DESIGN AND DEVELOPMENT OF A MESOSCOPIC SIMULATOR SPECIALIZED IN INVESTIGATING CAPACITIES OF RAILWAY NODES

Roman Diviš^(a), Antonín Kavička^(b)

^{(a),(b)} Faculty of Electrical Engineering and Informatics, University of Pardubice

^(a)roman.divis@student.upce.cz, ^(b)antonin.kavicka@upce.cz

ABSTRACT

To investigate capacities of railway stations and junctions, analytic methods or methods of computer simulations can be used. Analytic methods are slowly getting outdated. Classic microscopic simulations offer very good results but the life cycle of simulation studies is relatively long. Railway companies nowadays require a lot of traffic simulations, which are supposed to be carried out in shorter periods of time than are typically related to microscopic simulations. A simulation tool MesoRail, now in development, works on a mesoscopic level that abstracts away from some details affected by the microscopic simulations. An appropriate choice of the level of abstraction enables to achieve still very good results. At the same time, the tool simplifies the stage of building a simulation model. MesoRail tool is mainly specialized in investigating the capacities of railway stations and junctions, but it can also be used for studying other traffic problems concerning railway transport.

Keywords: mesoscopic traffic simulation, railway station, capacity

1. CAPACITY OF RAILWAY NODES

Capacities of railway nodes can be, in principle, investigated using two approaches: (i) analytical methods or (ii) methods based on computer simulation. Analytical methods utilize the theory of probability and mathematical statistics in order to establish formulas thanks to which it is possible to analytically calculate the throughput of individual railway nodes and their parts. Let us mention two analytical methods for determining capacities of railway infrastructures: (i) directive SZDC D24 - The regulation for determining capacities of railway lines applied by the state company*The Railway Infrastructure Administration*in the Czech republic (SZDC 2009) and (ii)*UIC Code 406 – Capacity*developed by*UIC - International Union of Railways*(UIC 406) (ETF 2013).

1.1. Analytical methods

Analytical methods provide a result in the form of an overall indicator - the *degree of occupancy* of individual parts related to a railway infrastructure. On the other hand, simulation methods enable to observe variety of operating characteristics during the simulation - e.g.

values of delays of individual trains within particular parts of a railway station, real occupancy of individual parts of railway segments, etc. Deploying analytical methods is getting outdated nowadays, their utilization is relatively fast, but the informative value of their results is significantly lower in comparison with the outcomes from simulation methods.

1.2. Methods of computer simulation

Simulation methods enable to test thoroughly the throughput of stations under various operational conditions. Nowadays, there is a variety of high quality tools for microscopic simulations specialized in railway traffic, which can support investigations connected with capacities of railway infrastructures. As examples can be mentioned tool *Villon* (Simcon 2014), *OpenTrack* (OpenTrack Railway Technology Ltd. 2015) and *RailSys* (RMCon 2015). Applying computer simulation, it is possible to test the behaviour of the investigated railway node according to various operational scenarios. Thus, it is possible to test normal traffic without delays, or to test the sensitivity of a train schedule for various combinations of delay values from several directions etc.

2. SIMULATOR OVERVIEW

The next chapter provides some examples of simulators, which specialize in rail transport simulation, mainly the microscopic simulation tools.

2.1. Microscopic simulation

Typical representatives of microscopic simulation tools include *Villon* (Simcon 2014). Infrastructure is prepared in scale to reality and it contains very detailed data about individual rail segments. The actual simulation tool itself further requires defining characteristics of individual wagons, locomotives and trains created from them. Controlling a train in the railyard is defined by a network graph which consists of individual activities the train gradually performs. Technological activities, train paths, and other elements are easily parameterized. Thus, a wide range of train control during the simulation is provided, but the preparation of the simulation is difficult and lengthy.

A tool similar to *Villon* is the simulation tool *RailSys* (RMCon 2015). It realizes a very detailed microscopic

simulation. Although the railway infrastructure is depicted in a scheme, it still contains all the necessary underlying data. *RailSys* is well applicable for extensive simulation studies, which can automatically detect any capacity problems in individual sections of the rail network.

