railML.org

railML 3.x
Data Modelling Patterns

(DRAFT)

Overview

- Model hierarchy
- Inheritance
- Layers
- Extension points
- Common domain
- Names of elements and attributes
- Element vs. attribute
- Unknown information
- Default values
- Boolean information
- […]

Draft - For Forum discussion only.

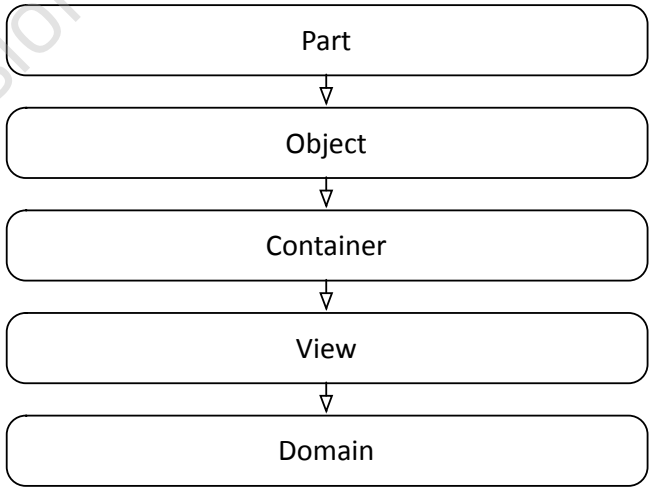
Overview

- […]
- Units
- References
- Multiplicity
- Polymorphism
- XSD Design Patterns
- Sequences
- ID and ID reference

Draft - For Forum discussion only.

railML 3 Data Modeling Patterns

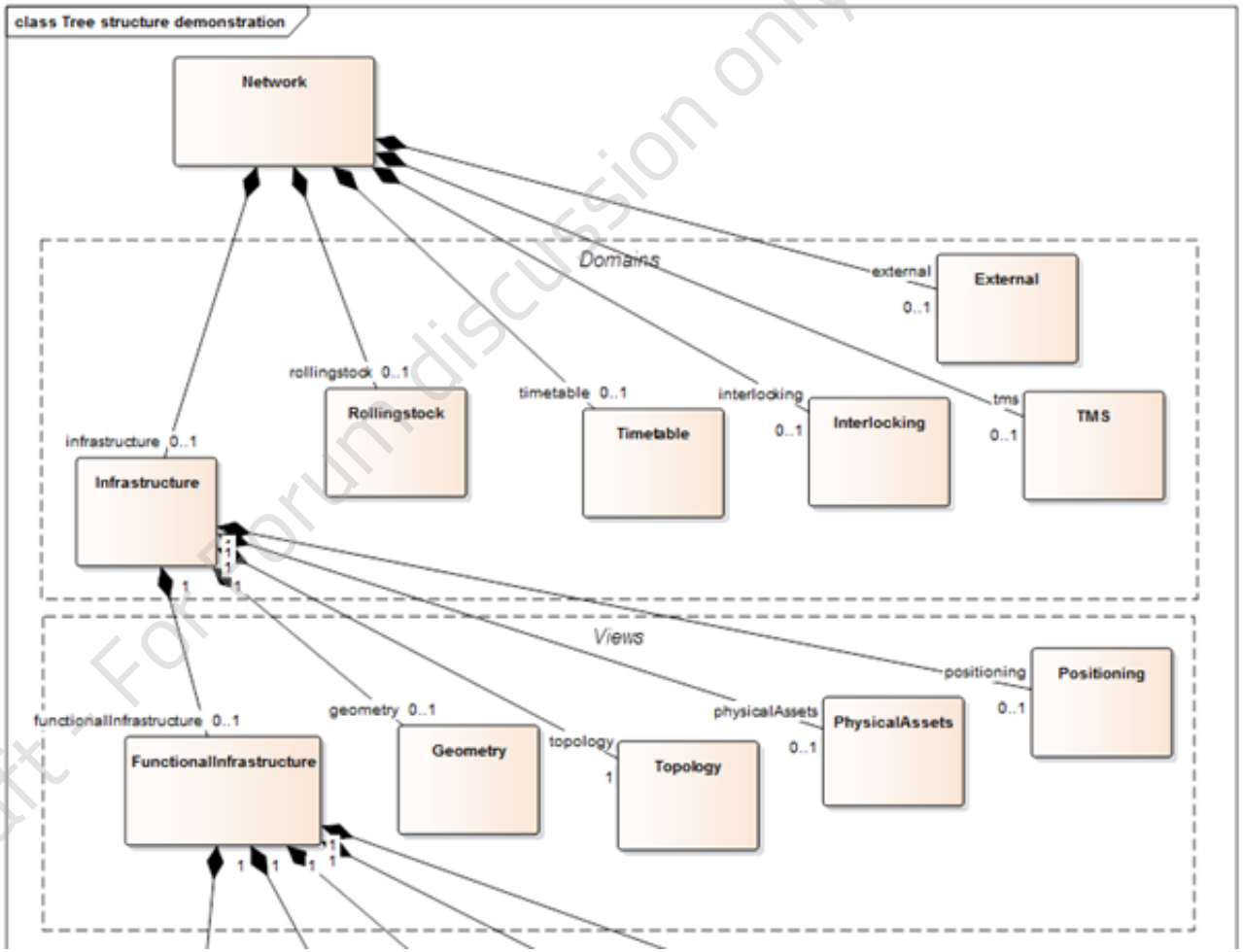
- Model hierarchy
  - **Domain** of knowledge, e.g. infrastructure, timetable, interlocking
  - **View** collects data about a specific aspect of the domain that is of interest to a particular group of use cases, e.g. topology, geometry
  - **Container** represents a type of objects within a view that are functionally different from the objects in other containers, e.g. tracks, signals, horizontalCurves
  - **Object** represents a physical or logical entity of the type indicated by the container, e.g. track, signal, horizontalCurve
  - Optionally, a complex object may be decomposed into **Parts**



Draft - For Forum discussion only.

