



# **EULYNX DATA PREP & RailSystemModel**

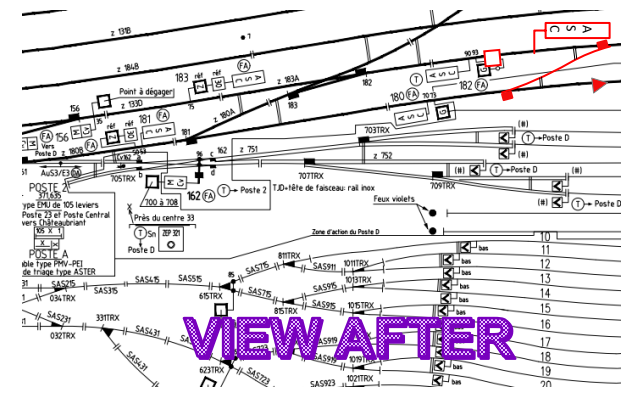
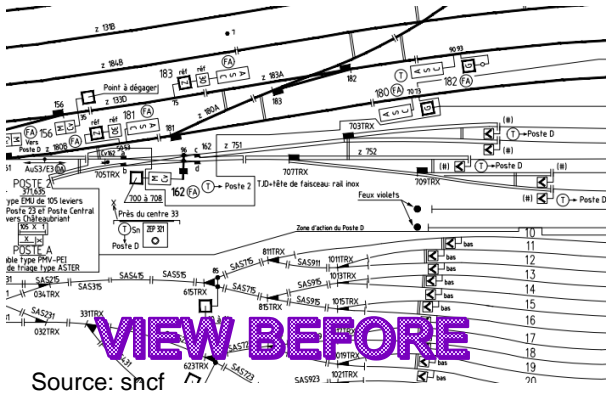
[dataprep.eulynx.eu](http://dataprep.eulynx.eu)

**June 30<sup>th</sup> Web conference**



# Data Prep in a signalling workflow

Defines what work needs to be done  
 “Painting by Numbers”



Data are leading

- single source of truth

View follows data

- Algorithms don't understand pictures

But what do the data mean ?

- “I know a signal when I see one”
- Data ≠ information

```

</generic:usesLocation>
<generic:usesLocation xsi:type="rsmCommon:SpotLocation">
  <rsmCommon:id>d9ca6472-b0d3-b454-b4e4-451fff6fe4e1</rsmCommon:id>
  <rsmCommon:associatedNetElements>
  <rsmCommon:appliesInDirection>undefined</rsmCommon:appliesInDirection>
  <rsmCommon:bounds value="0" />
  <rsmCommon:coordinates ref="dd718aee-d90a-8855-8ac9-fce1e9172b46" />
  </rsmCommon:bounds>
  <rsmCommon:isLocatedToSide>left</rsmCommon:isLocatedToSide>
  <rsmCommon:netElement ref="4ee66dfa-c583-8358-9327-0c925f062db6" />
  </rsmCommon:associatedNetElements>
</generic:usesLocation>
<generic:usesLocation>
  <rsmCommon:id>718aee-d90a-8855-8ac9-fce1e9172b46</rsmCommon:id>
  <rsmCommon:associatedNetElements>
  <rsmCommon:appliesInDirection>undefined</rsmCommon:appliesInDirection>
  </rsmCommon:associatedNetElements>
  </generic:usesLocation>
  
