our understanding, but the xml-validation always fails
> with:
>
> 269:46 cvc-type.2: The type definition cannot be abstract
> for element ownsLogicalDevice.
>
> We would be thankful if someone could post a valid example
> for keyLocksIL.
>
> Thank you and best regards,
> Dominik Looser
> trafit solutions gmbh