railML 3 Data Modeling Patterns

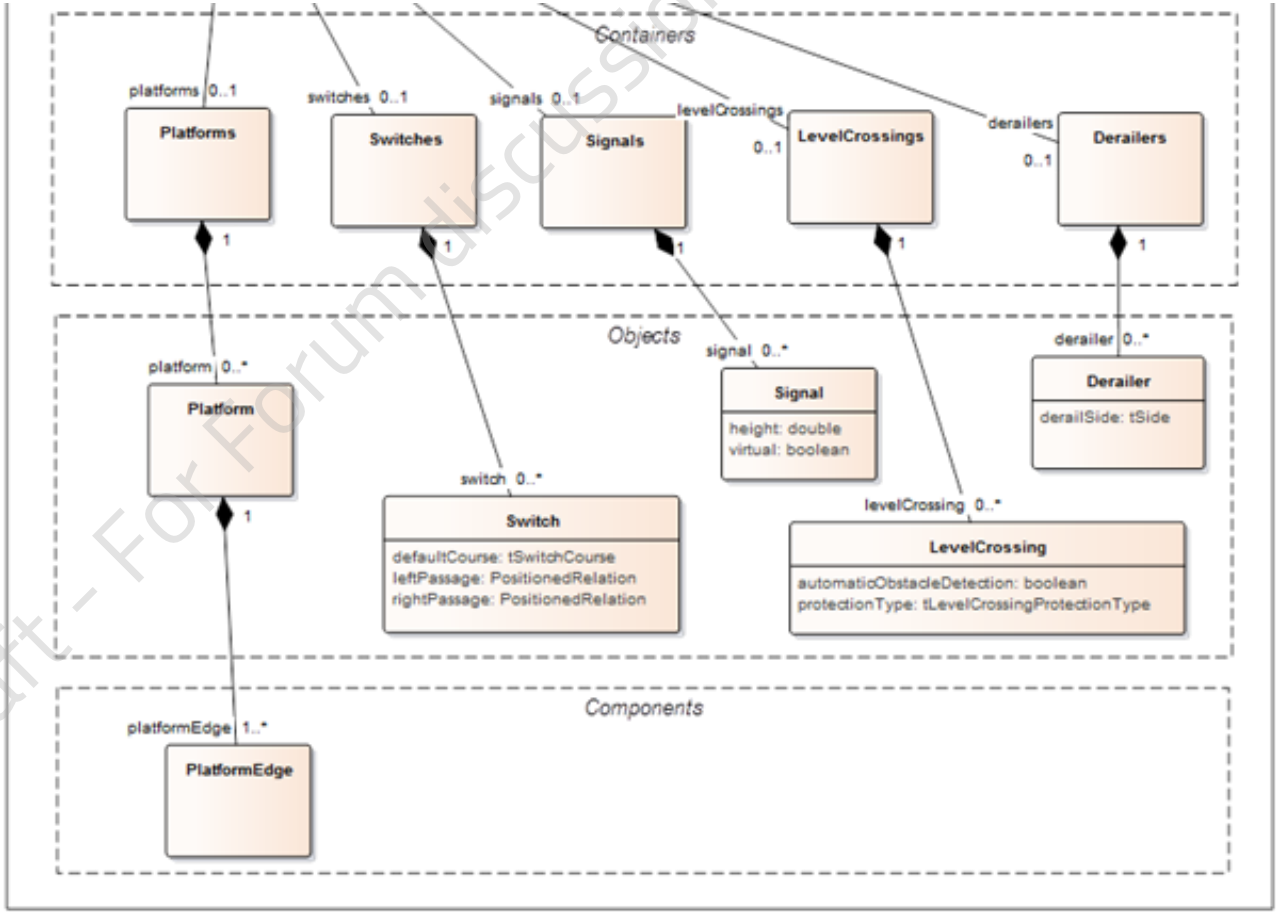
- Model hierarchy



Draft - For forum discussion only.

railML 3 Data Modeling Patterns

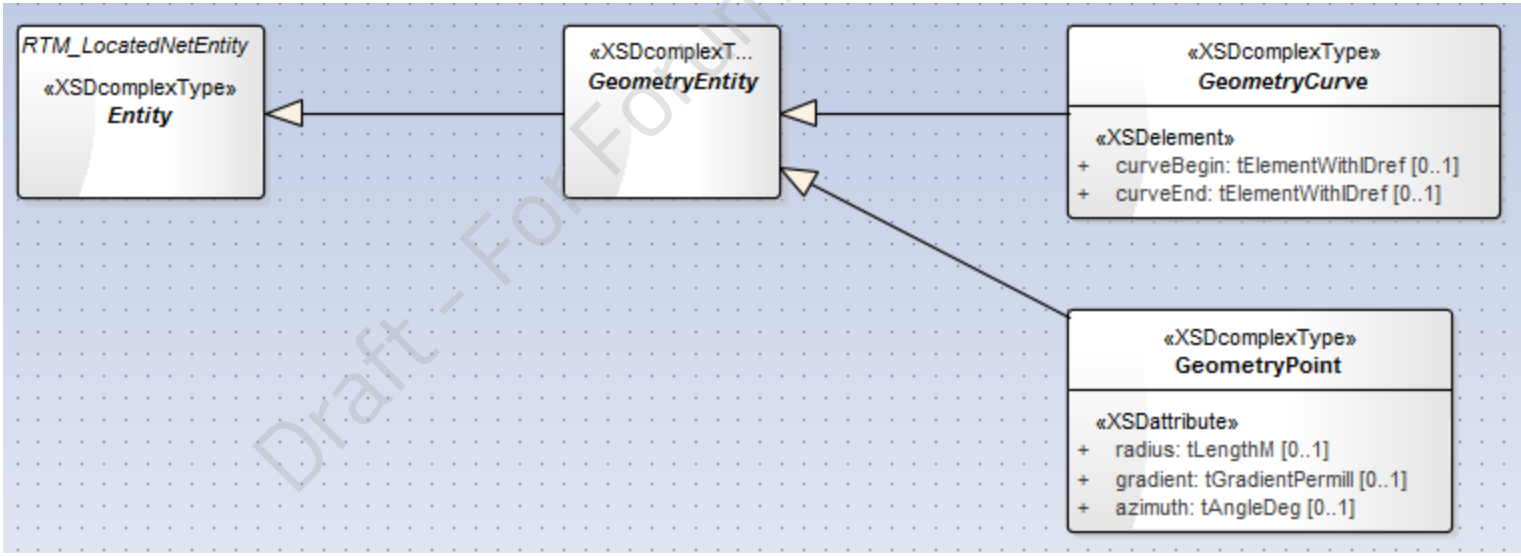
- Model hierarchy



Draft - For Forum discussion only.

railML 3 Data Modeling Patterns

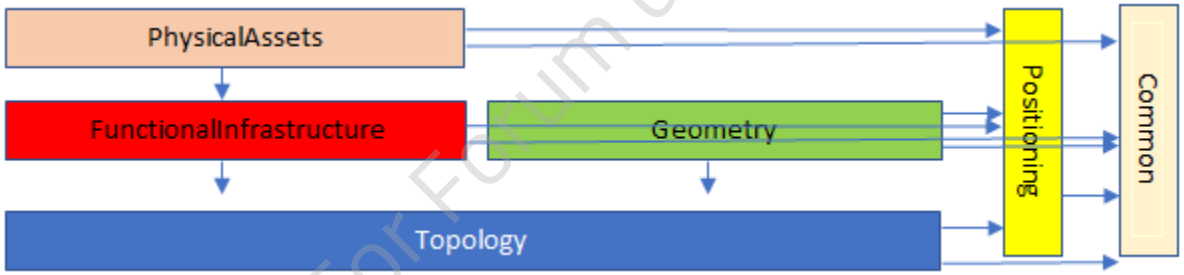
- Inheritance
  - All objects in a *Container* are (normally) of the same class
  - All container specific classes in one *View* inherit from the same abstract base class
  - All view specific base classes shall inherit directly or indirectly from a root class providing a unique object identifier



Draft - For Forum discussion only.

railML 3 Data Modeling Patterns

- Layers
  - Views constitute informational layers within one domain
  - Layers (*Views*) can exist independent from each other
  - See also https://trac.railml.org/ticket/154



Draft - For Forum discussion only.