Another used simulation tool is *OpenTrack* (OpenTrack Railway Technology Ltd. 2015). Like *RailSys* it uses infrastructure defined in a scheme while preserving real underlying data. Train control is realized by defining a train schedule, which includes train stops, and other various parameters. The same solution is applied to train joining and train splitting, and waiting for other trains.

All the above mentioned tools support online animation of occupying the rail network by trains or the movement of trains, and they also support generating postsimulation protocols. Commonly generated outputs include tachograms of trains, occupation of individual track sections, overall statistics, and more.

2.2. Mesoscopic simulation

The disadvantage of using a computer simulation is that building a quality microscopic simulation model and conducting simulation experiments is a rather time consuming process. In addition, railway companies nowadays require that the infrastructure capacity investigations take shorter time than the current approach which uses microscopic simulations. Using the mesoscopic approach to simulation provides the ability to abstract away from several details and thus to shorten the life cycle of simulation studies and in result satisfy the needs of railway companies.

The results of mesoscopic simulations cannot provide the same level of accuracy as the outcomes of microscopic simulations, but by choosing an appropriate level of abstraction it is possible to achieve satisfactory results for practical usage.

Nowadays, models of mesoscopic transport systems are already fairy well adapted and a number of tools to simulate road traffic also exists. A summary of simulation tools and different mesoscopic techniques can be found in, for example, article by Xu, Song, Weng and Tan (2014).

In the field of railway transport systems, the usage of mesoscopic simulations is currently mainly in its investigation phase. Stallybrass in his article describes the use of mesoscopic simulations in determining the optimum interims for preparing train graphicons (Stallybrass 2014). Mesoscopic simulation of freight trains using the event-oriented simulation tool *SIMUL8* is described by the authors Marinov and Viegas (2010).

Due to the lack of availability of a suitable tool which would be appropriate for determining the throughput of railway stations, we decided to develop our own simulation tool called *MesoRail*.

3. THE MESORAIL TOOL

MesoRail is a mesoscopic simulator of train traffic with support of rapid prototyping of railway infrastructure and fast specifications of traffic scenarios. *MesoRail* is being developed in the programming language of Java. Mesoscopic level of simulation enables to model important aspects on the same level as in microscopic simulations, and at the same time it allows to apply higher level of abstraction to some aspects. Such approach enables to achieve reliable results from simulation experiments without any significant loss of their informative value.

The main problem is to find a suitable compromise and the degree of abstraction for individual parts of the simulator in order to achieve the desired result. To determine the appropriate abstractions, consultations with railway experts were used. The following abstractions are implemented within the *MesoRail* simulator (i) schematic depiction of the railway infrastructure, (ii) simplification of train dynamics calculations, (iii) mobile human resources (working groups) are not considered, (iv) service technological procedures provided to the trains are not considered.

The simulation tool *MesoRail* will find its application mostly in determining the throughput of railway nodes, therefore it is essential to meet the following requirements - (i) rapid infrastructure prototyping, (ii) real train ride characteristics, (iii) respecting the functionality of track signal and interlocking systems, (iv) considering station and railway line intervals, (v) simulating deterministic and stochastic train flows, (vi) animation outputs during the simulation experiment, (vii) post-simulation statistics and graphic protocols. The provided list represents only the basic requirements expected from the simulator. Based on this list it is necessary to establish the levels of abstraction for individual components.

3.1. Phases of development

Developing such a complex simulation tool is not a trivial matter and it takes quite a long time. Individual development steps are separated into stages and development takes place gradually over specified stages.

At the beginning of development, it was necessary to carefully specify the desired properties of the simulator and to determine the appropriate abstractions that move the microscopic simulation to the mesoscopic level on that basis. At this stage, consultations with railway experts with considerable experience with microscopic simulation of rail transport were found very useful.

After completing the basic specification of requirements and abstractions, it was necessary to design the architecture of the simulator – to select the method of simulation, used libraries, file formats, and to specify concrete requirements for train entities and railway infrastructure.



Figure 1: Train Deformation Example (single train placed and rendered in different parts of railway infrastructure)

In the following stages, specification of requirements for the *MesoRail* simulator were completed and the gradual implementation of various components of the simulator was scheduled. Individual partial results of simulations were verified and validated to ensure correct operation of the simulator and its credibility.

