

---

Subject: New area types for railML 3.3

Posted by [Terje Nordal](#) on Wed, 07 Feb 2024 13:23:47 GMT

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

---

Dear railML experts,

Referring to previous posts on the forum discussing avalanche and derailment detection ( [https://www.railml.org/forum/index.php?t=msg&goto=3006&amp;#msg\\_3006](https://www.railml.org/forum/index.php?t=msg&goto=3006&amp;#msg_3006)) and suggestions of new Emergency Stop Areas (ESA) for railML 3.3 ( [https://www.railml.org/forum/index.php?t=rview&goto=3167 &th=927](https://www.railml.org/forum/index.php?t=rview&goto=3167&th=927)) Bane NOR have had some internal discussions, accompanied by the Norwegian Jernbanedirektoratet, on suggestions for implementation.

We agree to the suggestion of letting ESA use a similar definition to workZone or localOperationArea. We also suggest an additional area type to allow definitions of Danger Areas (IL:dangerArea), to allow definitions of avalanche areas. Both areas need the ability to reference a detector for activation of the area. For the differentiation between conditional and unconditional ESA we suggest having a isConditional(0..\*) (true/false).

For Bane NOR the typical use cases would be:

dangerArea: An area that may be activated by a sensor (typically one or more avalanche detectors), resulting in the trains typically having to run slowly in case of dangerous conditions ahead.

(unconditional)emergencyStopArea (isConditional=false): An area that may be activated either by a dispatcher or by a sensor (typically a derailment indicator), resulting in all trains inside and outside the area having to stop.

(conditional)emergencyStopArea (isConditional=true): An area that may be activated by a sensor (for instance a fire alarm in a tunnel, to avoid trains outside the tunnel entering), resulting in trains already inside the area (and close to entering) being allowed to continue while trains that are able to stop before entering will be stopped.

These three scenarios are pretty similar, but would affect the interlocking (train movements authorities) differently. That is the reasoning for needing the differentiation.

Our suggestion is therefore the following area definitions:

- IL:dangerArea using the following children: assetName (0..\*), belongsToOperationalPoint (0..1), designator (0..\*), detectorInState (0..\*), hasCommand (0..\*), hasIndication (0..\*), isLimitedBy (0..\*), signalWithAspect (0..\*), trackAssetInArea (0..\*).

- IL:emergencyStopArea using the following children: assetName (0..\*), belongsToOperationalPoint (0..1), designator (0..\*), detectorInState (0..\*), hasCommand (0..\*), hasIndication (0..\*), isLimitedBy (0..\*), isConditional(0..\*)1, signalWithAspect (0..\*), trackAssetInArea (0..\*).

Best regards  
Terje Nordal

---

---

Subject: Re: New area types for railML 3.3

Posted by [Jörg von Lingen](#) on Sun, 25 Feb 2024 07:52:24 GMT

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

---

Dear all,

I have made a draft of the proposed areas in the model derived from RestrictedArea.

```
<xs:complexType name="DangerArea">
  <xs:complexContent>
    <xs:extension base="rail3:RestrictedArea">
      <xs:sequence>
        <xs:element name="detectorInState" type="rail3:DetectorInState"
          minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="signalWithAspect" type="rail3:SignalWithAspect"
          minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

```
<xs:complexType name="EmergencyStopArea">
  <xs:complexContent>
    <xs:extension base="rail3:RestrictedArea">
      <xs:sequence>
        <xs:element name="detectorInState" type="rail3:DetectorInState"
          minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="signalWithAspect" type="rail3:SignalWithAspect"
          minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="keyLockInState" type="rail3:KeyLockInState" minOccurs="0"
          maxOccurs="unbounded"/>
        <xs:element name="levelCrossingInState" type="rail3:LevelCrossingInState"
          minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="isConditional" use="optional" type="xs:boolean">
      </xs:attribute>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

In addition to the proposal from BaneNor I have added keylock and level crossing in the emergency stop area as proposed in <https://www.railml.org/forum/index.php?t=rview&goto=3167&th=927>.

There is one question remaining: Do we need to refer to the triggering detector for the conditional ESA?

Best regards,  
Joerg v. Lingen - Interlocking Coordinator

---

---

Subject: Re: New area types for railML 3.3  
Posted by [Terje Nordal](#) on Wed, 13 Mar 2024 12:38:40 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Dear Jörg,

This sounds very good. Adding `keyLockInState` and `levelCrossingInState` makes sense based on the referenced forum post from Heidrun.

On the topic of referring to a trigger detector for CESA, I think it depends on what railML want to include in the standard. As I mentioned, the detector in this case would typically be a bit outside of the railway system itself, such as a fire alarm. Since that configuration is possible in our signalling system, directly effecting the state of an Emergency Stopping Area I think it also makes sense to add the possibility to model it in railML. However, are the current definitions in `IL:detectorInState` sufficient to do so? How would a fire alarm (or a similar detector) fit into that definition?

Best regards,  
Terje Nordal

---