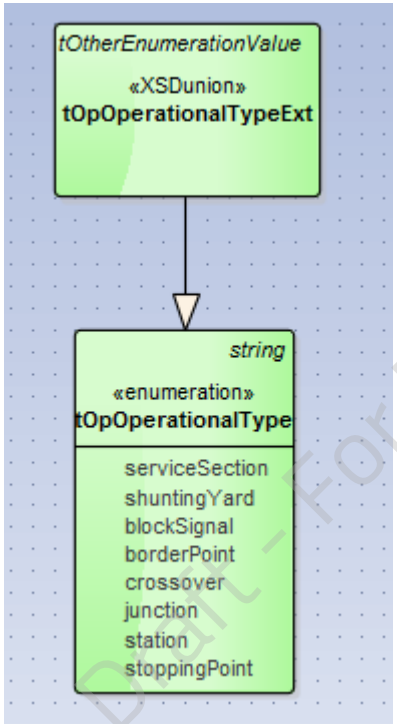
railML 3 Data Modeling Patterns

- Extension points
  - Additional values can be added to enumeration attributes → Other enumeration value
    - New value must not be included in existing list of values
  - Additional attributes can be added to *Objects* and *Parts* → Any Attribute
    - New attribute must not describe information that are already modelled
  - Additional *Containers* can be added to a *View* (including corresponding *Objects* and *Parts*) → Any Element
    - Added objects shall derive from view specific base class
  - Adding new *Views* or *Domains* is to be avoided
    - Major changes → to be discussed with whole railML community

Draft - For Forum discussion only.

railML 3 Data Modelling Patterns

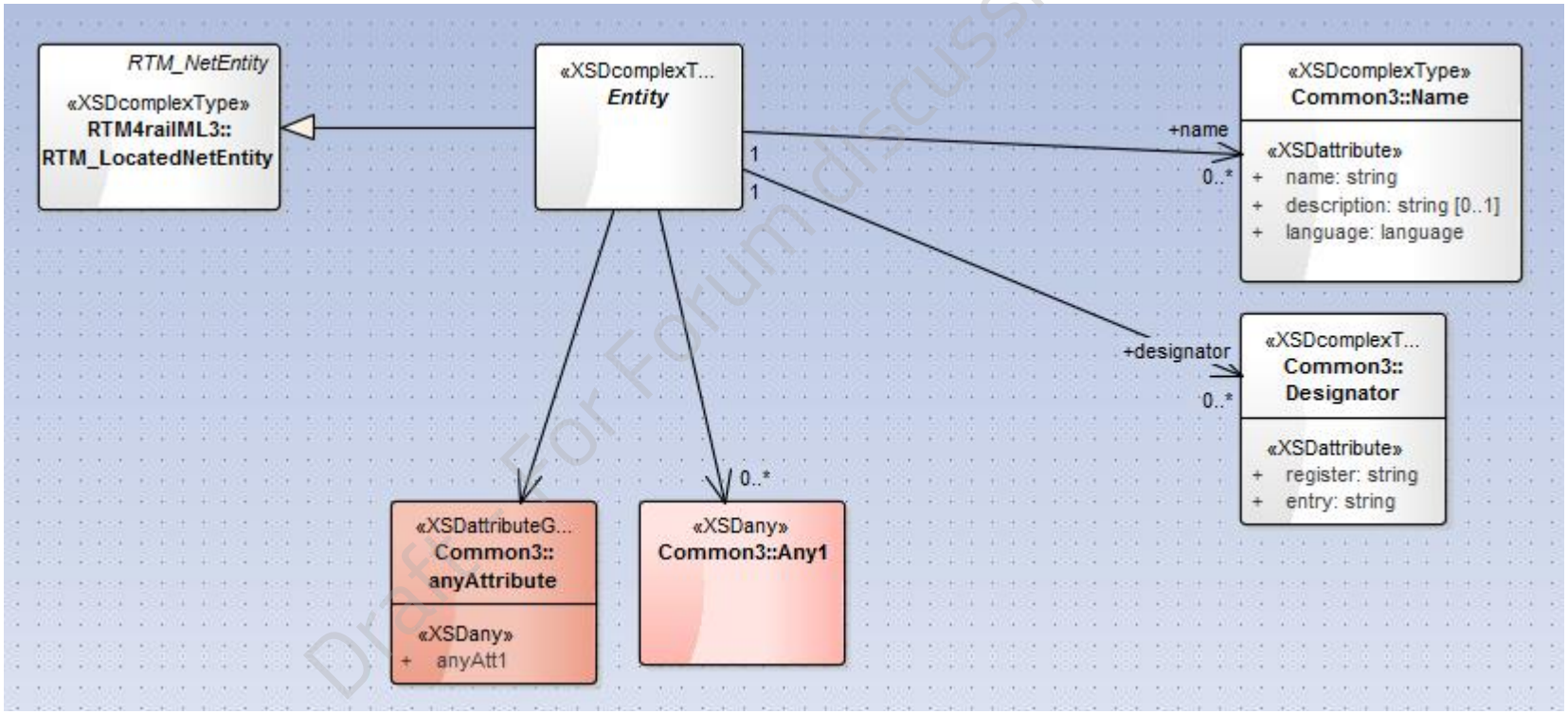
- Extension points
  - Example for Other enumeration value:



Draft - For Forum discussion only.

railML 3 Data Modeling Patterns

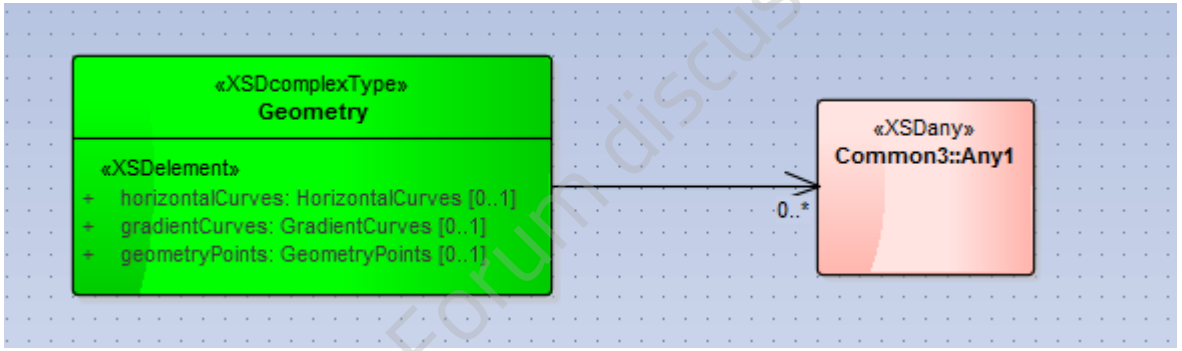
- Extension points
  - Example for Any attribute:



