MODEL OF A RAILWAY INFRASTRUCTURE AS A PART OF A MESOSCOPIC TRAFFIC SIMULATOR

Radek Novotny[(a)], Antonin Kavicka[(b)]

[(a),(b)]Faculty of Electrical Engineering and Informatics, University of Pardubice

[(a)]radek.novotny3@student.upce.cz, [(b)]antonin.kavicka@upce.cz

ABSTRACT

In order to investigate or to optimize railway traffic and infrastructure within complex railway nodes (stations and junctions), computer simulation methods are frequently utilized. Relevant simulation models can apply different degree of granularity, which defines the level of details within the computer simulating system with regard to the particular object of investigation.

For railway traffic simulators performing investigations on a mesoscopic level, it is important to support rapid prototyping of a target model in order to shorten the life cycle of simulation projects/studies. From that point of view, relevant track infrastructure submodels are supposed to be created quickly and efficiently. The contribution deals with the issue of rapid prototyping and verification of infrastructural submodels which belong to mesoscopic simulators focused on the traffic within railway stations and junctions. An appropriate editor of infrastructure provides a variety of sophisticated functionalities that support efficient and partially automated constructions of track layouts for different scenarios of simulation experiments.

Keywords: track infrastructure model, railway station, mesoscopic traffic simulation

1. INTRODUCTION

One of the main prerequisites for conducting simulations of railway traffic is to have a suitable model of railway infrastructure corresponding to the investigated segment of a railway network at disposal. The structure of the model does not have to suit the specific simulator or simulation experiment accurately, and thus even if the model is available, its modification in the professional graphic editors can be very laborious and time consuming (especially if the infrastructure model is processed on a real scale).

For these reasons, a strong motivation to develop an editing tool, which would significantly simplify, partially automate and accelerate the construction of infrastructure models, exists. The mentioned editor must meet the needs of the mesoscopic traffic simulator within railway nodes, which does not require an infrastructure model with such accuracy of metric ratios within the model of track infrastructure as microscopic simulators do, but it must contain a level of detail sufficient for investigating the throughput of railway stations or the effects of delay on railway traffic.

2. CURRENT STATE

Nowadays there are many simulation tools that use their own approaches to constructing models of a railway infrastructure. Villon, RailSys and OpenTrack can be mentioned as examples of such simulation tools (Simcon 2014, Huerlimann and Nash 2010, Radtke and Bendfeldt 2011). RailSys and Villion are simulation tools that utilize modeling of a railway infrastructure on the microscopic level and therefore the infrastructure model is constructed in a real scale. In contrast, OpenTrack uses schematic visualization, which allows it to partially accelerate the preparation of the final model. More complex railway objects, for example switches and crossings, must be created by merging individual segments.

The following two different implementation concepts of editors are used for the above mentioned simulation tools:

- implementation within the simulator (OpenTrack),
- secondary software (RailSys and Villon).

The simulation tools, mentioned above, describe the railway network using graph representation and they allow to manage properties, such as gradients, speed limits, signals, etc, and objects on the graph's edges and nodes. But only OpenTrack supports the latest version of the railML standard (RailML 2015) as a uniform prescription for storing and migrating the infrastructure model.

3. APPROACHES TO VISUALIZATION

From the viewpoint of the editor (called *TrackEd - Mesoscopic Track layout Editor*) in development by the author of this article, it is necessary to distinguish between a *visual scene* (Figure 1) presenting railway infrastructure in a graphical form and a *data scene* defining *numeric* values related to metric/topological and vertical alignments (attributes). The *data scene* is composed of appropriate data structures utilized by relevant calculations of train movement dynamics (Iwnicki 2006).

For the needs of a different approach to visualization, a higher degree of abstraction is applied. It enables significantly faster creation of the railway infrastructure model, which is based on the schematic visual depiction of the track infrastructure. Thus, from the viewpoint of visualizations there is no requirement for graphical

