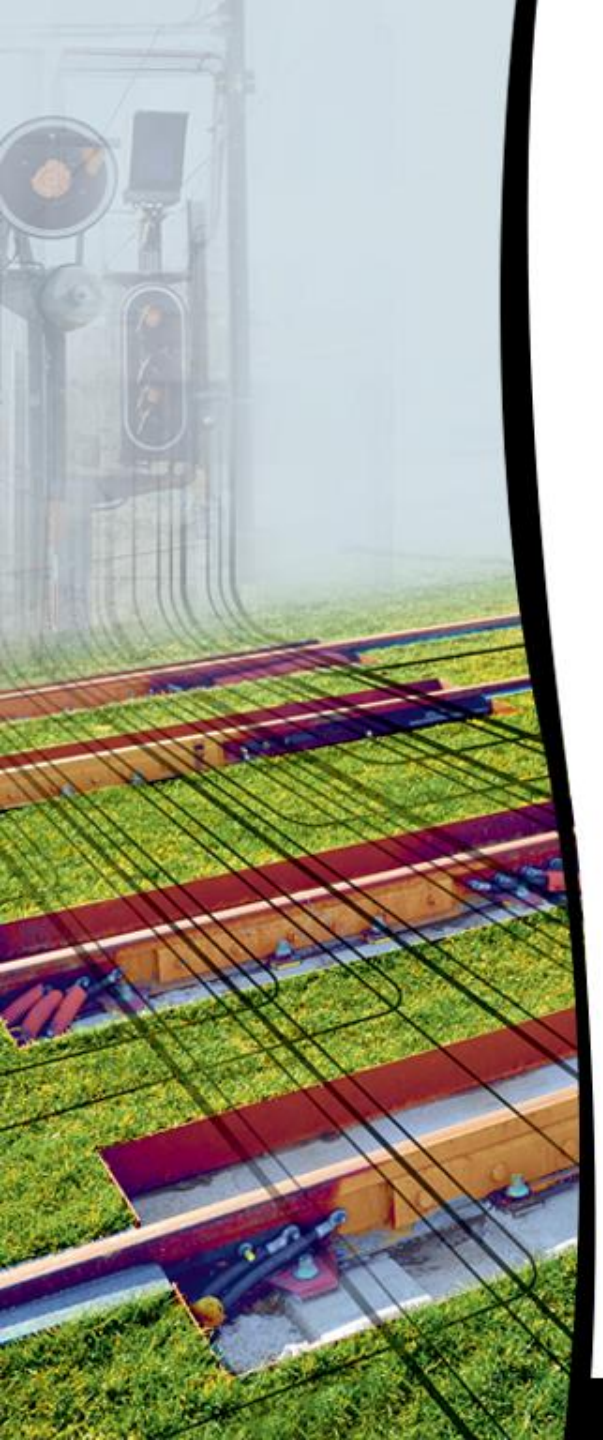


SafeRail

DEVELOPMENT OF A TOOL USING RTM

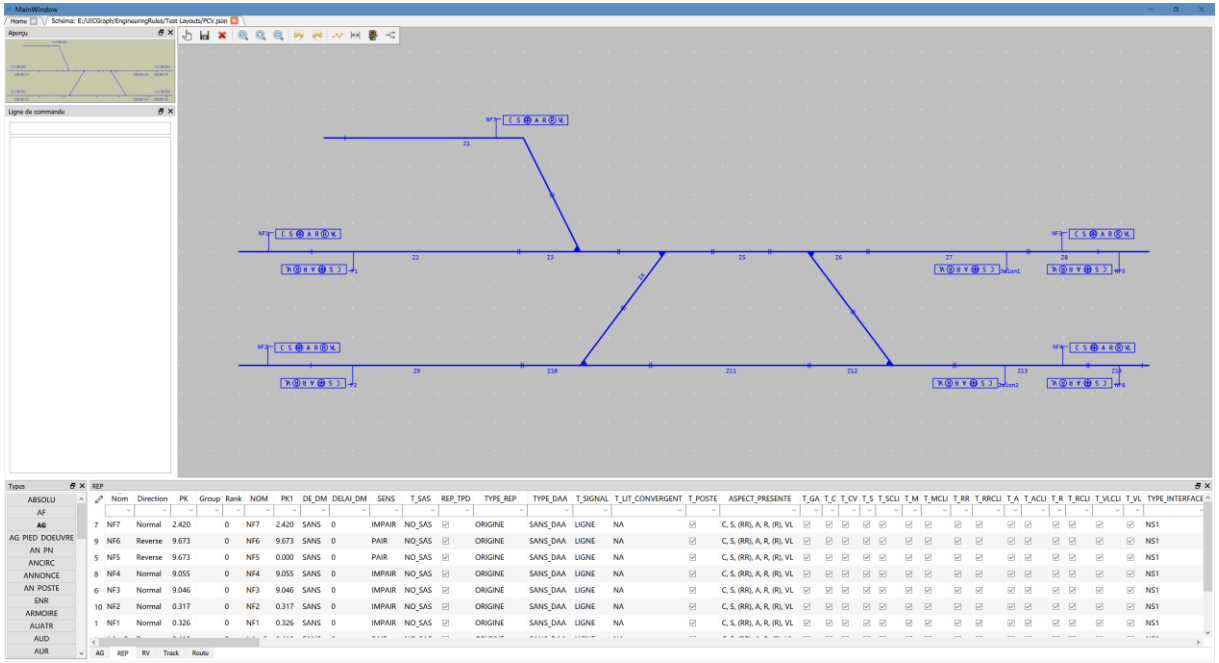
Thierry de Visme : R&I Manager

Zheng Jiang : R&I Software Engineer

SafeRail

- SafeRail has been created in 2003.
- 100 persons in 3 geographical sites (Nantes, Paris and Nanjing).
- Railway Control Command and Signalling.
- Engineering, Consulting, Training, Technical Verification, Testing, Maintenance, Design, Build.
- High Speed Lines, Conventional Lines, Metro, Tram-Train, Tramway, Harbour Railways, Industrial Railways.

PRESENTATION OF OUR TOOL



- Generic tool
- Configurable
- Based on UIC RailTopoModel
- SIL4 certifiable (!)

Technical Plan

Data Preparation

Design Schemes

Data Validation Scenarios

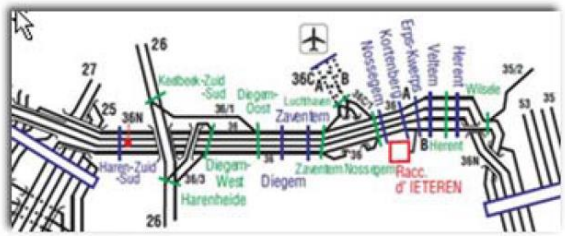
RTM® railTOPOMODEL

PRESENTATION OF OUR TOOL

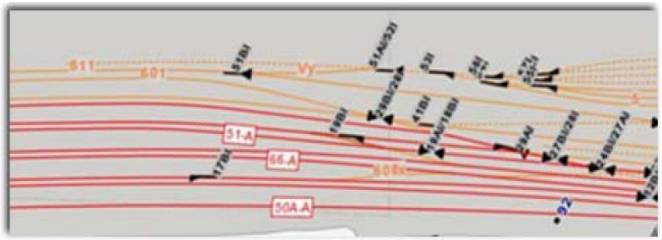
- **RailTopoModel (RTM) ➔ Generic Modeling**

  - ✓ Logic modeling
  - ✓ Interoperability (IRS 30100)
  - ✓ 3 levels of modeling

Macro



Meso



Micro

RAILTOPOMODEL IMPLEMENTATION

- Project started in 2014.
- Based on version 1.0.
- Level limited in micro for specific use in signaling context :
  - Simplified model.

- The use of RailTopoModel saved our time on topologic modeling.
- The graph part is very stable and effective.
- The modeling is proven working without interlocking output.

- However, we have gone through difficulties on the positioning/location part

Difficulties and Suggestions

- The signaling domain modeling is yet done in RailTopoModel.
  - Expecting the work from Eulynx
- Some common object modeling like switches or crossovers are not defined.

DIFFICULTIES AND SUGGESTIONS

- Some equivalent transformations…
  - **PositionedRelation** is well done in term of decoupling of data, but not as easy to use directly in graph search algorithm under context of OOP.