Draft - For Forum discussion only.

railML 3 Data Modeling Patterns

- Extension points
  - Example for Any element:



Draft - For Forum discussion only.

railML 3 Data Modeling Patterns

- Common domain
  - Data that are used by more than one domain shall be placed in a Common domain
    - Time aspects
    - Physical units
    - Railway sector stakeholders (IM, RU, …)
    - Positioning systems
    - …

Draft - For Forum discussion only.

railML 3 Data Modeling Patterns

- Names of elements and attributes
  - Instances of a specific element type shall be named like the element type, but with small letter at the beginning (see Camel Case)
  - Element *Containers* shall be named in plural of the element type name
    - E.g. <borders> contains list of <border> elements
  - Attribute names shall be kept short and should avoid repeating the element name
  - Boolean associations shall be specified by prefix verbs describing the functional relation in more detail
    - E.g. @**requires**PowerLimitation [y/n]
  - Reference attributes and elements shall be specified by prefix verbs describing the functional relation in more detail
    - E.g. <operationalPoint>: <opBorder> → <**limitedBy**Border>
    - E.g. <operationalPoint>: <lineRef> → <**connectedTo**Line>

Draft - For Forum discussion only.

railML 3 Data Modeling Patterns

- Names of elements and attributes
  - […]
  - Names shall be clear w.r.t. their meaning and element/attribute usage
    - Avoid abbreviations
    - There is no limit for the length of element /attribute names from semantic point of view
  - Camel Case shall be used for railML element, attributes and enumeration values
    - E.g. levelCrossing instead of LevelCrossing or level_crossing
    - See also https://trac.railml.org/ticket/110

Draft - For Forum discussion only.

railML 3 Data Modeling Patterns

- Element vs attribute
  - (simple) Information that occurs up to once shall be modelled as attributes
  - Information that can be repeated or that contain further detailed information shall be handled with elements (and child elements)

Height occurs only once

```
<platformEdge id="ple03" height="550">
  <linearLocation id="ple03_lloc01" applicationDirection="both">
    <associatedElement netElementRef="ne_b01" intrinsicCoordBegin="0.4" intrinsicCoordEnd="0.8" keepsOrientation="true">
    </associatedElement>
  </linearLocation>
  <name name="2" language="de"/>
  <trackSection trackRef="trc04" fromPos="200.0" toPos="400.0"/>
  <length type="physical" value="200.00" validForDirection="both"/>
</platformEdge>
```

Length may occur several times (operational length, physical length, ...)

Draft - For forum discussion only.

railML 3 Data Modeling Patterns

- Unknown information
  - Optional attributes that are not used shall be interpreted as „information is unknown“
    - An explicit (enumeration) value „unknown“ is not necessary
  - Mandatory attributes / elements can never be unknown
    - The information whether an element / attribute is mandatory is defined by the use case

Under Discussion

Draft - For Forum discussion only.

railML 3 Data Modeling Patterns

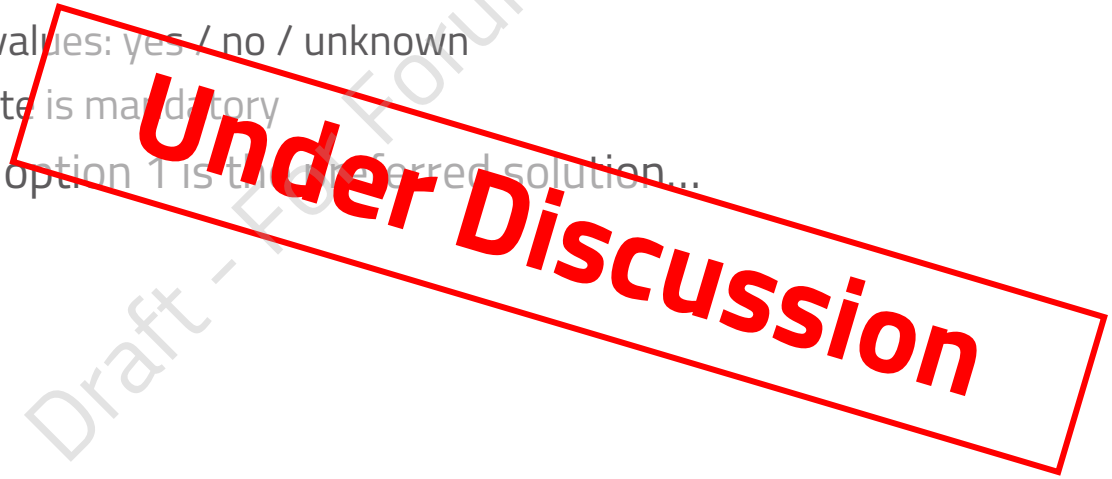
- Default values
  - A default value defines a value to be considered even without explicit mentioning of the attribute / element
  - There shall be no default values
    - Necessary to guarantee the previous pattern about „unknown information“

Draft - For Forum discussion only.

railML 3 Data Modeling Patterns

- Boolean information
  - There are two basic principles of modelling boolean information:
  - Option 1:
    - boolean (1 or 0 / true or false)
    - Attribute is optional
    - if not stated = „information unknown“
  - Option 2:
    - Three values: yes / no / unknown
    - Attribute is mandatory
  - Currently, option 1 is the preferred solution...

Draft - For Forum discussion only.



railML 3 Data Modeling Patterns

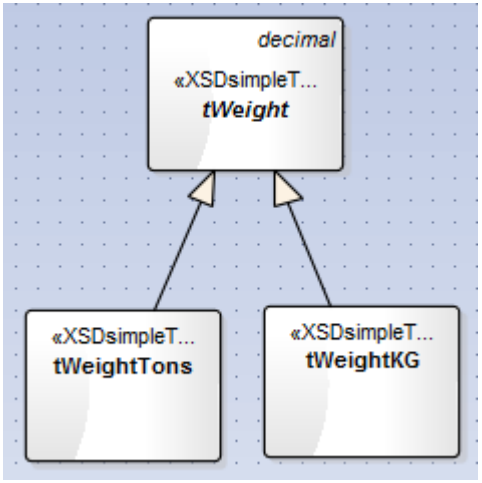
- Units
  - There shall be no flexibility in using physical units wherever applicable
    - The required physical unit shall be indicated by the name of the data type

```
<xs:complexType name="Length">
    <xs:sequence/>
    <xs:attribute name="value" use="required" type="rail3:tLengthM"/>
    <xs:attribute name="type" use="required" type="rail3:tLengthType"/>
    <xs:attribute name="validForDirection" use="optional" type="rail3:tExtendedDirection"/>
</xs:complexType>
```

Draft - For Forum discussion only.

railML 3 Data Modeling Patterns

- Units
  - There shall be no flexibility in using physical units wherever applicable
    - The required physical unit shall be indicated by the name of the data type
    - **SI units** are preferred except for the topics where the railway sector generally uses different units (e.g. train speed in km/h instead of m/s)
    - There shall be maximum 1 to 2 different units for a physical dimension (e.g. length in meters (SI) and millimeters)



Draft - For Forum discussion only.