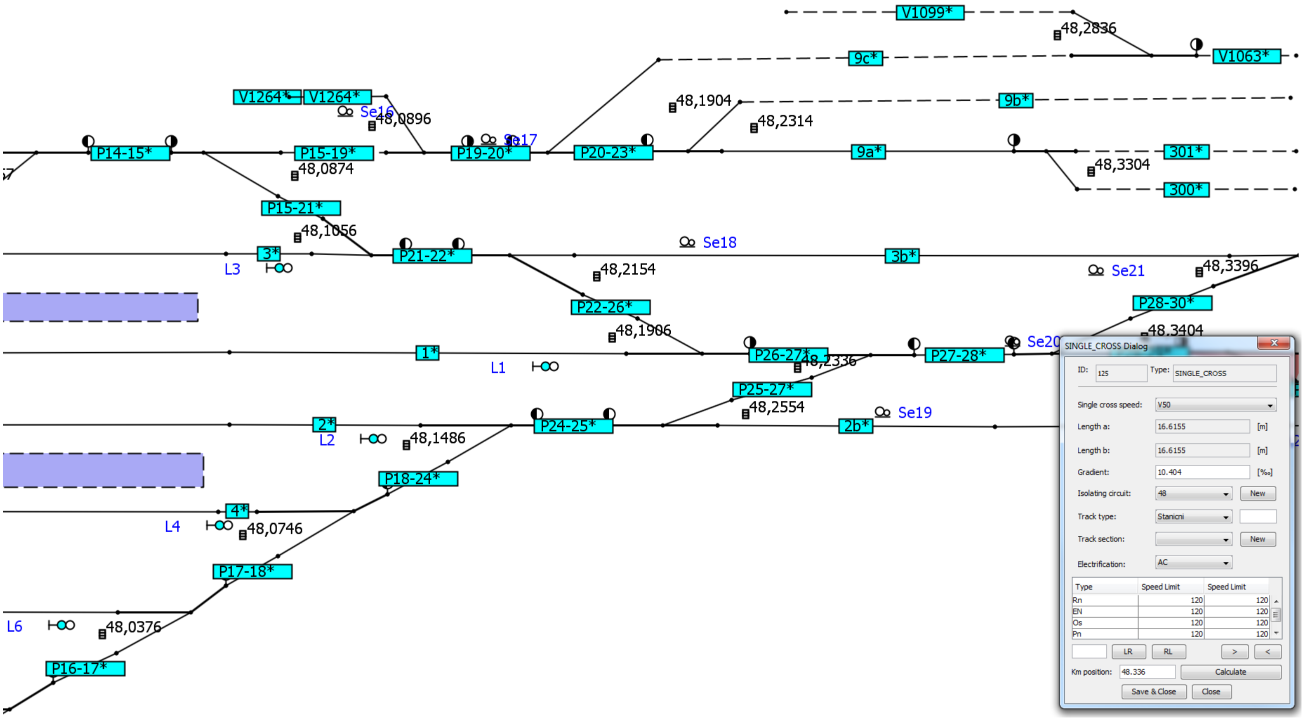

Figure 1: Schematic plan of a track infrastructure within an illustrative station

depiction of the infrastructure in a scale. On the other hand, parameterization of infrastructural objects within relevant data structures can support realistic calculations concerning train rides. The mentioned data structures store relevant entries which specify topological, metric, and vertical alignments within the track layout and mirror real situation connected with the investigated station/junction.

While creating the final model of a railway infrastructure in the *TrackEd* editor, isolated circuits (according to an installed interlocking system), locations of signaling devices, electrification of tracks and useful lengths of rails, and finally the speed limits on individual railway segments are also realistically concerned.

Schematic visual depiction of track layout is based on using polylines, which means that more complex shapes, like track arcs or track crosses, are not depicted in a highly authentic way. The concept of schematic representation of railway infrastructure with the support of visually deform individual track objects within the infrastructure without losing or changing the real topological, metric, or vertical alignments defined within the *data scene*, accelerates the constructing of the model.

4. OBSERVED OBJECTS

The typical model of railway infrastructure within the stations/junction is composed of the following basic elements (Kubat 1999, Jirsak 1979):

- tracks,
- switches,
- crossings,
- semaphores,
- limit signs for train positions on tracks,
- platforms,
- and alternatively, station buildings.

Basic atomic objects serve not only for schematic visualization, but also for describing the physical infrastructure, especially the real metric and topological ratios in the *data scene*.

