

POSSIBILITIES OF HIGH-SPEED RAILWAY TURNOUT DATA DESCRIPTION

ADAM HLUBUČEK

Czech Technical University in Prague, Faculty of Transportation Sciences, Konviktská 20, Prague, Czech Republic

correspondence: hlubuada@fd.cvut.cz

ABSTRACT.

This paper aims to summarize possibilities how to create data models of high-speed railway turnouts. The turnouts designed for high-speed operation require specific geometric solution. As the UIC RailTopoModel is considered an international recommendation in the field of data modeling of railway infrastructure, this issue is assessed in terms of models based on its principles. Solving this problem can affect the future development of the Multipurpose Railway Infrastructure Model gradually emerging at the CTU Railway Laboratory in Prague using the RailTopoModel principles.

Whereas the RailTopoModel itself does not define any specific types of entities, the railML® 3.1 specifications are also used for assessment purposes. Turnouts are viewed both in terms of topology and in terms of functional infrastructure. In the final sections, recommendations are given on how to deal with the problems found, e. g. in terms of implementation into the Multipurpose Railway Infrastructure Model.

KEYWORDS: High-speed turnout, railway infrastructure, RailTopoModel, railML.

1. INTRODUCTION

A railway turnout is a very complex technical equipment consisting of many components. It allows the vehicle movement to be transferred between two continuing tracks. Conventional turnouts can be simply described by a not very large set of parameters, as the radius of each of their branches has a constant value. In order to increase speed in the branching direction, the high-speed turnouts are designed using a more complex geometric solution. This is because they require precise track geometry adapted to limit the high dynamic forces generated by the passage of a vehicle. [1, 2]

In order to fulfill multiple needs of many specific process information systems (for example automatic train control systems, automatic train operation systems and train scheduling applications), an exact and unambiguous data description of railway infrastructure is necessary. Detailed knowledge of the track geometry parameters is especially essential to calculate the static speed profile, which defines the allowed speed for certain types of railway vehicles all along the route.

The CTU Railway Laboratory in Prague focuses to support development of the above-mentioned technologies. [3] Among others, the Multipurpose Railway Infrastructure Model is being developed at the workplace. The data model is designed to create the infrastructure core database, which structure should enable to meet the needs of related applications and allow to export the data to standardized formats (for example railML® 3). This paper proposes several possible ways how to implement the high-speed turnouts

to the data model.

2. SELECTED SPECIFICS OF HIGH-SPEED TURNOUTS

Maximum allowed running speed of a railway vehicle when passing a horizontal curve is limited by the value of unbalanced lateral acceleration (caused by superelevation deficiency) and its rate of change with respect to time. Abrupt change of superelevation deficiency and the rate of change of superelevation deficiency as a function of time are often used to express these characteristics. If no superelevation is applied to a segment of a track (or to a branch of a turnout) the speed essentially depends on the radius of curvature and the way it changes. [4]

In order to avoid abrupt change of superelevation deficiency, transition curves are usually provided between a straight and curved segment of a track. Their purpose is to change the radius from infinity at the straight segment of the track to the radius of the following circular arc gradually, i. e. without abrupt change of curvature. Transition curves can be used to join two circular arcs with different radius of curvature as well. Nowadays, clothoids are mostly adopted to provide transition curves. The clothoid is a curve with a linear variation of curvature. In the case of a superelevated arc, gradient due to superelevation (also called as superelevation ramp) is often applied within the adjacent transition curve. [5]

The most of ordinary turnouts are designed without using transition curves and without superelevation. It causes relatively high values of abrupt change of su-

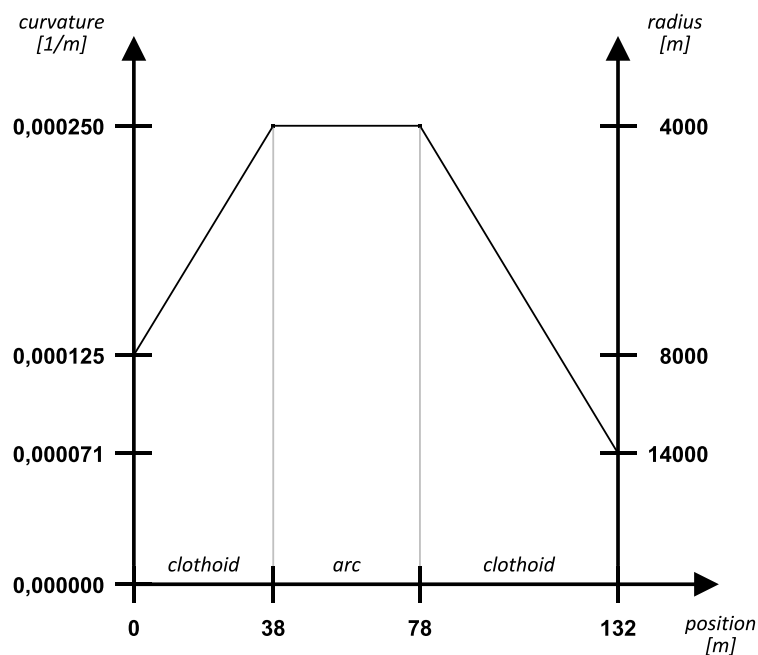


FIGURE 1. The course of curvature in the turning branch of the $J60-1:33,5-8000/4000/14000$ -PHS turnout