```

```

<generic:usesLocation xsi:type="rsmCommon:SpotLocation">
  <rsmCommon:id>08aae8e3-a887-d85e-956a-a92e659a3c99</rsmCommon:id>
  <rsmCommon:associatedNetElements>
  <rsmCommon:appliesInDirection>undefined</rsmCommon:appliesInDirection>
  <rsmCommon:bounds value="0" />
  <rsmCommon:coordinates ref="eb00807b-7496-7156-951b-ffb4c786208f" />
  </rsmCommon:bounds>
  <rsmCommon:isLocatedToSide>right</rsmCommon:isLocatedToSide>
  <rsmCommon:netElement ref="d676ac98-b811-8059-999e-e5b913104492" />
  </rsmCommon:associatedNetElements>
</generic:usesLocation>
<generic:usesLocation xsi:type="rsmCommon:SpotLocation">
  <rsmCommon:id>d9ca6472-b0d3-b454-b4e4-451fff6fe4e1</rsmCommon:id>
  <rsmCommon:associatedNetElements>
  <rsmCommon:appliesInDirection>undefined</rsmCommon:appliesInDirection>
  <rsmCommon:bounds value="0" />
  <rsmCommon:coordinates ref="dd718aee-d90a-8855-8ac9-fce1e9172b46" />
  </rsmCommon:bounds>
  <rsmCommon:isLocatedToSide>left</rsmCommon:isLocatedToSide>
  <rsmCommon:netElement ref="4ee66dfa-c583-8358-9327-0c925f062db6" />
  </rsmCommon:associatedNetElements>
</generic:usesLocation>
  
```

1111111000101101000000011101111111111  
 10001011010000000111011111111111100010  
 1101000000011101111

Uninformed data is meaningless



EULYNX

# EULYNX DP builds on RSM topo foundations

What we need to know

## How things connect

- Network topology
- Location on the network

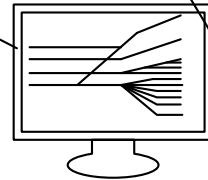
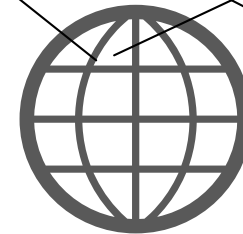
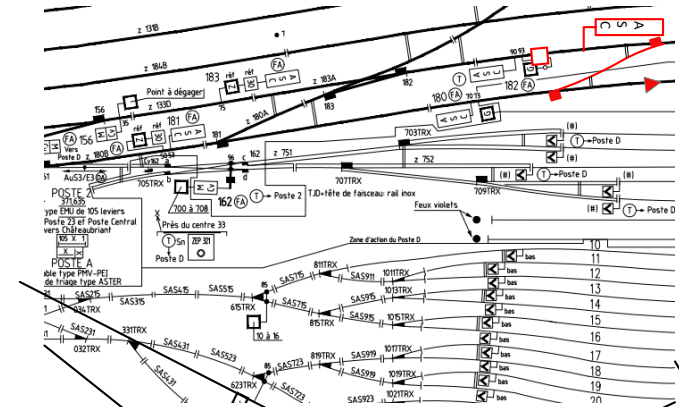
## Where is my kit

- Equipment is positioned ...
  - On earth
  - On a map/plan/display

## Data are structured

- Query/count/delta
- Exploit graph algorithms

People read graphics,  
Computers read graphs





EULYNX

# What's inside the RSM box?

class Horizontal and vertical alignment

## Horizontal and vertical alignment

This diagram features classes representing horizontal and vertical alignment with parametric curves. Refer to the examples on this page for details.

Projection of the alignment curve on the horizontal plane. The horizontal alignment delivers the metric for the linear reference system used to position the vertical alignment segments and the cant segments.

A geometric primitive for constructing a horizontal alignment.

Aggregation Notes: The length of the segment when projected on the horizontal plane. The length is optional because it may be irrelevant or maybe derived from position information associated with the linear location.

Geometric primitive for a straight linear segment.

Geometric primitive for an arc of circle.

Geometric primitive for a transition curves.

Geometric primitive for a straight line segment representing a slope of constant elevation angle.

Geometric primitive for a vertical arc of circle.

Geometric primitive for vertical transition curves.

Geometric primitive for constructing a vertical alignment.

Geometric primitive for vertical transition curves.

Geometric primitive for vertical transition curves.

Geometric primitive for vertical transition curves.

Geometric primitive for vertical transition curves.

Geometric primitive for vertical transition curves.

Geometric primitive for vertical transition curves.

Geometric primitive for vertical transition curves.

Geometric primitive for vertical transition curves.

Geometric primitive for vertical transition curves.

Geometric primitive for vertical transition curves.

Geometric primitive for vertical transition curves.

Geometric primitive for vertical transition curves.

Geometric primitive for vertical transition curves.