Data structure of the undirected graph implemented in operation memory and composed of two types of objects, nodes and edges, type is used as a representative of a railway network. The edge represents an atomic track section, hereinafter referred to as *segment*, containing the basic characteristics of length, gradient, arc, speed limits, isolated circuits, signals, track type, electrification, etc. Individual nodes of the graph then represent a link between such segments. The *node* is an important object which describes not only the position of segment endpoints in the schematic plan, but it also contains the coordinates of their real positions in the railway infrastructure in the background (Lewis and Denenberg 1997).

5. PROTOTYPING

To accelerate the construction of the infrastructure model, it is possible to use predefined *prototypes* of selected track objects (e.g. switches, crossings, or aggregated track sections). Prototype can be seen as a part of an infrastructure comprised of elementary objects described in the previous section.

As it was said, the *atomic prototype*, used to construct the simplest track object representing one edge in the

undirected graph, is a *segment*. Decomposing the infrastructure into segments allows modelling those parts of railway infrastructure, in which its properties change (gradient, radius of the arc, speed limit, etc.). To ensure automatic calculations common for railway traffic, it is possible to aggregate a set of adjacent segments into the track. This is especially needed for tracks in stations which change the gradient in some places, but in the context of railway traffic, an example can be a calculation of useful length, they are designated as a one track.
Other typical representatives of prototypes are *switches* and *crossings*, which consist of group of segments. In the context of future development, will be implemented even more complex aggregation of prototypes mentioned above, i.e.a part of development of switches composed of atomic segments, switches and crossings.
Each selected prototype is realistically parameterized, i.e. it is related to a table of standardized dimensions (for example a switch prototype is connected with standardized maximal allowed train speed on the secondary branch) (Kubat 1999). It means in fact that lots of predefined track objects can be utilized without the need to customize their parameters and massive manual specifications are not needed.

6. OBJECTS POSITION

The *TrackEd*, the railway infrastructure editor, as already mentioned, distinguishes between the mesoscopic visual scene and real parameterization of the *data scene*. The mentioned real parameterization is important for e.g. signals, location of which is specified by real coordinates and not by the distance from the endpoint of the segment it belongs to. For this reason, each node contains an attribute of real position within the railway infrastructure. In the *TrackEd* editor, so called stationing (localization) for the input of node position in the *data scene* is supported. It is such approach, in which each node contains a kilometric position from the start of the railway line (start of railway line corresponds to a zero kilometric position). Stationing in the Czech Republic serves for unambiguous determination of the position of all objects in the railway infrastructure (Kubat 1999).
In aggregated prototypes forming a continuous subgraph of the railway infrastructure, which consists of nodes and edges (e.g. switch and crossing), users simply assign a location to only one node. The reason for it being the calculation of kilometric positions of the remaining nodes based on the dimensions of the prototype listed in table of standardized dimensions (Fridrich 2008, Kubat 1999). In simple switches is it a node, which indicates the start of the switch in the setting-out diagram (Figure 2). In contrast, the location is determined only for the node which represents a point of the intersection for crossings.

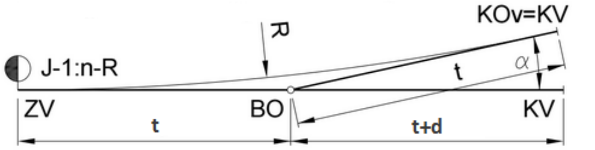

Figure 2: Setting-out diagram of a simple switch (Fridrich 2008)

In practice, there are other approaches to entering real positions of objects than stationing. An example is the use of real GPS coordinates or real coordinates of different geographical coordinate system. To use these approaches, it is necessary to have a very detailed source model to build the schematic plan of a railway infrastructure and, as already mentioned in the introduction, it does not have to be available.

7. TRACK ROUTES

Within the simulation, the train movements are closely connected with track routes. Those routes can be by either dynamically calculated (during a simulation experiment) or pre-calculated before the simulation starts. The train routes can differ for various types of trains. The *TrackEd* editor supports semi-automatic calculation of a complete set of primary and alternative train routes for all possible train transfers within the investigated track layout. In other words, each train route consists of segments forming a continuous subgraph of the railway infrastructure and each train is assigned a set of train routes with added priority. The priority is used during the simulation experiment to select a train route, which is not occupied or built for another train, for the next train movement (Hansen and Pachl 2008).
Filtering mechanisms by specified criteria, such as the maximum route length, allowed gradient, the radius of an arc, or electrification, can be applied to a selection set of train routes. In addition, for each route we can define the maximum allowed speed for a passing train in percentage to the maximum speed of each individual segment of the route. This corresponds with real train movement on track layout, which does not move by speed corresponding to the maximum design speed of each element of the railway network.
From the train movement, especially in the station, it is clear that each train can be assigned a large set of train routes. In the station, the train is assigned a route to the platform through the input development of switches and a route from the platform to the output development of switches. Because the same types of trains frequently occupy almost always the same platforms or station tracks thus use the same routes, it would be very inconvenient to assign these trains the input and output routes individually. For this reason, the train routes can be divided into groups, and before starting the simulation we can assign a group of all primary and alternate routes from the beginning of the train's movement to the end.
Such an approach enables a final control of the whole system of train routes by a professional - railway