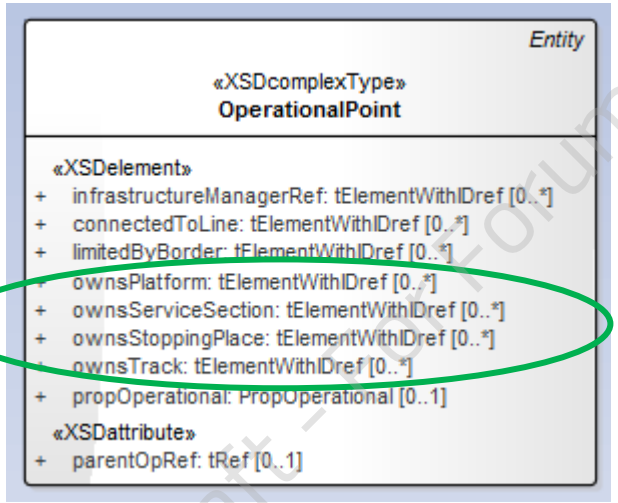
railML 3 Data Modeling Patterns

- Units
  - Deviations from SI unit approach
    - Speed: use **km/h** instead of m/s
    - Energy: use **kWh** instead of Ws (J)

Draft - For Forum discussion only.

railML 3 Data Modeling Patterns

- References
  - Option 1: A macroscopic element shall reference its (microscopic) components
    - E.g. an Operational Point references its platforms, signals, tracks



Draft – for forum discussion only.

railML 3 Data Modeling Patterns

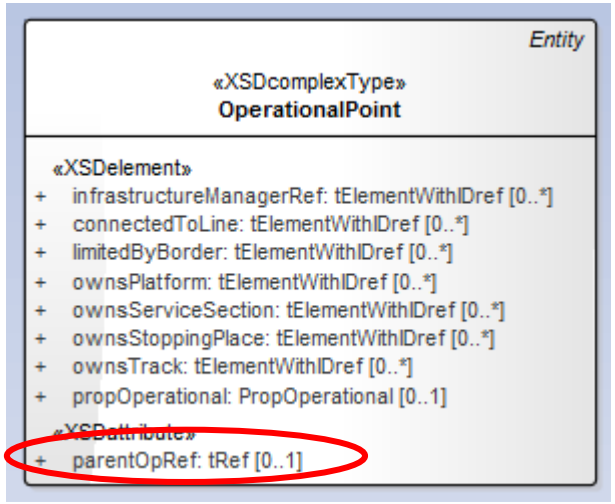
- References
  - Option 1: A macroscopic element shall reference its (microscopic) components
    - E.g. an Operational Point references its platforms, signals, tracks
  - Option 2: A microscopic element shall reference its (macroscopic) container
    - E.g. a signal references its OP

Draft - For Forum discussion only.

Under Discussion

railML 3 Data Modeling Patterns

- References
  - For grouping of elements of same type, the „parent concept“ shall be applied
    - E.g. an OP references its parent OP (for grouping station parts to one station)
    - E.g. a platform edge references its parent platform edge (for grouping platform edges with different height)
    - E.g. a switch references its parent switch (for modelling switch crossings with the help of simple switches)



Draft - For Forum discussion only.

railML 3 Data Modeling Patterns

- Multiplicity
  - Attributes appear up to once (optional): 0..1
  - Containers must not be empty → a Container contains 1..n objects
  - Containers must not have any attributes
  - *Containers* appear up to once (optional): 0..1
  - *Views* appear up to once (optional): 0..1
  - *Domains* appear up to once (optional): 0..1
  - The railML root element must not be empty → at least one domain has to be provided

Draft - For Forum discussion only.