perelevation deficiency affecting the passing vehicle. In order to reduce lateral forces between wheel and rail while allowing high-speed train operation, it is necessary to adjust the turnout geometric parameters accordingly. High running speed requires high value of radius which significantly affects the shape and size of a turnout. Therefore, slim turnouts adapted for high-speed operation usually reach the designed length of several hundred meters. The radius of the branch arc typically reaches a value of several thousand meters also in the turning direction, in such cases. For this reason, long switch rails are necessary for vehicle to be able to turn onto another track. These switch rails often require several point machines to be handled, then. An additional point machine may be required to manipulate the movable parts of the frog. [6]

In some cases, the geometry of a high-speed turnout branch is optimized by the means of transition curves application. For example the turnout type designated as $J60-1:33,5-8000/4000/14000$ -PHS (which is the first turnout with unstable curvature developed in the Czech Republic) is designed using two clothoid curves in the turning branch. Turning away from the straight direction, the branch begins at the radius of curvature of 8000 m, gradually changing to 4000 m (at the length of 38 m), which is the radius of the following circular arc (of the length of 40 m), followed by the second transition curve (54 m long, in this case) ensuring change the radius of curvature to 14000 m. The course of curvature in the turning branch of this turnout is shown in the *Figure 1*. Continuation of this clothoid can ensure that the inflection point is reached at an appropriate distance so that it allows the construction of a simple crossover assembled from these

turnouts with axis spacing of 4,75 m. [7]

3. RAILWAY INFRASTRUCTURE DATA DESCRIPTION BASED ON THE UIC RAILTOPOMODEL

The RailTopoModel initiative was introduced in 2013 [8] in order to create a common generic standard as regards the railway infrastructure data modeling. In 2016, the RailTopoModel was released as UIC International Railway Solution 30100 [9]. The UIC RailTopoModel [10] represents a topological model of railway infrastructure based on a „connexity graph“. The whole model consists of several subsystems, as they follow:

- *Base*
- *Topology*
- *Positioning*
- *Net Entity*
- *Location*

[11]

The model makes it possible to describe the railway infrastructure at several different levels of detail. So-called net elements and net relations are used to create the topology of a network. Individual net elements can be grouped in order to express an element at a less detailed level.

For functional reasons, the most detailed level should be defined with the use of linear elements, which can represent, for example, tracks or line sections (depending on the intended purpose of the specific model use case). It is possible to express certain position within a net element using intrinsic coordinate. This coordinate can take value from 0 to