### Related diagrams

[Geometry:RSM1.2 Geometry](#)

[Geometry:Introduction to geometry](#)

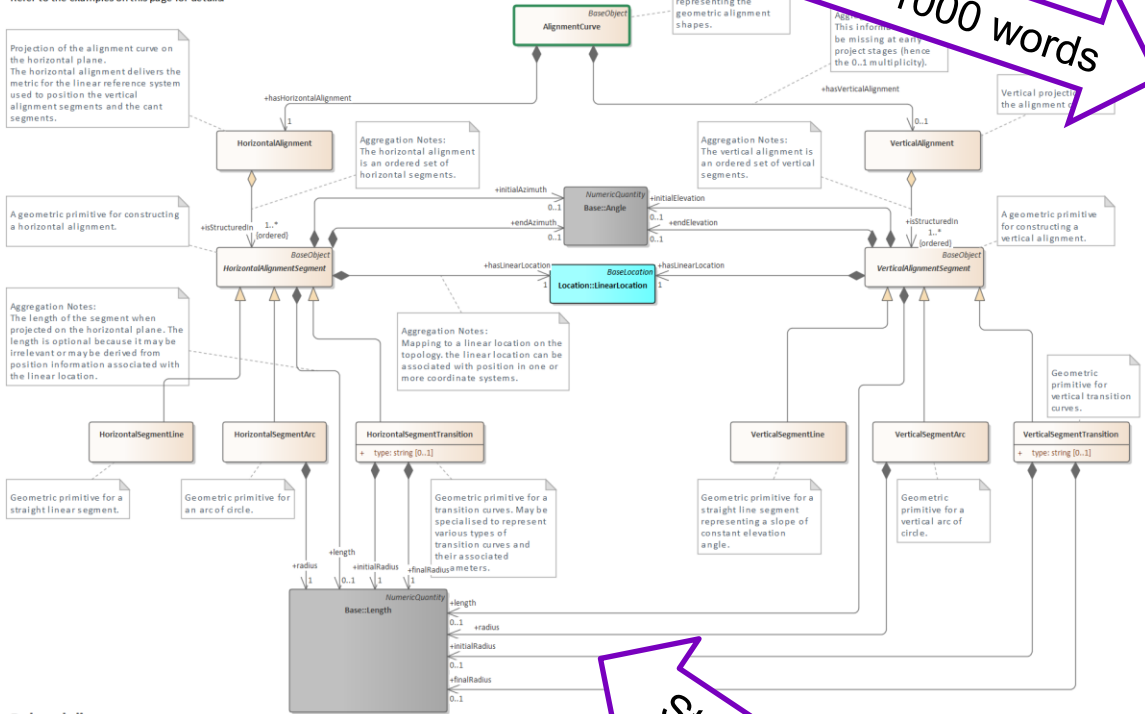
[Geometry:lateral inclination](#)

[Geometry:Discretised point set](#)

Name: Horizontal and vertical alignment  
Author: Bob Janssen  
Version: 1.2beta  
Created: 20-11-2020 16:16:08  
Updated: 17-6-2021 09:23:23

Picture ~1000 words

Structure & meaning





# So you've a nice new model

And now *what* ?

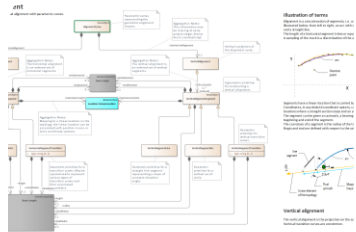
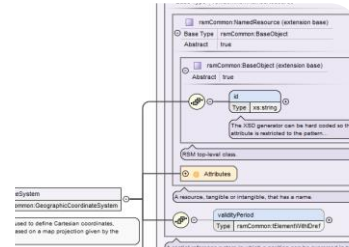


Illustration of terms  
Vertical alignment



```
<code></code>
```

**UML**

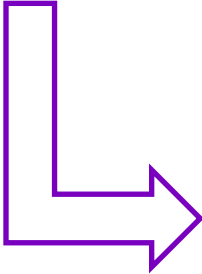
- semantics
- Structure
- Graphic documentation