At the present stage of development we focus on building complex functionalities. It is already possible to performs simulations of simple railway nodes and obtain results for evaluation. However, it is necessary to further establish support for intelligent behaviour of trains in complex railway nodes.

3.2. Railway infrastructure

While simulating railway traffic, the infrastructure the trains move on plays a very important part. To prepare the railway infrastructure for a microscopic simulator is a relatively lengthy process. Creating an infrastructure and defining all parameters (speed restrictions within the infrastructure, vertical alignment, location of light signals for traffic directions, etc.) takes a considerable amount of time. Sometimes, the necessity to prepare an infrastructure in a particular scale can be a problem, as input data can be missing or containing faults.

For these reasons, a concept of schematic depiction of rail infrastructure with the support of rapid prototyping of infrastructural elements is proposed. Switches, railway crossings, and other more complex track elements are prepared as prototypes that can easily be parameterized and inserted into the infrastructure. That way it is possible to quickly build an infrastructure that can mirror real environment in detail, or it is possible to omit some parameters and leave standardized default values if needed.

The created infrastructure maintains the precision in the matter of meters. Unlike the microscopic simulations it is possible to prepare such infrastructure in substantially shorter time. For more details about infrastructure and its editor *TrackEd*, see the article by Novotný.

3.3. Train driving dynamics

Train moving on a track infrastructure corresponds to real riding properties, i.e. train and track railway characteristics are respected (Iwnicki 2006, Diviš 2014). However, in the *MesoRail* tool it is possible to abstract away from some parameter,s as their influence on the train or on its speed is insignificant. Such parameters are wind resistance in a tunnel or effects of transition curve on the train ride. Defining such parameters would prolong creating a simulation model in comparison with the insignificant effect on the riding time of trains.

The train driving dynamics is described by a differential equation (Diviš 2014), which contains parameters mentioned in the paragraph above. Calculation of the equation is performed by numerical integration using the Euler method with an intermediate step of calculation for greater accuracy. Since it is a numerical integration and the characteristics of tracks are different with the changing position of the train, it is necessary to calculate dynamics repeatedly according to a defined time frame.

Driving dynamics of trains is closer to the microscopic approach in the *MesoRail* tool. Accuracy and speed of the calculation can also be affected by the choice of the time frame in which the train driving dynamics is recalculated.

3.4. Railway traffic

The simulated train flows can be either deterministic or stochastic. The trains can then follow an estimated flow diagram of train traffic or they can be burdened by various delays.

Delays are specified according to appropriate methodologies of railway companies. Simulator's



Figure 2: Train Tachogram Example

decision-making algorithms enable, for example overtaking for priority trains or choosing another alternative without user interference in case of an occupied track. All train movements are carried out in compliance with deployed signal and interlocking systems in such a way, in which the simulated scenarios mirror real railway traffic.

3.5. Animation and simulation protocols

MesoRail features online animation for easy and clear overview of the progress of a simulation. The simulator is capable of setting the speed of animation based on the user's needs or to pause the animation if a detailed inspection of the simulation status is needed.

Considering the schematic depiction of the infrastructure, deformed depiction of trains can occur. Depending on an accurate length of a track segment in scale to the depicted length, the length of a train or the speed can change, even though both values are constant. However, the simulation correctly observes the location and the speed of a relevant train and the depiction deformations are an expected consequence of the schematic infrastructure layout. Example of this behaviour could be seen at Figure 1.

After completing the simulation, post simulation protocols can be displayed. From these protocols, data can be filtered and further processed into, for example, tachograms of trains (see Figure 2).

3.6. Simulation core

The simulation engine of *MesoRail* is based on the method of discrete events planning. This method uses a calendar of events, from which events are removed in the correct chronological order and executed. In classical simulation using discrete event method, any entity, that has been included in processing the previous event, can create an event and include it in the calendar.

There is no hierarchy or delegation of competences and despite its simplicity, in complex simulators there can occur a situation in which the simulation will be messy and every entity will plan a considerable amount of its own events.