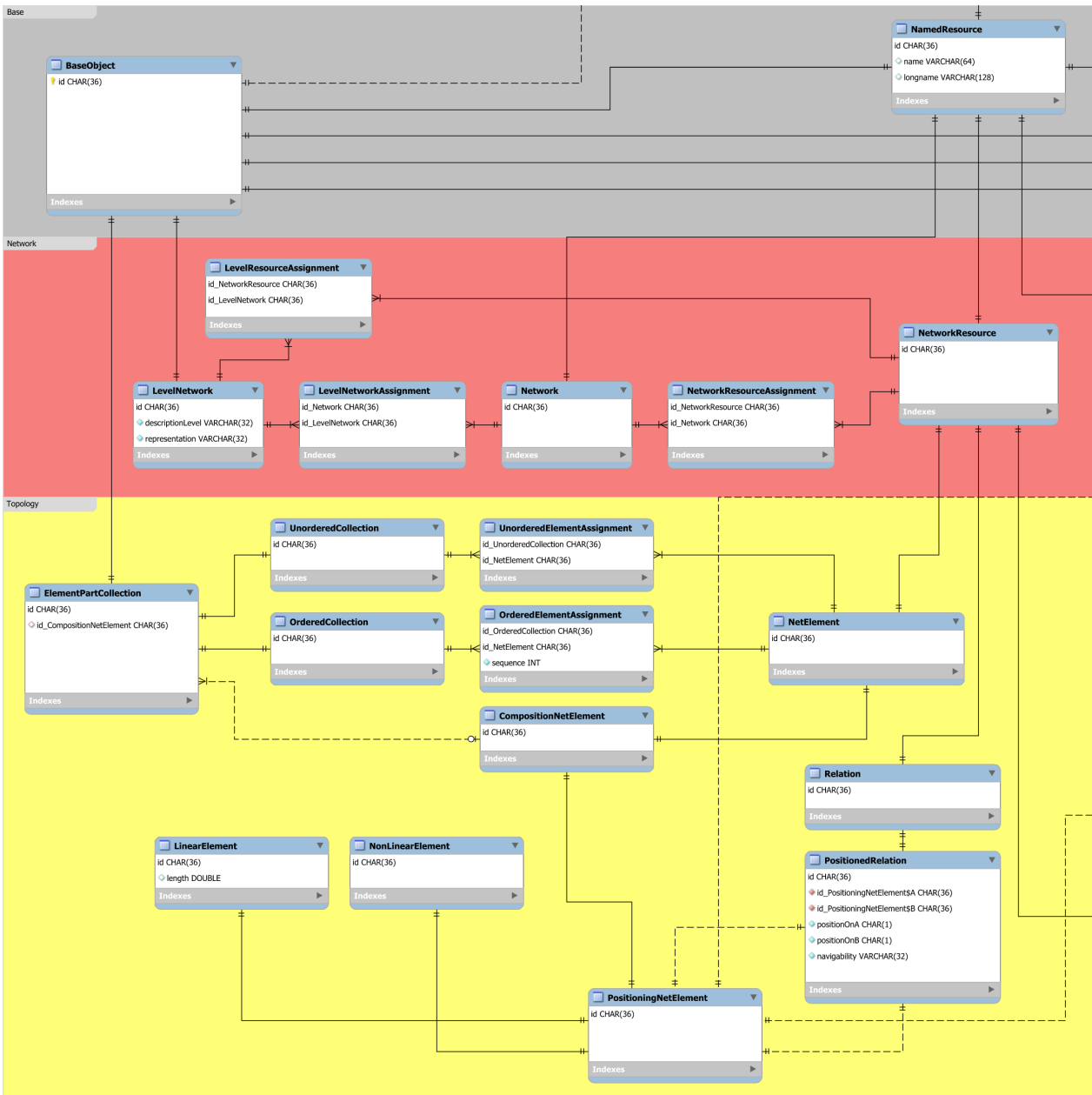


FIGURE 2. The base, network and topology modules of the Multipurpose Railway Infrastructure Model

1, whereas 0 means the beginning and 1 indicates the end of a linear net element. Two net elements can be joined only at the position of 0 or 1, which is realized by the means of net relations. The navigability attribute defines whether is enabled to pass through the given net relation or not (resp. in which direction).

The RailTopoModel also allows to define several positioning systems of two types. It is possible to create both linear and geometric positioning systems. Linear and geometric coordinates belonging to these positioning systems can be assigned to individual intrinsic coordinates of net elements as well as to appropriate locations of net entities. Net entities themselves represent different types of railway infrastructure objects and characteristics which can be localized to the topological layer. The RailTopoModel itself do not define

any specific types of net entities. These types can be introduced as individual extensions to the model.

Depending on the nature of localized object or property (eventually level of detail), the net entities can be placed using spot, linear and area type of location. A location of a net entity can be created based on both intrinsic and external (linear or geometric) coordinates. Several different locations for the same net entity instance are enabled. Linear and area locations can cover several (sections of) net elements which is provided by the associated net element class. In some cases, as regards these locations, it is essential to specify values of the keeps orientation and application direction attributes. [9, 11]

Currently, the most advanced use case of the RailTopoModel approach is apparently the railML® 3 data format, [12] as mentioned in more detail further.