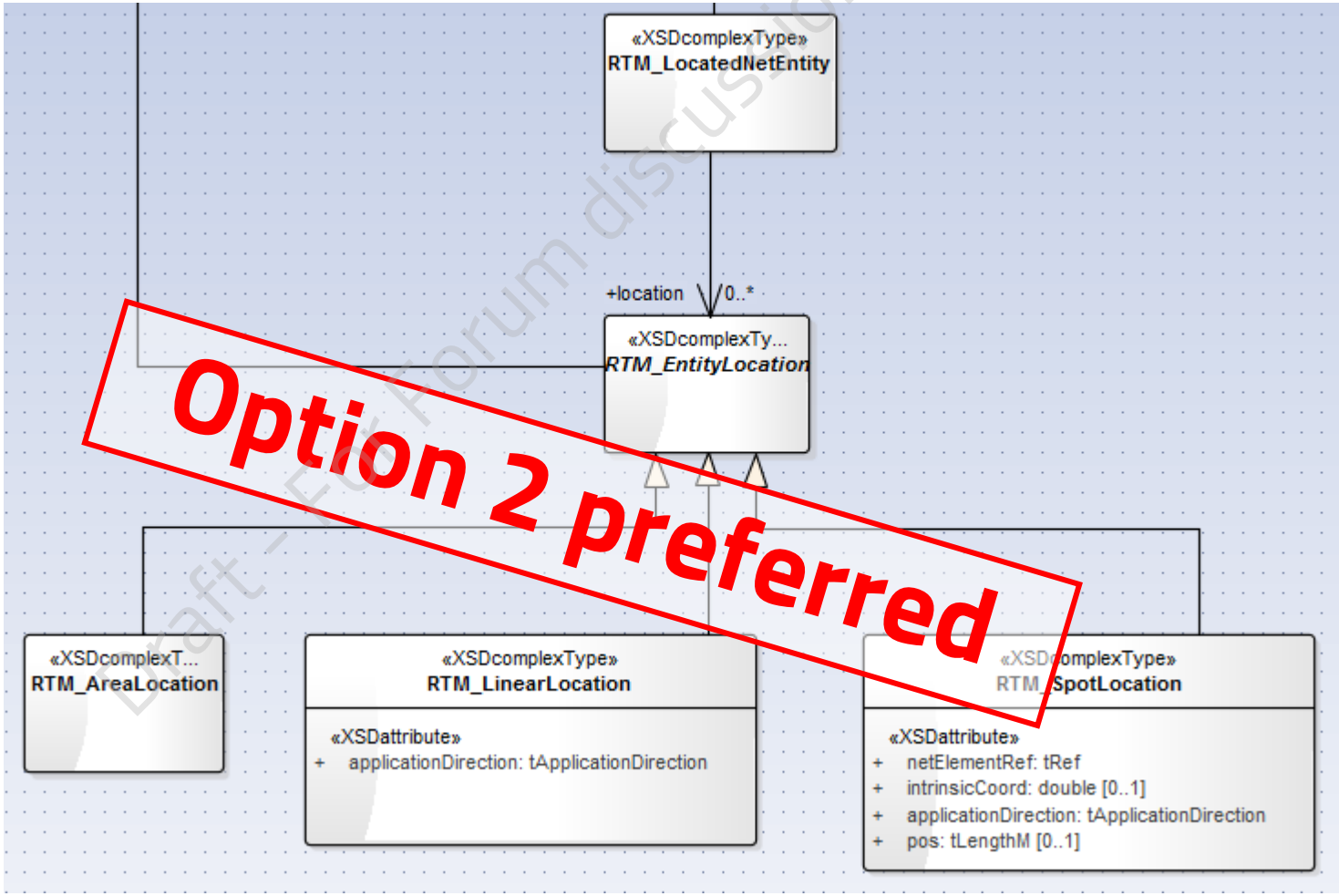
railML 3 Data Modeling Patterns

- Polymorphism:
  - railML 3.x uses the concept of inheritence and polymorphism in modelling
  - There are different ways to model polymorphism in XML, but not all of them can be modelled in UML (Enterprise Architect)
    - Option 1: making use of xsi:type
      - Used in railML 3.1 interlocking
    - **Option 2: direct implementation of derived classes (preferred)**
      - **Used in railML 3.1 infrastructure**
    - Option 3: choices
      - Used in railML 2.x
    - Option 4: substitution groups (not supported by EA)

Draft – For Forum discussion only.

railML 3 Data Modeling Patterns

- Polymorphism
  - Option 1: XSI Type (UML):



Draft – For Forum discussion only.

railML 3 Data Modeling Patterns

- Polymorphism
  - Option 1: XSI Type (XML):

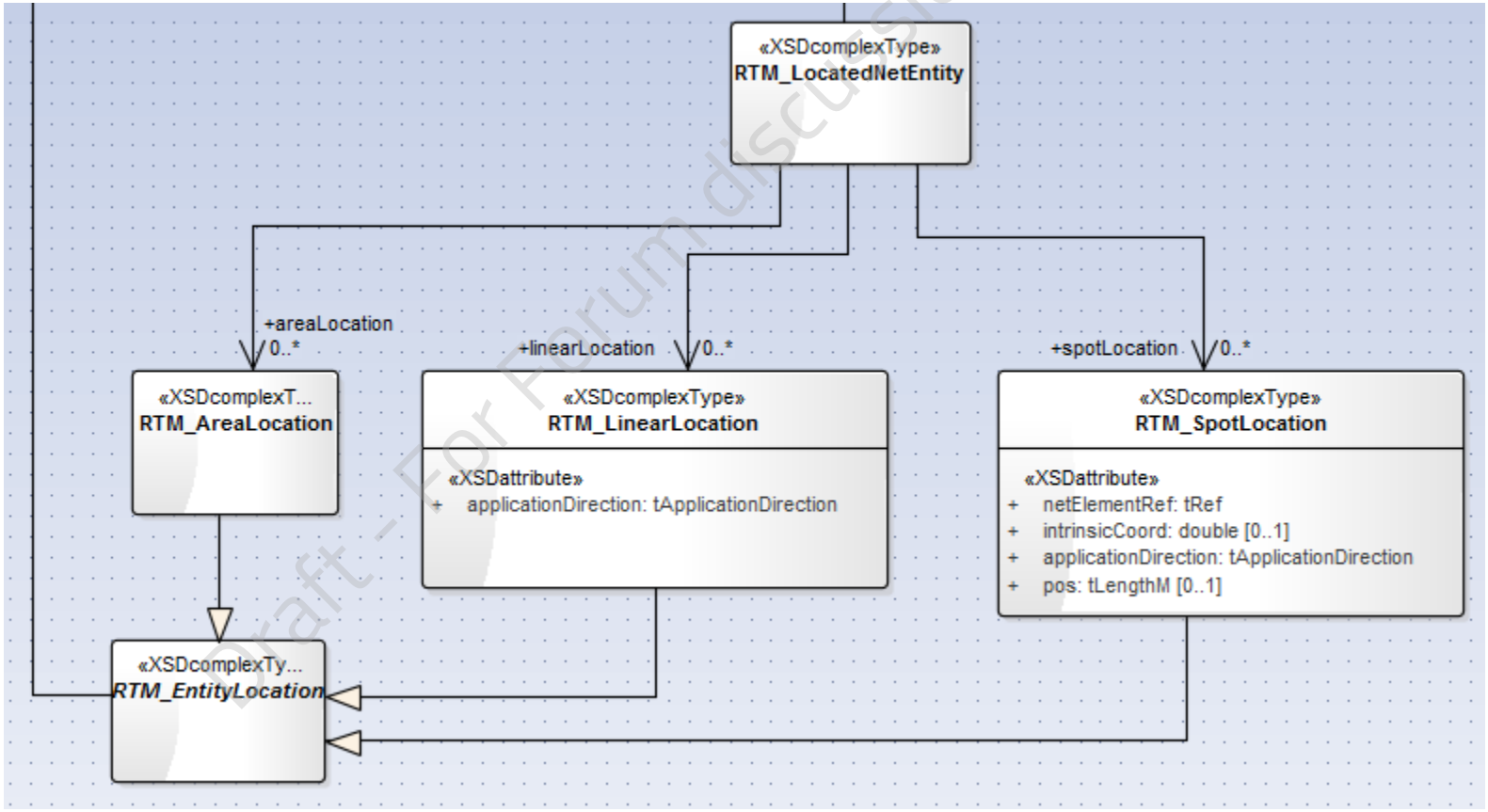
```
<operationalPoint id="opp02">
  <name name="Bf Cstadt" language="de"/>
  <name name="Bouzov" language="cz"/>
  <location xsi:type="RTM_SpotLocation" id="opp02_sloc01" netElementRef="ne_b11" applicationDirection="both">
    <linearCoordinate positioningSystemRef="lps01" measure="4700.0"/>
  </location>
  <designator register="RL100" entry="OCS"/>
  <infrastructureManagerRef ref="im_01"/>
  <opEquipment>
    <ownsPlatform ref="plf02"/>
    <ownsPlatform ref="plf03"/>
    <ownsSignal ref="sig04"/>
    <ownsSignal ref="sig0
    <ownsSignal ref="
  </opEquipmen
  <opOperations>
    <opOperation operationalType="station" traffi Type="passenger"/>
  </opOperations>
</operationalPoint>
```

