

---

Subject: [railML 2] ocp modeling

Posted by [Larissa Zhuchyi](#) on Wed, 05 Apr 2023 16:18:22 GMT

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

---

Dear all

In this example, railway station X is described that has three yards (parts of the station). Yard X.Y as part of station X is used only for freight trains, yard X.Z is used by passenger trains and yard X.J is only operational (no freight nor passenger traffic). Additionally, during the winter time, yard X.Y is used for passenger trains as well. In summer the whole station is closed.

Station X has three spatial parts (yards) X.Y, X.Z, X.J. Yard X.Y has a temporal part as soon as its properties change over time. Please, communicate your ideas on questions (1, 2) to clarify the principles of modelling properties that change over time using railML 2:

(1) Does yard Y1 referring to yard Y imply that yard Y is always "freight" and "passenger" only in winter?

(2) Is there any other way to represent properties that change over time in railML 2?

The source code in railML 2 is as follows:

```
<operationControlPoints>
  <ocp id="X" name="X" type="operationalName">
    <propOperational operationalType="station"/>
    <propOther>
      <states>
        <state status="disabled"
          startDateTime="2023-06-01T08:00:00"
          endDateTime="2023-08-31T18:00:00"/>
      </states>
    </propOther>
  </ocp>
  <ocp id="Y" name="X.Y" type="operationalName" parentOcpRef="X">
    <propOperational trafficType="freight"/>
  </ocp>
  <ocp id="Z" name="X.Z" type="operationalName" parentOcpRef="X">
    <propOperational trafficType="passenger"/>
  </ocp>
  <ocp id="J" name="X.J" type="operationalName" parentOcpRef="X">
    <propOperational trafficType="operational"/>
  </ocp>
  <ocp id="Y1" name="X.Y" parentOcpRef="Y">
    <propOperational trafficType="passenger"/>
    <propOther>
      <states>
        <state status="operational"
          startDateTime="2023-01-01T08:00:00"
          endDateTime="2023-02-28T18:00:00"/>
      </states>
    </propOther>
  </ocp>
</operationControlPoints>
```

```
</states>
  </propOther>
</ocp>
</operationControlPoints>
```

Here, winter, spring and summer are represented by infrastructure/operationControlPoints/ocp/propOther/states/state element.

Yards (ocp) are the spatial part of the station (ocp).

Freight, passenger and staff aspects are represented by infrastructure/operationControlPoints/ocp/propOperational/@trafficType attribute.

"Parts" are linked with the "whole" using the @parentOcpRef attribute.

See the figure below.

## File Attachments

1) [2023-04-04 \(2\).png](#), downloaded 241 times

---

---

Subject: Re: [railML 2] ocp modeling

Posted by [Thomas Nygreen](#) on Thu, 11 May 2023 14:35:57 GMT

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

---

Dear Larissa,

Your second question about how to model properties that change over time is very interesting and challenging. The design of <state> (and similarly <infrastructureState> and <elementState> in railML 3) is more geared towards modelling the existence or availability for an element as a whole. Since we can only apply the state to the entire element, not to any single attributes or subelements, it is not really adequate for modelling that properties of an element change over time. To my knowledge, we do not have an established best practice for how to deal with this. From earlier discussions, I know at least two strategies:

A: Only present a snapshot, valid at one specific date. To reflect changing properties, you would have to exchange several files, each representing a given date or time interval.

B: Duplicate the relevant elements and give them states to make sure at most one of them exists at any time. To know that they are actually the same element, you would have to look at <designator>s and/or @code. In railML 3 this would raise a question of whether two elements can have the same UUID if they represent the same object at different times. Other elements that refer to a duplicated element, may also have to be duplicated. For instance, a <trainPart> referring to an <ocp> would turn into two <trainPart>s with separate operating periods, each referring to one of the versions of the <ocp>.

Seeing your example, I think B can be improved in a slightly different way:

C: As B, but you group the copies using a parent element (which contains all the unchanging properties) and @parentOcpRef and <state>s with disjoint time periods on the copies. For this to work for other element types in railML 2, we would have to include a @belongsToParent as in railML 3, or say that such use cases are best solved with railML 3.