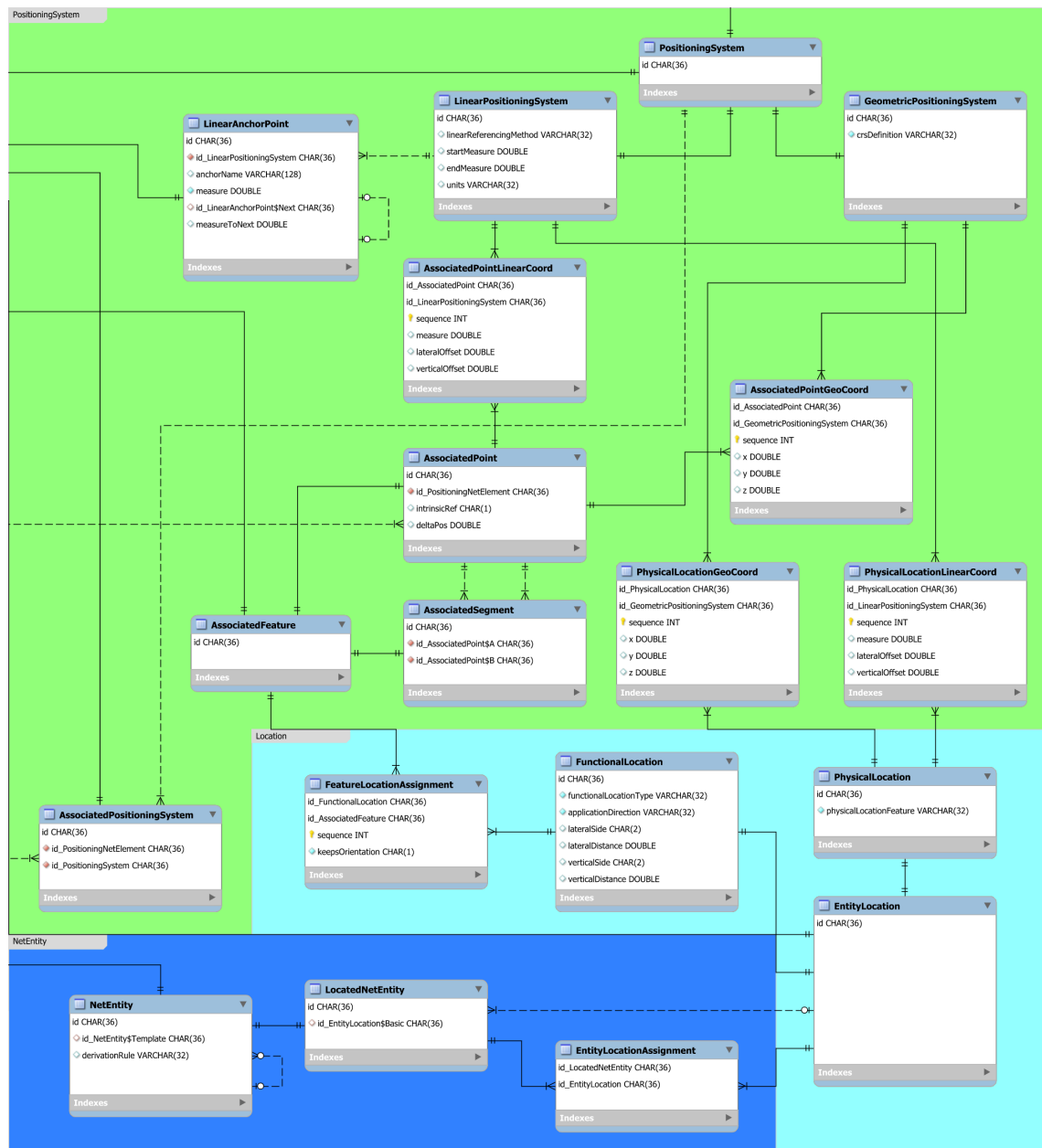


FIGURE 3. The positioning system, net entity and location modules of the Multipurpose Railway Infrastructure Model

4. MULTIPURPOSE RAILWAY INFRASTRUCTURE MODEL DEVELOPMENT

In order to deploy infrastructure data repository for several key projects dealing with spatial aspects of the railway domain, the CTU Railway Laboratory in Prague decided to implement the core model, largely based on the RailTopoModel principles. The Multipurpose Railway Infrastructure Model is a relational data model which directly allows it to be used as a database meeting MySQL standards. At the same time, however, it reflects an object-oriented approach to railway data modeling.

Each type of data object is divided into several tables connected by identifying relations representing successive specializations of the particular object types.

By simply joining such interconnected tables, a comprehensive overview of individual specialized objects can be obtained. In addition, this approach allows to easily add, remove and change the attributes common for several different types of objects.

Whenever possible, the relational model tables reflect the RailTopoModel classes in terms of naming, attributes and connections. The base, network and topology modules (as can be seen in the *Figure 2*) basically correspond to the RailTopoModel layout, from this point of view. For functional purposes, the cardinality of n:n has been introduced, as regards some relations. In such cases, appropriate interconnection tables are added to the model to express the assignment. In the case of more substantial functional changes, new names of the tables are used in order to avoid confusion.

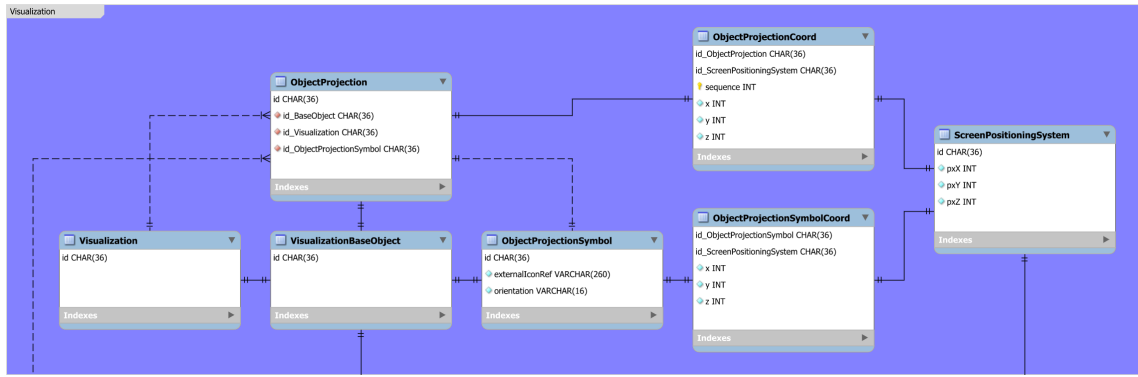


FIGURE 4. The visualization module of the Multipurpose Railway Infrastructure Model

Such changes are primarily about implementation of the associated coordinate and associated net element classes converted to the associated point and associated section tables and adjustment of the concept of the net entity locations which distinguishes between physical and functional (associated) location of net entities. For technical reasons, the mechanism of assigning linear a geometric positioning system coordinates to associated points and physical locations has also been modified. These issues are related to the positioning system, net entity and location modules shown in the *Figure 3*.

Inspired by the railML® 3.1 specifications, [12] the concept of net entity templates has been added to the The Multipurpose Railway Infrastructure Model. The net entity table has been adapted for this purpose. The table is located at the interface between the core and use case-specific part of the model. The model has also implemented the visualization module introduced by the railML® 3.1 data format. In addition to linear and geometric positioning systems, it also allows a user to create screen positioning systems and visualization objects. The visualization module implementation can be seen in the *Figure 4*.