Option 2 preferred

Draft - For Forum discussion only.

railML 3 Data Modeling Patterns

- Polymorphism
  - Option 2: Direct implementation (UML):



Draft - For Forum discussion only.

railML 3 Data Modeling Patterns

- Polymorphism
  - Option 2: Direct implementation (XML):

```
<operationalPoint id="opp02">
  <name name="Bf Cstadt" language="de"/>
  <name name="Bouzov" language="cz"/>
  <spotLocation id="opp02_sloc01" netElementRef="ne_b11" applicationDirection="both">
    <linearCoordinate positioningSystemRef="lps01" measure="4700.0"/>
  </spotLocation>
  <designator register="RL100" entry="OCS"/>
  <infrastructureManagerRef ref="im_01"/>
  <opEquipment>
    <ownsPlatform ref="plf02"/>
    <ownsPlatform ref="plf03"/>
    <ownsSignal ref="sig04"/>
    <ownsSignal ref="sig05"/>
    <ownsSignal ref="sig06"/>
  </opEquipment>
  <opOperations>
    <opOperation operationalType="station" trafficType="passenger"/>
  </opOperations>
</operationalPoint>
```

Draft - For Forum discussion only.

railML 3 Data Modeling Patterns

- Polymorphism
  - Option 3: Choices
    - Used in railML 2.x

Option 2 preferred

Draft - For Forum discussion only.

railML 3 Data Modeling Patterns

- Polymorphism
  - Option 4: Substitution groups
    - Currently not supported by choosen modelling tool Enterprise Architect v11 *(... and current v14 too ...)*

Draft - For rough discussion only.

railML 3 Data Modeling Patterns

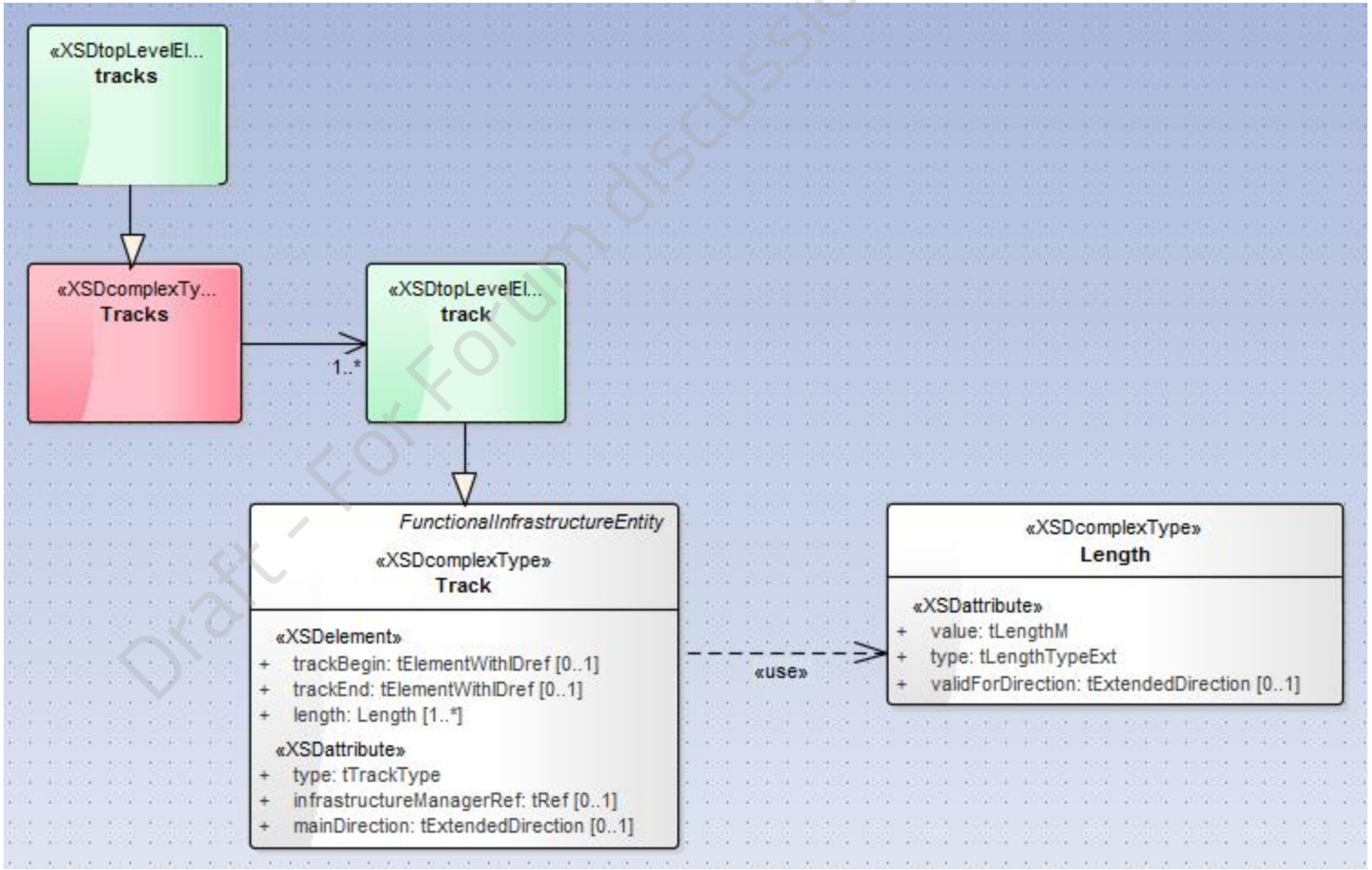
- XSD Design Patterns
  - railML 3.x shall follow commonly known XSD Design Patterns
  - There are different XSD Design Patterns, as described e.g. in https://www.oracle.com/technetwork/java/design-patterns-142138.html
  - Two options are considered being relevant for railML 3.x
    - Option 1: Garden of Eden
    - Option 2: Venetian Blind
    - The options „Salami Slice“ and „Russian Doll“ have been excluded following discussions in the modeling groups

Under Discussion

Draft - For Forum discussion only.

railML 3 Data Modeling Patterns

- XSD Design Patterns
  - Option 1: Garden of Eden



Draft - For Forum discussion only.