In contrast, agent-oriented architectures, such as ABAsim, divide competences and responsibilities between agents and allows hierarchic classification of agents (Kavička, Klima and Adamko 2005, Adamko 2009). Agent-oriented architecture has become the inspiration for the design of the MesoRail simulator. Although it uses a simple method of event planning, the concept maintains simulator architecture of ABAsim along with the Java programming language and object oriented programming features. Classes represent different types of agents. Communication within each part of the agent is performed by the class attributes (also serves to maintain the agent's state) and by method calling. Communication among various agents is also carried out by public method calling. Agents can also add an event to the calendar of events and cause a shift in simulation time.

In the actual simulation, trains represent dynamic mobile agents that move along the track. Their movement is controlled by a central dispatcher (managing agent), which controls all traffic on the railway infrastructure.

The concept of a simulation based on the use of autonomous agents allows good division of responsibilities and competences of the individual agents and thus creates a transparent environment, in which it is easier to navigate, and provides an easy possibility to expand for future development simulator.

3.7. Verification and validation

In order for the simulator to be used in practice, it is necessary for simulation results to be correct and real. Since it is a mesoscopic simulator, the results will not reach the same level of accuracy as other simulator using a microscopic approach. Nevertheless, it is necessary to ensure correct functionality of individual parts of the simulator.

In principle, it is possible to identify several basic parts of the simulator which need to be verified, or rather validated:

- functionality of basic data structures,
- train driving dynamics,
- correctness of the decision-making algorithms governing the operation of trains (compliance with track and station intervals, obeying speed restrictions on the line, the correct function of safety devices on railways ...).

To verify the functionality and proper behaviour of data structures, it is useful to use unit testing using package *JUnit* in *Java*. The tests themselves, however, do not provide assurance of correct behaviour, tests must then be written by a developer or a tester, and it is necessary to include as many possible situations so that the behaviour of data structures is tested thoroughly. In essence, the quality of the written test to test critical points of implementation matters. Unit tests are used to test data structures such as graph implementation, basic functionality of methods for discrete event planning, and other parts in *MesoRail*.

Proper implementation of the train driving dynamics is a very important part of the simulator. To be able to validate the driving dynamics, it is necessary to first perform the simulation to obtain information about train movement in the form of a tachogram. Next, it is possible to compare it with a similar tachogram obtained from a real environment or other simulation tool. In case of the *MesoRail* simulator, validation against real data provided by *ŠZDC* (Diviš 2014) was used in the initial phase. Other, more complex experiments evaluating the effect of a gradient, curves on the track during acceleration or braking were compared with results from the simulator *Villon* (representing a validated simulation tool widely accepted by experts from railway practice).

Rail traffic control is also an essential element on which the credibility of the simulator depends. For simpler problems related to traffic management, it is possible to ask railway experts who examine the simulation results in detail and assess the accuracy of the simulation. In the case of a complex scenario, it is possible to use an alternative simulation tool, in which the same simulation will be created and then their progression can be compared. If stochastic events occur in the simulation, it is possible to compare the various statistical indicators. With ensuring a sufficient number of replications, the results should be comparable.

4. CASE STUDY

As part of the initial testing of the *MesoRail* simulator, an infrastructure, which consists of a prototype station and several adjacent sections of tracks, was used. The total length of tracks in the station and the adjacent track sections is almost 20 km and the track includes a series of arches and various gradients. This infrastructure has been implemented in the *Villon* and *MesoRail* tools. Next, a series of simulation experiments was conducted. Types of trains used were -(i) a passenger train, (ii) an express train, (iii) and a freight train.

After the experiments were performed, statistical evaluations of the results were done and the differences in driving characteristics were compared. Sections where trains accelerated and braked were compared separately. Table 1 summarizes the average deviations of the measured distances that the trains had to travel to reach the prescribed speed in comparison with the *Villon* tool.

Table 1: Average Differences in Measured DistanceDuring Deterministic Experiments

Туре	Percent difference	Absolute difference
Acceleration	2,01 ± 0,01 %	$26,33 \pm 14,08$ m
Braking	5,56 ± 0,03 %	33,44 ± 16,22 m

The difference of measured values between the *MesoRail* tool and the *Villon* tool is in the matter of units of percent and it is expected due to the mesoscopic simulation level. Braking, however, shows higher difference than the acceleration of trains. The reason for this difference will be the subject of further examination during the development of the simulator.