technologist. Those pre-calculated and checked train routes are then used during traffic simulations.
From the view point of determining the occupancy of train routes during simulation experiments, mainly the effect on the train movement in the station, an automatic calculation of useful lengths of station tracks is applied. The user is then informed about the maximum length of the train that can stand on the track element (Kubat 1999).

8. PLATFORM

One of the most important objects of railway stations, that should not be overlooked in the railway infrastructure editors, are platforms, because they have an influence on the movement of the train in the station. Platforms are not only important objects for visual inspection of the railway station model and an important factor for technologist that creates train routes, but they also directly affect the realization of the train arrival dynamics on the station track or in other words, they define the border place for the train to stop.
In the *TrackEd* editor, each platform is always associated with a set of segments that form the platform track. For the needs of a mesoscopic railway traffic simulator, the start and the end position (border places) are parameterized for the platform as for any other objects of the infrastructure.

9. MIGRATION OF THE MODEL

An important part of every transport networks editor, in our case the railway network, must be the ability of the implemented product to store or further edit an unfinished design of an infrastructure model.
In the *TrackEd* editor, the resulting (visual) schematic plan, reflecting a given version of track infrastructure within the station, can be either further processed (i.e. another layout version can be formed), or exported into *MesoRail* - mesoscopic simulation tool currently in development. For such migration of the infrastructure between two different products, it is necessary to choose the type of migration that will be universal, easily validated, and quickly changeable in case of extended requirements for the detail of the infrastructure. For these purposes, transmission using a XML file, which can also be viewed and possibly edited in any text editor, was chosen.
During the development our own custom template, inspiration for which is the universal standard railML that prescribes uniform templates for saving the infrastructure in XML (RailML 2015), has been created. Such standard provides many XML templates, which can be used also for specifications of railway infrastructure. Some of railML-templates inspire our own template used within *TrackEd* editor (direct utilization of railML-templates was not suitable, because they are excessively complex and do not obtain any elements needed for schematic plans).
The custom template is designed in a way, in which any change of object properties of the infrastructure has little influence on the template structure and the mechanisms of its processing in the editor. The template is divided into separate blocks that describe different parts of the infrastructure model, i.e. nodes, segments, track routes, and platforms. For the needs of the editor, the template structure enables to save prototypes, all of its elementary objects (nodes and segments), in such a way, in which the editor is able to identify and correctly put them together during their import into the editor. This is especially important in situations, in which we need to save an unfinished model, which we want to modify in the future.
Nowadays, requirements for interoperability of software applications are more and more prominent. Such requirements are meet by the railML standard, which can be implemented in a mesoscopic editor in the future and therefore use this editor with other tools than *MesoRail*, for which it has been primarily developed.

10. THE CONSEQUENCES OF A SCHEMATIC PLAN

The consequence of using the concept of a schematic layout plan for the needs of mesoscopic traffic simulations is realization of the train movement on a deformed infrastructure. That means that the graphic representation of a train, or its parts, always has to dynamically conform to the track element (with a given degree of deformation) on which the train is currently moving. The cause is, as already mentioned, the difference between the visual presentation of the track layout and metric and topologic ratios set in the *data scene*. Such consequence can be partially eliminated by the user when creating the infrastructure model.

11. MODEL VALIDATION

From the perspective of the simulator every inaccuracy in the infrastructure model can be the cause of distorted or bad results of simulation studies. The most common reason is an insufficiently checked model of the railway infrastructure in the editor before its final importing into the simulator. Inaccuracies, such as incorrectly input gradient, radius of the arc, or speed limit of every track elements, are not undebuggable in the simulator. Therefore, a strong emphasis to implement sophisticated functionalities in the editor for detecting coarse and even minor faults of this type exists.
Functionality for validating the infrastructure model can be divided into the following groups:

- visual support,
- automated control.