Your post also raises a more practical third question of how to interpret the state outside of the given time intervals. According to the Wiki documentation for <state>, when "no specific <state> is defined for the parent infrastructure element this infrastructure element remains "valid" without any further constraints." [1] But when there is at least one <state> defined, how should we interpret the state outside of the given time interval(s), where no state is explicitly given?

We should also document (and make sure we agree) how <state>s are inherited, e.g. from <infrastructure> to any infrastructure element or from <track> to any <trackElements> or <ocsElements>.

Going back to your first question: In your example, X is disabled in June, July and August of 2023 (and so are Y, Z and J by inheritance). Its state during the rest of 2023 or in previous or later years is not given explicitly. Since Y1 has its own <propOther> any inheritance of the states of X or Y is broken, so all we know is that it is available in January and February of 2023, and with no explicit status for other dates. What safe assumptions can we make (and leave to a computer to make) for other dates?

Furthermore, on your specific example: I would interpret Y1 as being yet another spatial part, inside Y, which is only for passenger traffic and is open during the winter (while its state at other times is not given, as I comment above). Maybe using the C strategy, it could be something like:

```
<infrastructure>
  <operationControlPoints>
    <ocp id="X" name="X" type="operationalName">
      <propOperational operationalType="station"/>
      <propOther>
        <states>
          <state disabled="true"
            status="disabled"
            startDateTime="2023-06-01T08:00:00"
            endDateTime="2023-08-31T18:00:00"/>
        </states>
      </propOther>
    </ocp>
    <ocp id="Y" name="X.Y" type="operationalName" parentOcpRef="X">
      <!-- Inherits propOperational and propOther from X -->
    </ocp>
    <ocp id="Z" name="X.Z" type="operationalName" parentOcpRef="X">
      <propOperational trafficType="passenger"/>
    </ocp>
  </operationControlPoints>
</infrastructure>
```

```
<!-- Inherits propOther from X -->
</ocp>
<ocp id="J" name="X.J" type="operationalName" parentOcpRef="X">
  <propOperational trafficType="operational"/>
  <!-- Inherits propOther from X -->
</ocp>
```

```
<!-- passenger and freight version of Y, operational in winter -->
<ocp id="Y1" parentOcpRef="Y">
  <!-- Inherits propOperational from X -->
  <!-- no specific traffic type, implying no restriction -->
  <propOther>
    <states>
      <state disabled="true"
        status="disabled"
        startDateTime="2023-01-01T00:00:00"
        endDateTime="2023-01-01T08:00:00"/>
      <state disabled="false"
        status="operational"
        startDateTime="2023-01-01T08:00:00"
        endDateTime="2023-02-28T18:00:00"/>
      <state disabled="true"
        status="disabled"
        startDateTime="2023-02-28T18:00:00"
        endDateTime="2023-12-31T23:59:59"/>
    </states>
  </propOther>
</ocp>
```

```
<!-- freight version of Y, operational in spring and autumn -->
<ocp id="Y2" parentOcpRef="Y">
  <propOperational trafficType="freight"/>
  <propOther>
    <states>
      <state disabled="true"
        status="disabled"
        startDateTime="2023-01-01T00:00:00"
        endDateTime="2023-02-28T18:00:00"/>
      <state disabled="false"
        status="operational"
        startDateTime="2023-02-28T18:00:00"
        endDateTime="2023-06-01T08:00:00"/>
      <state disabled="true"
        status="disabled"
        startDateTime="2023-06-01T08:00:00"
        endDateTime="2023-08-31T18:00:00"/>
      <state disabled="false"
        status="operational"

```

```
        startDateTime="2023-08-31T18:00:00"
        endDateTime="2023-12-31T23:59:59"/>
    </states>
</propOther>
</ocp>
</operationControlPoints>

<!-- (optional) general state for the whole infrastructure -->
<states>
  <state disabled="false"
    status="operational"
    startDateTime="2023-01-01T00:00:00"
    endDateTime="2023-12-31T23:59:59"/>
</states>
</infrastructure>
```

Best regards,  
Thomas

---