railML 3 Data Modeling Patterns

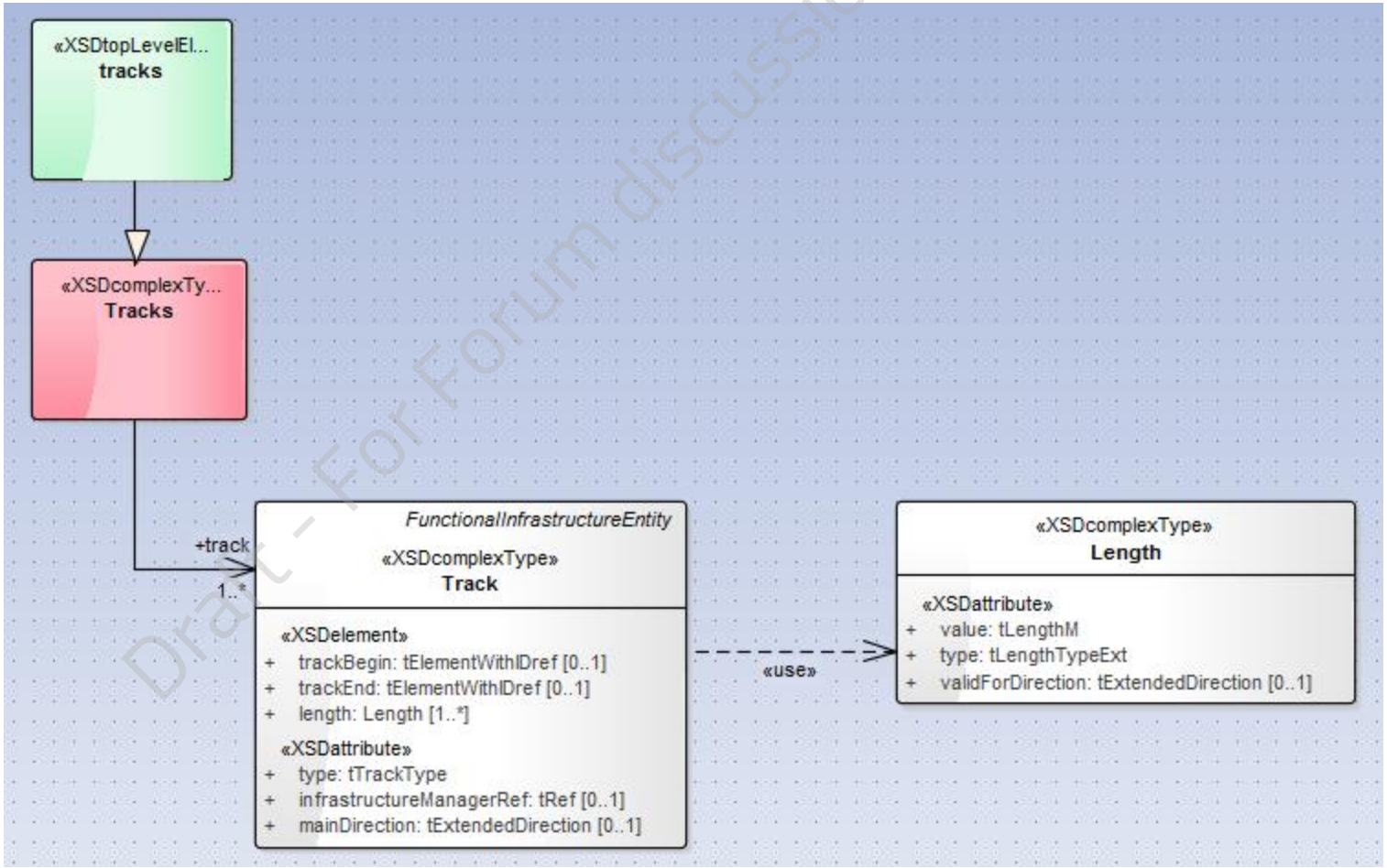
- XSD Design Patterns
  - Option 1: Garden of Eden

```
<xs:complexType name="Tracks">
    <xs:sequence>
        <xs:element ref="rail3:track" minOccurs="1" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="Track">
    <xs:annotation>
    <xs:complexContent>
        <xs:extension base="rail3:FunctionalInfrastructureEntity">
            <xs:sequence>
                <xs:element name="trackBegin" type="rail3:tElementWithIDref" minOccurs="0" maxOccurs="1">
                <xs:element name="trackEnd" type="rail3:tElementWithIDref" minOccurs="0" maxOccurs="1">
                <xs:element name="length" type="rail3:Length" minOccurs="1" maxOccurs="unbounded">
            </xs:sequence>
            <xs:attribute name="type" use="required" type="rail3:tTrackType">
            <xs:attribute name="infrastructureManagerRef" use="optional" type="rail3:tRef">
            <xs:attribute name="mainDirection" use="optional" type="rail3:tExtendedDirection">
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:element name="track" type="rail3:Track"/>

<xs:element name="tracks" type="rail3:Tracks"/>
```

Draft - For forum discussion only.

railML 3 Data Modeling Patterns

- XSD Design Patterns
  - Option 2: Venetian Blind



Draft - For Forum discussion only.

railML 3 Data Modeling Patterns

- XSD Design Patterns
  - Option 2: Venetian Blind

```
<xs:complexType name="Tracks">
    <xs:sequence>
        <xs:element name="track" type="rail3:Track" minOccurs="1" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="Track">
    <xs:annotation>..
    <xs:complexContent>
        <xs:extension base="rail3:FunctionalInfrastructureEntity">
            <xs:sequence>
                <xs:element name="trackBegin" type="rail3:tElementWithIDref" minOccurs="0" maxOccurs="1">..
                <xs:element name="trackEnd" type="rail3:tElementWithIDref" minOccurs="0" maxOccurs="1">..
                <xs:element name="length" type="rail3:Length" minOccurs="1" maxOccurs="unbounded">..
            </xs:sequence>
            <xs:attribute name="type" use="required" type="rail3:tTrackType">..
            <xs:attribute name="infrastructureManagerRef" use="optional" type="rail3:tRef">..
            <xs:attribute name="mainDirection" use="optional" type="rail3:tExtendedDirection">..
        </xs:extension>
    </xs:complexContent>
</xs:complexType>


<xs:element name="tracks" type="rail3:Tracks"/>
```

Draft – For forum discussion only.

railML 3 Data Modeling Patterns

- Sequences
  - xs:sequences shall be used consistently in the railML 3.x schema (instead of xs:all)
    - Only xs:sequence can be used together with xs:any extensions

Draft - For Forum discussion only.

railML 3 Data Modeling Patterns

- ID and ID reference
  - Elements and attributes that represent ID or references to ID have to be based on the ID related data types
    - tID, tRef
  - Elements and attributes that are not ID or references to ID must not be based on the ID related data types
  - See also https://trac.railml.org/ticket/107

Draft - For Forum discussion only.

DLR **Deutsches Zentrum für Luft- und Raumfahrt** e.V.
in der Helmholtz-Gemeinschaft

Institut für Verkehrssystemtechnik

Christian Rahmig
railML Infrastructure Coordinator

www.railml.org

coord@infrastructure.railml.org

+49 – (0)531 – 295 3461
+49 – 173 – 2714509