11.1. Visual support

For visual inspection of the constructed infrastructure model, we can use *visual views*, which emphasize user-defined groups of attributes for the relevant track objects. In the *TrackEd* editor, two types of visual views are supported. They are visual views that highlight nodes with incident edges (segments), for which the gradient, radius of the arc, and the maximum allowed speed for the selected train for both directions of the move change. The second type is visual views that color code individual segments according to their values set in the *data scene*.

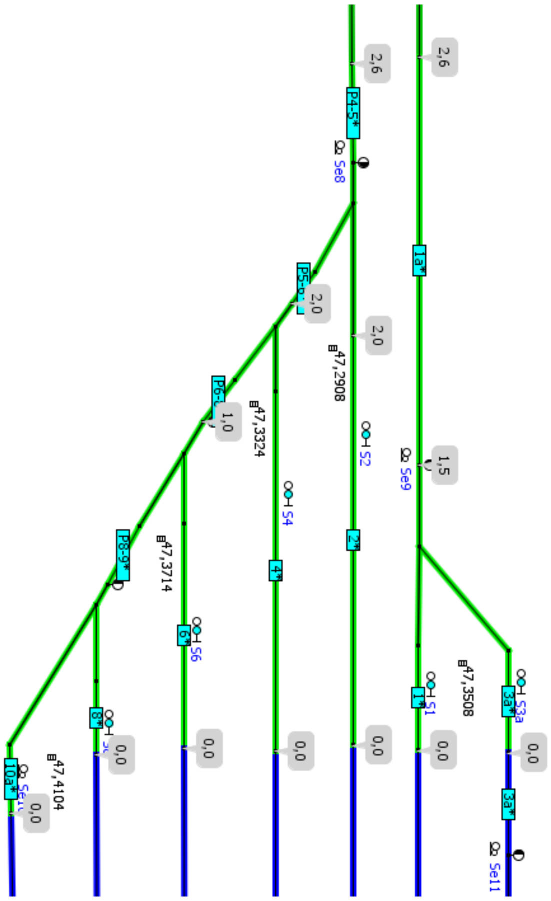

Figure 3: Visual view of the gradient changes

With this support, the user has an overview of changes across the infrastructure using one selected visual view (Figure 3).

11.2. Automated control

In the *TrackEd* editor an automated control of the result infrastructure model, in addition to the visual views, is supported. It represents a group of functionalities which serve to detect gross and informative errors in the model. A gross error is such an error which has a major effect on the simulation study. A list of gross errors is as follows:

- localization of all nodes,
- localization of neighboring nodes,
- localization corresponding to the segment length,
- isolated circuit in all segments and their adjacency,
- continuity of the train routes.

On the other hand, an informative error is the absence of a warning signal or inappropriately chosen type of a signal device on the track element and their incorrect placement from the beginning of the isolated circuit according to the standards or missing limit of switches (Kubat 1999).

Results of the automated control are always shown by the respective track element (Figure 4) on which they are discovered. It gives the user the possibility of a quick change without having to search for it.

Verification of the concept of creating a railway infrastructure model was conducted above testing schematic plans of selected railway stations in the Czech Republic. The resulting models were validated with the official schematic plans used by *The Railway Infrastructure Administration* (SZDC), management of the railway infrastructure in the Czech Republic, in mind.