The use case-specific part includes concrete types of net entities representing various objects and properties which can be localized to the topological layer. The relevant data classes are specializations of the net entity class allocated within the core model at the interface with the use case-specific part. In order to ensure feasible data export and import, it is possible to design these classes and their attributes taking into account the structure of the railML® 3 exchange format. It is appropriate to focus on the geometry, functional infrastructure and protectively physical assets (not yet defined in version 3.1) substructures of the infrastructure subschema.

5. RAILML® 3 INFRASTRUCTURE SUBSCHEMA

RailML® [13] is an open source XML-based data format used in the railway branch. The railway markup language was originally designed to enable data exchange

in the scope of timetabling. Later on, the specifications [14] were extended to the subsystems of rolling stocks, infrastructure and interlocking. In recent years, attention has been focused especially on the third version development.

As regards the infrastructure schema, railML® 3 is essentially based on the RailTopoModel principles, whereas the implemented core model is extended to specific aspects mainly related to the net entity subsystem. The first public version of railML® 3 was released on 19 February 2019. [12] Noting the fact that railML® 3 respects the use case-oriented approach, it can be said that railML® 3.1 takes into account the following use cases:

- *Network Statement*
- *Schematic Track Plan*
- *Interlocking Module Engineering Data*
- *Simulation* [15]

Each of the above listed use cases is related to either the infrastructure or the interlocking schema or both. Depending on further implementation of other use cases, the particular schemas will be complemented with new aspects, especially new specific types of net entities and their attributes.

Regarding the railML® 3 file structure, the infrastructure schema is represented by the *infrastructure* element contained in the top-level *railML* element. The *infrastructure* element can contain several other subelements representing particular thematic views, as stated below:

- *Topology*
- *Geometry*
- *Functional Infrastructure*
- *Physical Infrastructure*
- *Infrastructure States*
- *Infrastructure Visualizations* [12]

The allowable structure has been gradually evolving during the the railML® 3 development and was established as described further. The topology module is

essentially based on the RailTopoModel topology subsystem. The corresponding *topology* element includes container elements for net elements, net relations and networks. Their subelements represent specific instances of the respective types of topological objects.

All available information about net entities is contained in the *geometry* and *functionalInfrastructure* elements, as the *physicalInfrastructure* element does not yet have internal structure defined. These modules also use the pattern of container elements with nested specific instances of net entities of its type.

The geometry module consists of horizontal curves, gradient curves and geometry points. These track geometry objects are specializations of the net entity class as defined in the RailTopoModel specification. Therefore, they can be localized to the topology layer and using coordinates of positioning systems.

Another group of net entities introduced by the railML® 3.1 infrastructure schema is represented by the functional infrastructure module containing various railway infrastructure facilities and characteristic (e.g. lines, operation points, tracks, switches, crossings, signals, platforms, speeds and track gauges) from a functional point of view. That makes it probably the most complex part of so far evolved railML® 3.

Next to the functional infrastructure, the module of physical facilities (in the sense of assets) should be developed at a later date, depending on the upcoming use cases. The *physicalInfrastructure* element does not yet have internal structure defined.

Furthermore, railML® 3.1 is able to express infrastructure states in relation to individual elements. That makes it more powerful tool than the current version of RailTopoModel which can only define one period of validity, as regards selected objects.

The last of the railML® 3.1 infrastructure modules deals with infrastructure visualizations. This module enables the infrastructure to be visualized by making it possible to display various objects on the screen using screen coordinates. [12]

6. POSSIBLE REPRESENTATION OF TURNOUTS USING THE RAILTOPOMODEL AND RAILML® 3.1 MODELING PRINCIPLES

A turnout – as the subject of data description – can be viewed from several different points of view. As regards the RailTopoModel and railML® 3 methodics, we can distinguish between the topological and the net entity approach. Concerning the railML® 3.1 version, it can be described either as part of a functional infrastructure or as an interlocking device. Prospectively, the physical infrastructure perspective should also be implemented. [9, 12]

6.1. TOPOLOGICAL APPROACH

Regarding the topological point of view, an ordinary turnout can be modeled using three interconnected linear net elements representing the tracks adjacent to the turnout. There is no net element to represent the turnout itself, at this level of detail. The turnout is only implicitly expressed from the functional perspective with the use of net relations connecting the net elements (see the subfigure *a*) of the *Figure 5*). Each net relations connects two net elements in the roles *A* and *B*. Each net element can be connected at the point with an intrinsic coordinate of the value of *0* or *1*. For each net relation, the attribute of navigability must be defined. The *navigability* attribute can reach the value of *AB*, *BA*, *both* or *none* expressing the direction in which the net relation is navigable. [9, 12] For modeling the ordinary turnout as described above, two net relations with navigability of the value of *both* and one net relation with navigability of the value of *none* are used.

It is also possible to express the turnout as a separate non-linear net element connected to three linear net elements. This kind of description can facilitate the localization process of the net entity expressing the appropriate technical equipment of the turnout. Nevertheless, there is no possibility to express the *navigability* attribute in an appropriate manner, as it cannot be distinguished in which direction to continue (or not) after reaching the non-linear net element (see the subfigure *b*) of the *Figure 5*).