**XSD**

- semantics
- structure
- No graphics

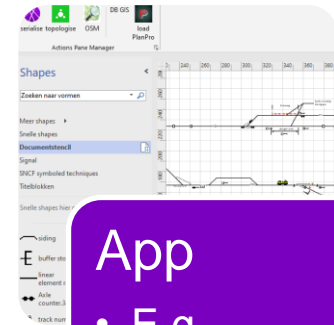
**XML**

- Machine readable
- Validated to XSD
- Structured & meaningful



Or this:

```
<code></code>
```



```
<code></code>
```

**C#**

- Generated dll
- Or other OO code

**App**

- E.g. AutoCAD
- Visio
- Python

**Linearise**

- json
- XML
- yaml

Batteries included: XSD and code generator exist

# A tool demo

## MS Visio front for data

- Includes a library of generated C#-code

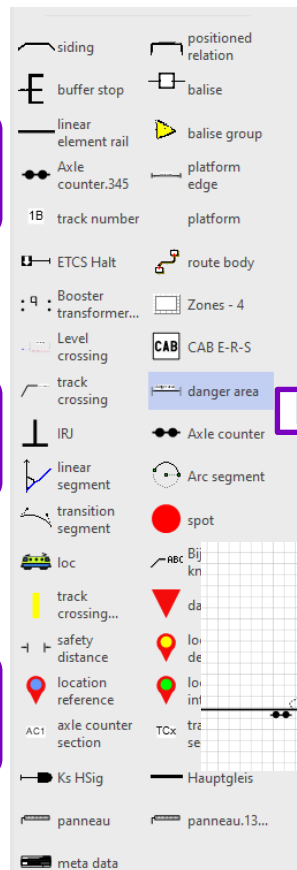
## Signalling stencil

- Clever master shapes
- Signals, points, tracks...

## Linearisation

- In rail-speak: marshal the shapes into a train of information
- *Structured according to model*

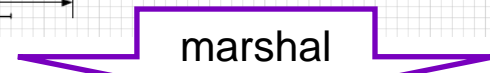
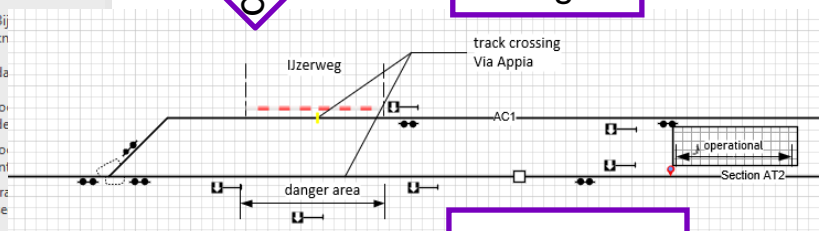
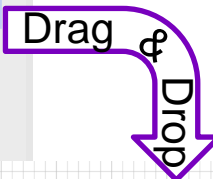
A stencil with master shapes



Shapes have properties

Shapegevegens - balise.1135

OFM name	ACMF Balise 2300
Type description by the OEM, e.g. Eurobalise S21 or tr	
Other product name	
supplier name	
created by role	
short comment	
comment	
date and time stamp	
controlled	FALSE
display short name	FALSE
hideName	FALSE
sleeper material	timber
track type	main
Trackway	sleeper
description	
validity	



```

• EulynxDpInterface "http://data.eulynx.eu/schema/Generic" 6e04152b-f293-965e-a363-abdf141e7563
• rsmCommon:id 6e04152b-f293-965e-a363-abdf141e7563
• generic:containsReleaseOfSubjectData
  • generic:ownsRsmEntities 6419015b-674c-2c5c-ac7f-376219101
  • generic:ownsSignallingEntities
    • generic:ownsDangerAreaTrack f168391d-68ce-2b51-a885-ff
    • generic:ownsSafetyDistance "sig:DistanceToDangerpoint" ab
  • generic:ownsTrackAsset "sig:EtcbsBalise" 7f269573-46fa-d652
    • rsmCommon:id 7f269573-46fa-d652-a947-5598ec736118
      • generic:hasConfiguration 09a81fd7-a31a-0e59-85fc-4b7d
      • sig:refersToRsmTpsDevice "ece53ac8-1f34-3d51-9a79-a7
  
```