    - Project solution: two shadow objects which represents A and B end of the same **PositionedRelation** are created for quick querying in search

➔easy to overcome

DIFFICULTIES AND SUGGESTIONS

- Some equivalent transformations…
  - **AreaLocation** and **LinearLocation** are composited by a struct “Segment” which is represented by two boundary **SpotLocationIntrinsic.**

➔to replace **AssociatedNetElement** with interface compatibility, since we find oriented **SpotLocationIntrinsic** very handy in graph search algorithms and use it a lot, this reduces lot’s of unnecessary conversions and calculation.

Difficulties and Suggestions

- Some problems…
  - Constraint of location not defined, for example:
    - **NetEntity** with N locations in the same level is legal or not?
    - If it's legal, N **AreaLocation** = ?
    - If not, needs to have **NetEntity** with n **SpotLocationIntrinsic** for objects such as switches or level crossing
    - Maybe need of a rule to regulate this.

DIFFICULTIES AND SUGGESTIONS

- Some problems…
  - Needs of **NetEntity** without any location, especially for interlocking concepts and states represented by object.
    - Example: to establish the route X or Y we need to check switch A is on the left -> in route object X and Y we need a state object (A, Left) to represent check

      => a state object without location
    - Example: to establish the route X we need condition Y that checks TC01 is ordered to down direction and TC02 is not occupied

      => a conceptual object without any location

DIFFICULTIES AND SUGGESTIONS

- Some problems…
  - Milestone definition not standardized, **LinearPositioningSystem** and **LinearCoordinate** is decimal
    - Example: KM 556 has only 732 meters, KM 780 has 1012 meters, and you can have KM 780+1011 which can't be represented by a decimal number.

    ➔Private struct "PK" used to replace decimal numbers.

SafeRail

Demonstration