The possible solution is to consider the turnout as a linear net element (see the subfigure *c*) of the *Figure 5*) or to proceed to an ever more detailed level in order to distinguish the individual branches of the turnout to be considered as two separate net elements (see the subfigure *d*) of the *Figure 5*). The approach of two separate branches allows to easily model directional conditions and other detailed features related to the turnout (especially in the case it is not desirable to express them for the switch as a whole). Nevertheless, the necessary number of net elements and net relations increases.

In general, the RailTopoModel net elements are supposed to be dimensionless. [9] Nevertheless, in terms of some use cases, e.g. as regards the railML® 3 format, the need to introduce the *length* attribute for the net element instances has arisen. [12] In order to assign the value of the *length* attribute to each net element correctly and precisely, it is desirable to disintegrate the turnout non-linear net element to three separated linear net elements interconnected at one specific point (see the subfigure *e*) of the *Figure 5*). The internal topology of every single turnout is similar to the first said topological example, then.

In addition, there are several possible interpretations of how his model could be explained. The specific point of interconnection may represent the point of the frog, the intersection of the continuous and branching track axis, the point of the track axis at

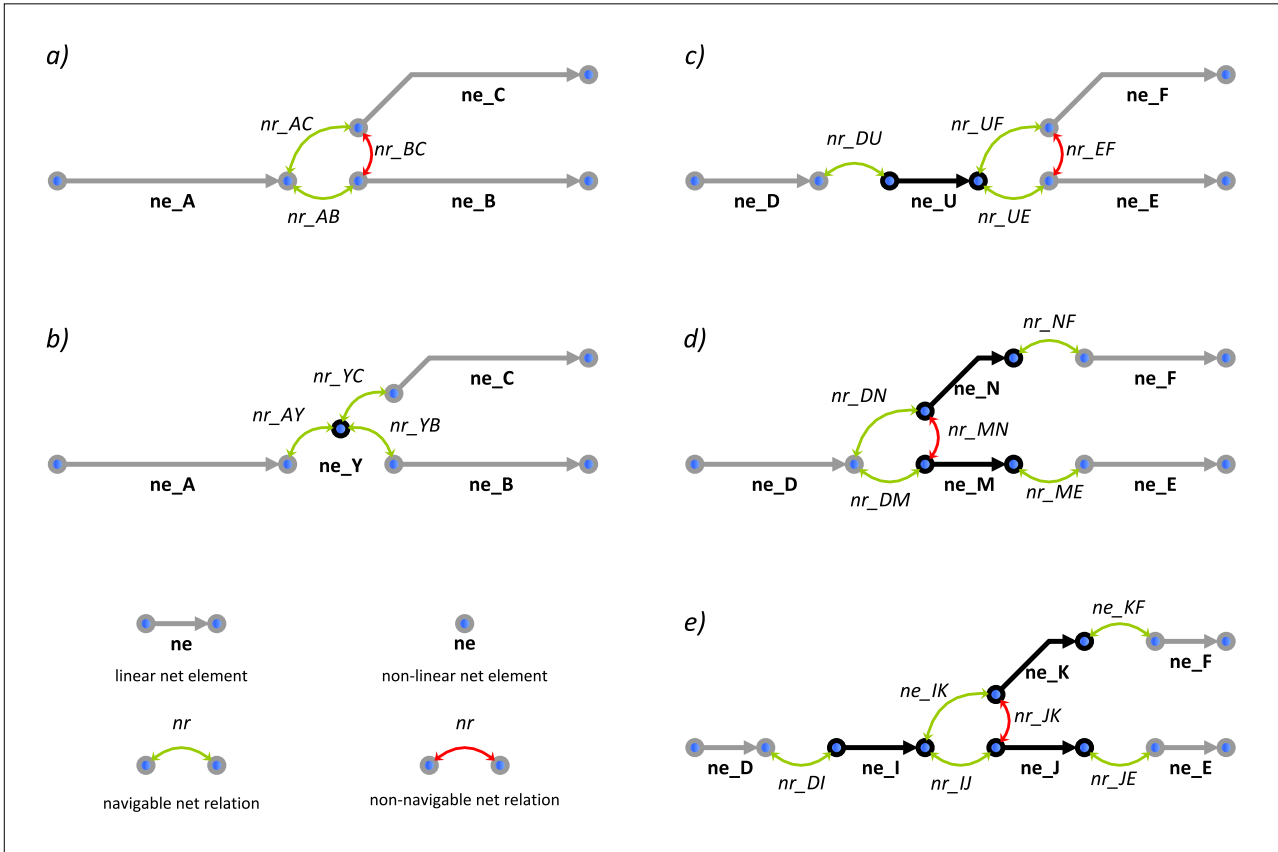


FIGURE 5. Possible expressions of an ordinary turnout using net elements and net relations

the level of the tip of tongue etc. Each of these specific representations has its significance for a particular use case and affects in which way the attribute values are assigned to particular net elements. In particular, this approach makes it possible to create predefined and easily repeatable patterns of internal topological representation of turnouts.

6.2. NET ENTITY APPROACH

While the topological point of view allows to express turnouts using net elements and net relations, the perspective of net entities makes it possible to describe them with factual parameters and to locate them to the topological layer and using external coordinates. As already mentioned, the RailTopoModel specifications itself state no specific types of net entities, as this is a matter of particular use cases. [9]

Regarding the railML® 3.1 infrastructure schema, there is a *SwitchIS* complex type defined in order to describe a turnout from the functional perspective. It serves to express the permissible structure of individual *switchIS* elements, nested in the *switchesIS* container element, being a part of the functional infrastructure view.