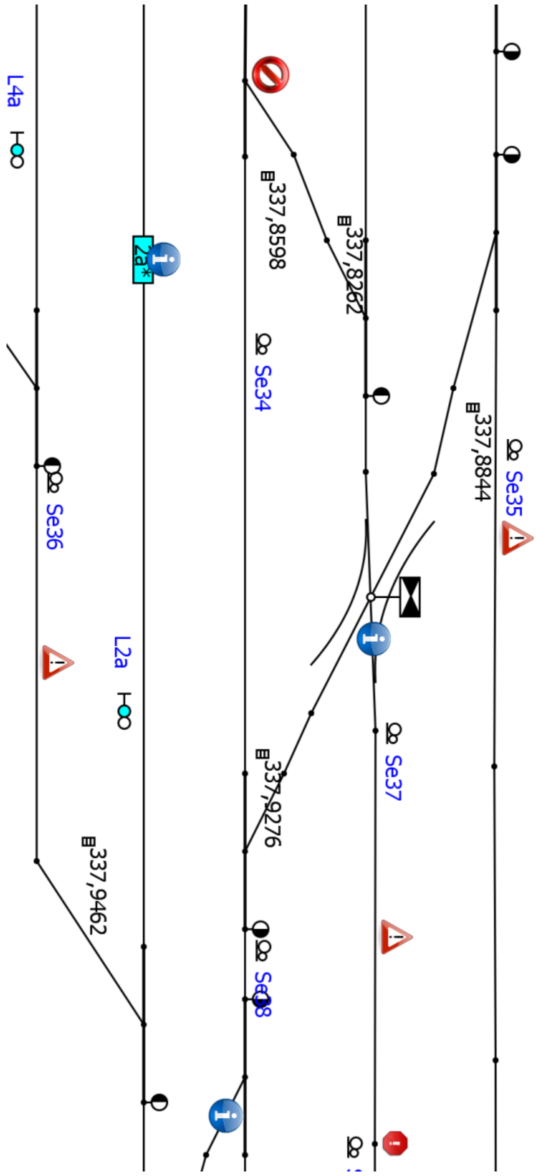

Figure 4: The result of automated control

12. PERSPECTIVE OF DEVELOPMENT

The perspective of further development is related to certain extensions of *TrackEd* editor. Within the frame of a new editor mode the special constructions of scalable track layout models are supposed to be supported. It means that the infrastructural model can combine submodels built on different levels of details (mesoscopic and macroscopic) (Hansen and Pachl 2008).

Another perspective option is to extend the functionality of the editor by automatic transformation of the designed model, or rather its visual schematic plan, into a plan in a real scale by using metric ratios from the *data scene*. The benefit of such transformed model is brought especially for users performing simulation experiments. User will have a better idea about the actual metric properties behind the scene (eliminates the effect of deformed infrastructure to animate the movement of the train).

13. CONCLUSION

The developed *TrackEd* editor is used to construct railway infrastructure models by using a schematic concept of visualization and thus distinguishes between *visual scene* and *data scene*.

It applies accelerating and innovative practices while constructing the infrastructure model, especially *rapid prototyping*, mainly if we do not have a digitized plan of tracks layout in real scale of the investigated segment of a railway network (for example in AutoCAD).

The editor also supports many sophisticated features which help automate some phases of the methodological approach focused on creating track layout models. An example is an automated principle of calculating train routes for different types of trains. In the editor, support functionalities helping to validate the model before it is imported into the railway traffic simulator are also kept in mind.

REFERENCES

Iwnicki Simon, 2006. Handbook of railway vehicle dynamics. BocaRaton: CRC/Taylor.

RailML, 2015. RailML: The XML interface for railway applications. RailML.org. Available from: http://www.railml.org/index.php/home.html [Accessed 11 July 2015].

Simcon, 2014. Villon. Simcon. Available from: http://www.simcon.sk/en/tools/villon [Accessed 11 July 2015].

Radtke Alfons, Bendfeldt Jan-Philipp, 2011. Handling of railway operation problems with RailSys. Institute of Transport, University of Hanover, Germany. Available from: http://www.railway-research.org/IMG/pdf/235.pdf [Accessed 11 July 2015].

Huerlimann Daniel, Nash Andrew, 2010. OpenTrack - Simulation of Railway Networks: User's ManualVersion 1.6. ETH Zurich: Institute for Transportation Planning and Systems.

Fridrich Karel, 2008. SZDC S3 rule – Switches and railway station. Prague: CVUT. Available from: http://kzs.fsv.cvut.cz/4/zst3/zst3_vyhybky.pdf. [Accessed 11 July 2015]. [in Czech]

Lewis Harry, Denenberg Larry, 1997. Data Structures and Their Algorithms. Boston: Longman.

Kubat Bohumil, Fliegel Tomas, 1999. Railway buildings 30. Prague: CVUT. [in Czech]

Jirsak Zbynek, 1979. Railway stations and junctions. Bratislava: Alfa. [in Czech]

Hansen Ingo, Pachl Jorn, 2008. Railway timetable and traffic. Hamburg: Eurailpress.