For investigating the simulation, support for time shift was implemented in the simulator. The core allows to automatically save the state space of the simulator, or it can store it under certain conditions or upon user request. During the course of the simulation and after its completion, it is possible to seamlessly move between states and to view the previous states, and thus find out what conditions led to the particular behaviour of the simulation.

Individual states of simulations are stored in memory. However, the complete status information (including all the variables at a given point in time) is not stored, but only differences from the last recorded state are created. This method saves memory but slows down the saving of state itself (it is necessary to calculate the difference), and in particular, it slows down the recovery of state to the state space of the simulator. Restoring a complete state from partial differences must proceed in an iterative way back into the past, until complete status information is not created.

Although storing states of the simulator leads to increased overhead of the simulator and also to an increase in requirements on the amount of RAM, this functionality enables detailed analysis of the simulation and identification of potential critical points, where a bad decision due to an error in the code or configuration of simulator occurred. Support for the time shift is being further developed and alternative storage options for states for comparing demand on memory, disk space and processor utilization during saving and restoration of states are tested.

5. CONCLUSION

The need for faster examination of railway station throughput has led to the design of a new mesoscopic simulation tool, *MesoRail*. The difference between a classic microscopic simulator and the *MesoRail* tool is that *MesoRail* uses several different abstractions that enable faster execution of simulation projects at the cost of lower accuracy.

In the article, the *MesoRail* tool was introduced, and at the end a case study with comparison of results against the microscopic simulator *Villon* was demonstrated. The achieved results correspond with the estimated accuracy of the simulation, but parts that need to undergo further examination and possible repairs were also discovered.

Now, the *MesoRail* tool is able to perform simulations of simple railway nodes. Currently, the development is focused on the implementation of advanced decision-making algorithms and support for complex node modelling.

REFERENCES

- Adamko N., 2009. Enhancing discrete simulation executive with simple continuous simulation and animation support. Proceedings of 23rd European Simulation and Modelling Conference, ESM 2009, p. 35-42. October 26-28, Leicester, United Kingdom.
- Diviš R., 2014. Train movement dynamics within AnyLogic tool. Proceedings of the European Modeling and Simulation Symposium, 307-312. September 10-12, Bordeaux, France.
- ETF, 2013. Capacity leaflet (UIC 406). Available from: http://www.uic.org/etf/codex/codexdetail.php?langue_fiche=E&codeFiche=406 [accessed 6 April 2015]
- Iwnicki, S., 2006. Handbook of railway vehicle dynamics. Boca Raton: CRC/Taylor.
- Kavička A., Klima V., Adamko N., 2005. Agentovo orientovaná simulácia dopravných uzlov (Agentbased simulation of transportation nodes, in Slovak language). Žilina:Žilinská univerzita.
- Marinov M., Viegas J., 2010. A mesoscopic simulation modelling methodology for analyzing and evaluating freight train operations in a rail network. Simulation Modelling Practice And Theory Vol. 19: p. 516-539
- OpenTrack Railway Technology Ltd., 2015. Railway Simulation. Available from: http://www.opentrack.ch/ [accessed 26 June 2015]
- RMCon, 2015. RailSys. Available from: http://www.rmcon.de/index.php?page=railsysclassic-planning [accessed 6 April 2015]
- Simcon, 2014. Villon. Available from: http://www.simcon.sk/en/tools/villon [accessed 6 April 2015]
- Stallybrass M., 2014. Mesoscopic simulation: a tool for analysing and optimising headways and capacity.

Proceedings of AusRAIL 2014, 15 p. 11 - 12 November 2014, Perth.

- SŽDC, 2009. D24 The regulation for determining capacities of railway lines. Available from: http://www.szdc.cz/dalsi-informace/dokumenty-apredpisy/provoznetechnicke.html?page=detail&docid=1%3B%237ea cf179-1551-468f-949a-641e9600c22d [accessed 6 April 2015]
- Xu Y., Song X., Weng Z., Tan G., 2014. An Entry Time-based Supply Framework (ETSF) for mesoscopic traffic simulations. Simulation Modelling Practice and Theory Vol. 47.: p. 182-195