Together with the *Crossing*, *BufferStop* and *Border* complex types, the *SwitchIS* complex type is one of the specializations of the *TrackNode* complex type derived from the *FunctionalInfrastructureEntity* complex type. Further, this is a specialization of the *Enti-*

tyIS complex type which is derived from the complex type based on the *LocatedNetEntity* RailTopoModel class. [12] the “IS” substring was added to the designation of the complex type in order to distinguish between the infrastructure and interlocking net entity. For this time, we focus on the switch entity as part of the functional infrastructure view, accordingly.

The *SwitchIS* complex type is designed to express all applicable types of turnouts, i. e. including the slip turnouts. It may have the following attributes defined:

- *basedOnTemplate*
- *belongsToParent*
- *branchCourse*
- *continueCourse*
- *type* [12]

The *type* attribute allows to distinguish between the ordinary switch, outside curved switch, inside curved switch, single switch crossing and double switch crossing constructions. The *basedOnTemplate* and *belongsToParent* attributes are intended to contain a reference to other elements of the *SwitchIS* complex type (in the role of the template or parent in relation to the described instance). The *branchCourse* and *continueCourse* attributes can take the value of *left* or *right* expressing the direction of the particular branches.

These turnout branches can be described as separate elements nested in the *switchIS* element. The *leftBranch* and *rightBranch* elements are relevant for an ordinary (resp. outside curved or inside curved) switch. The *straightBranch* and *turningBranch* elements can be used up to twice within any *SwitchIS* instance expressing either of the switch crossing types. All the above-mentioned elements expressing any of the turnout branches are of the *SwitchCrossingBranch* complex type. [12] For this reason, they can be described by the following attributes:

- *branchingSpeed*
- *joiningSpeed*
- *length*
- *netRelationRef*
- *radius* [12]

The *branchingSpeed* and *joiningSpeed* attributes indicate the permitted speed of the described turnout branch in kilometers per hour with respect to the direction of travel (which may be relevant, for example, in the case of some types of ordinary turnouts). The *length* attribute expresses the length of the turnout branch in meters. The *netRelationRef* attribute allows the turnout branch to be assigned to one of the net relations from the topological layer. Finally, the *radius* attribute means the radius of the arc of the relevant turnout branch expressed in meters.

Alternatively, the directional conditions of turnout branches can be modeled independently of the relevant *switchIS* elements, using the *HorizontalCurve* complex type from the geometry view. The net entity representing a horizontal curve can be described by the *curveType*, *azimuth*, *deltaAzimuth*, *radius* and *length* attributes. A similar approach can also be used to describe the permitted speed in the switch branch. The speed can be expressed using the *speedSection* elements with the possibility of nesting an unlimited amount of elements referring to related speed profiles. Next to the maximum speed, the *SpeedSection* complex type makes it possible to express, if the described speed limit is temporary and signaled. [12]

7. ASSESSMENT OF THE USABILITY OF THE RAILTOPOMODEL AND RAILML® 3.1 MODELING PRINCIPLES FOR THE NEEDS OF HIGH-SPEED TURNOUTS DESCRIPTION

As regards the topological point of view, there is no difference between high-speed turnout and conventional railway turnout, if it is an ordinary turnout. Depending on the desired level of detail and functional requirements, any of the above-mentioned approaches to modeling the railway network topology can be useful. If necessary, the branches of high-speed turnouts

can be modeled in even more detailed level, which the approach of net elements connected by net relations seamlessly allows. These net elements of more detailed levels can be standardly aggregated in order to create the less detailed ones.

The only possible difference from the current state of the RailTopoModel and railML® 3.1 data format may lie in the fact that new values of the *descriptionLevel* attribute should be introduced if the more detailed topological description needed. Only the *macro*, *meso* and *micro* values are currently available, while the macro and meso levels are not sufficiently detailed to show turnouts. Although the IRS 30100 document [9] also outlines the nano level, it has not been clearly specified and included among the usual levels yet.

Expressing high-speed turnouts as net entities (if we focus on the functional infrastructure view) is more problematic. The RailTopoModel itself is not in charge to express so specific types of objects such as turnouts at all. [9] Although the railML® 3.1 format provides a relatively extensive and sophisticated apparatus for describing the basic characteristics of railway turnout, [12] it is still not sufficient to express all the geometric properties of high-speed turnouts properly. Some turnouts designed for high-speed operation, as seen in the the *J60-1:33,5-8000/4000/14000-PHS* turnout type example, have a relatively complicated geometric solution of the branches.

The single *radius* attribute is far from sufficient to describe all directional conditions. It would be necessary to define several new attributes to express required properties of the above-mentioned turnout type. Its turning branch consist of three different curves, two of which are transition curves with unstable curvature. Although such parameterization of directional conditions would be feasible, it would never be able to cover all possible geometric solutions of turnout branches (including those that may be developed in the future). Many of these parameters would be completely unnecessary when describing most other turnouts.

A more progressive approach is to extend the allowable internal structure of the *SwitchCrossingBranch* complex type elements by introducing nested elements expressing subsections of a turnout branch. Each such subsection could represent a segment describable by one curve type. Such segments could be described by similar attributes as the *horizontalCurve* elements. These elements make it possible to distinguish between different curve types, such as *straight*, *arc* and *clothoid*. [12] Nevertheless, a significant disadvantage lies in the fact that it is not possible to assign the radius of the arc at different positions of the horizontal curve directly. This problem arises especially when describing transitions curves. For this reason, it would be appropriate at least to replace the *radius* attribute of the *HorizontalCurve* complex type with two similar attributes expressing the radius at the start point and at the end point of the described curve.

Another solution for one of the railML® 3 upcoming subversions is not to describe turnouts in terms of geometry within the *switchIS* elements at all and use the description of directional conditions exclusively through the *horizontalCurve* elements instead. This solution would require modifying the horizontal curve description structure similar to the previous proposal. In addition, it would be appropriate to provide the possibility to assign individual horizontal curve to a specific turnout branch.

As another point, the turnout branch entity itself could only be described using an aggregated data item of minimal radius, as this is critical in many respects. In cases where this would be sufficient, such a value could be used as the only descriptive aspect of the geometry, if so described. It is always a matter of a specific use case as accurate and detailed description is required.

8. CONCLUSIONS

This article provided insight into the specifics of high-speed railway turnouts and the possibilities of their data modeling. When compared with the principles of the RailTopoModel and railML® 3.1 data format, it was found that there is no significant difference between a conventional ordinary turnout data model and a possible high-speed turnout model, as regards the topological point of view. Nevertheless, the functional infrastructure interpretation of a turnout – as offered by the railML® 3.1 infrastructure model – is insufficient, in this respect. Some high-speed turnouts require the geometry to be modeled in a much more precise way.

If it is decided to include such turnouts in the Multipurpose Railway Infrastructure Model, there are several ways to implement it. We can either increase the number of attributes or extend the internal structure of the turnout data description or solve the geometry of the turnout separately and reference it using external links. These approaches would allow an accurate and detailed description of a high-speed turnout. It is also possible to use only an aggregated data item expressing the value of the minimum radius in each turnout branch instead of the detailed description of its geometry. Such a simplified approach is appropriate especially in less demanding use cases.

REFERENCES

- [1] M. Hamadache, S. Dutta, O. Olaby, et al. On the Fault Detection and Diagnosis of Railway Switch and Crossing Systems: An Overview. *Applied Sciences* **9**, 2019. <https://doi.org/10.3390/app9235129>.
- [2] M. Hamarat, S. Kaewunruen, M. Papaalias. A Life-Cycle Cost Analysis of Railway Turnouts Exposed to Climate Uncertainties. *IOP Conference Series: Materials Science and Engineering* **471**, 2019. <https://doi.org/10.1088/1757-899X/471/6/062026>.
- [3] M. Leso, D. Kamenický, P. Koutecký, A. Hlubuček. Dopravní sál FD – železniční laboratoř pro výuku i výzkum. *VTS Správy železnic* **1**, 2019.
- [4] M. Kapitsa, D. Bobyr, A. Desiak. Determining Permissible Speed of Tilting Train in Curved Track. *Science and Transport Progress Bulletin of Dnipropetrovsk National University of Railway Transport* **70**: 29–40, 2017. <https://doi.org/10.15802/stp2017/109537>.
- [5] M. Lindahl. *Track Geometry for High-Speed Railways*. Tech. rep., Royal Institute of Technology, Stockholm, 2001.
- [6] H. Ossberger. *Modern Turnout Technology for High Speed* [online]. https://www.unece.org/fileadmin/DAM/trans/doc/2016/TER/Heinz_Ossberger_Modern_Turnout_Technology.pdf.
- [7] L. Raif, B. Puda, J. Havlík, M. Smolka. Design of High-Speed Turnouts and Crossings. *IOP Conference Series: Materials Science and Engineering* **236**, 2017. <https://doi.org/10.1088/1757-899X/236/1/012044>.
- [8] UIC. *Feasibility Study UIC RailTopoModel and data exchange format* [pdf], 2013. http://www.railtopomodel.org/files/download/RailTopoModel/270913_trafiT_FinalReportFeasibilityStudyRailTopoModel.pdf.
- [9] UIC. *RailTopoModel v1.0* [pdf], 2016. http://www.railtopomodel.org/en/download/irs30100-apr16-7594BCA1524E14224D0.html?file=files/download/RailTopoModel/180416_uic_irs30100.pdf.
- [10] railTopomodel. *RailTopoModel* [online], ©2020. <http://www.railtopomodel.org>.
- [11] UIC. *RailTopoModel v1.1* [online], 2017. <https://uic.org/rail-system/railtopomodel>.
- [12] railML. *railML schema version 3.1* [xsd], 2019.
- [13] railML. *railML* [online], ©2002–2020. <https://www.railml.org>.
- [14] railML. *The railML subschemas* [online], ©2002–2020. <https://www.railml.org/en/user/subschemas.html>.
- [15] railML. *Use Cases* [online], ©2002–2020. <https://www.railml.org/en/user/use-cases.html>.