Simulation of a railway mainline junction using High level Petri nets

Dušan Jeremić*, Milan Milosavljević**, Sanjin Milinković***, Slavko Vesković****, Zoran Bundalo*****

*Railway College of Vocational Studies, Belgrade, Serbia, jeremicd@gmail.com
**Railway College of Vocational Studies, Belgrade, Serbia, mimilan89@gmail.com
***Faculty of Transport and Traffic Engineering, Belgrade, Serbia, sanjin@sf.bg.ac.rs
****Faculty of Transport and Traffic Engineering, Belgrade, Serbia, veskos@sf.bg.ac.rs
*****Railway College of Vocational Studies, Belgrade, Serbia, cheminot2@gmail.com

Abstract — *Railway transport is one of the most complex kind of transport. Solving railway infrastructure problems and conflict situations is very difficult task. This paper presents a simulation model of mainline railway junction using High level Petri nets. The aim of the paper is to detect the conflict situations in station and analyses impact of infrastructure elements malfunction on train delays. Model was tested on mainline station Mjölby in southeastern Sweden.*

I. Introduction

Modelling and simulating railway infrastructure is often challenging and demanding task which mostly depends on desired accuracy of the model. For microscopic models which are most precise and accurate large number of elements and connections need to be included. There are number of tools and software packages developed for this purpose, with different levels of complexity and accuracy. Petri nets as a modelling tool are mostly used in manufacturing processes and communication technology. Although it's not their primary usage, railway application of Petri nets could be found in literature in past 20 years. Van der Aalst and Odijk (1995) proposed the interval timed coloured Petri Nets (ITCPN) for modelling and analysis of railway stations, where train delay is specified by an upper and lower bound, i.e., an interval. Fanta, Giua and Seatzu (2006) used a coloured Petri Nets (CPN) model of a dynamic rail system for determining deadlock situations. Daamen, Goverde, and Hansen (2009) developed a CPN tool for route conflict identification and estimation of knock-on delay [1].

Main goal of this research was creation of a model of mainline railway junction using High level Petri nets. For this task station Mjölby in southeastern Sweden was chosen. Reasons for this decision lay in complex infrastructure layout of station Mjölby as in its busy timetable. Model was created using Exspect, a software tool for modelling discrete processes. Modelling was performed by creating basic modules like module for block section and switch, entering them in order as in real system and then connecting them to each other. Number of other elements is then added to establish dispatching logic and provide functionality of model.

II. Petri nets

A. *Petri net theory*

Petri net theory is based on mathematical graph theory, more precisely to the two-part (bipartite) graphs. Bipartite graph is a graph whose nodes can be divided into two different sets V1 and V2 so that each branch connects a node in V1 and node in V2 therefore there are no two branches in the same set. Extension of the theory of graphs has become possible with computer development and from graph theory Petri nets were created.

Petri nets are mathematical modelling tool that is used in the analysis and simulation of concurrent systems. The system is modelled as a two-part directed graph with two sets of nodes: a set of places that are representative of the state or system objects and the set of events or transitions that determine dynamics of the system. The distributed system is modelled as a bipartite directed graph with two sets of nodes: the set of places that represent state or system objects and the set of events or transitions that determine the dynamics of the system [1].

B. *High level Petri nets*

The term High level Petri nets is used for many Petri net formalisms that extend the basic Petri net formalism; this includes colored Petri nets, hierarchical Petri nets and others.

Colored Petri net (CPN) has its each token attached with a color, indicating the identity of the token. Moreover, each place and each transition has attached a set of colors. A transition can fire with respect to each of its colors. By firing a transition, tokens are removed from the input places and added to the output places in the same way as that in original Petri nets, except that a functional dependency is specified between the color of the transition firing and the colors of the involved tokens. The color attached to a token may be changed by a transition firing and it often represents a complex data-value [2].

III. Modelling railway infrastructure using Petri nets

A. *Railway infrastructure simulations*

Simulating a railway infrastructure is a complex and demanding task since there are many variables and dependencies which must be taken into account to provide accurate model. The infrastructure can be modeled on a macroscopic or microscopic level depending on the effects a specific study aims to capture. A microscopic model consists of a node-link system and can contain all necessary characteristics and parameters of the real infrastructure, such as switches, signals, and speed and gradient profiles. The track layout can be defined with high accuracy. Macroscopic models represent data much more aggregated, stations can for example be defined by single nodes with attributes regarding the handling capacity. Links contain information on number of tracks, average speeds and other relevant information [3].

There are number of software packages developed for the simulation of railway infrastructure, most notable OpenTrack and RailSys but also Dons, Peter, Fasta, Railcap and many others [4]. On the other hand, there are number of software packages which were not designed primarily for railway simulations, but nevertheless can be used effectively. Best example is Matlab with its component Simulink.

B. Simulations using Petri nets

Petri nets as a simulation tool are mainly used in manufacturing and communication systems. They are suitable for use in concurrent systems but they are applicable to other uses like biological and chemical processes. In general, all systems which can be shown as set of places and have clear transition rules can be modeled with Petri nets.

In Petri nets trains can be represented as tokens while all other infrastructure elements (block sections, switches, station tracks…) can be represented as places. Since tokens are colored, they can carry number of information about train attributes like train number, length, train type which are constant trough model but also some information which can be changed as train (token) passes through model. Those information include arrival, departure and waiting times at each place, train path and other relevant information concerning train movement through system. Information which include time be carried only on timed Petri nets.

Transitions in Petri nets are responsible for applying rules for token movements through model or in other words for token firing from input to output places. There are many rules with various complexity which can be used in transitions. Simpler rules usually do not include token color or other properties as a decision criterion, for example, rule that forbids transition (firing) if output place is busy. Simpler rules are usually embedded in simulation software. More advanced rules use token color as a decision input, for example, train type or train length are used to determine if transition would occur or not. Or in case of multiple output places rules which include train properties could be used to determine to which output place token is going to be transferred (fired).

IV. MODELLING ELEMENTS OF RAILWAY INFRASTRUCTURE

A. Moddeling software

For this research model of railway infrastructure was created using software ExSpect [5]. ExSpect is a software tool for discrete process modelling. It is suitable for business process modelling, production chain modelling, and use case modelling, etc. ExSpect embodies a colored Petri nets approach to process modelling. An ExSpect model describes a process in terms of a collection of subtasks that communicate by message passing. The network of subtasks is displayed and edited graphically. A subtask can itself have a process model; this allows large models to be decomposed into manageable parts. Models can be analyzed for structural correctness properties, and their behavior can be observed through simulation, either step-by-step, continuous or with breakpoints.

Components of the program include predefined blocks which can be used to create objects. Basic blocks are places (channels), processors (transitions) and connections (arcs) which are basic Petri net elements but also some other blocks. Those include store, which are used to store and export data during simulations, time generator and input and output pins, which are used for creation of modules inside a model.

ExSpect as a software package is easy to learn, uses intuitive programming language, allows creation of complex models through creation of smaller modules and supports exporting data to number of formats. Unfortunately, its development stopped in 2000 due to number of bugs and compatibility issues with newer versions of Windows.

B. Modelling elements of railway infrastructure

There are two ways how system can be modeled in ExSpect. The first way is to create a model by entering the places and transitions for each element of the model, and then link them together and define the transition conditions for crossing (firing). This method requires that each element or object it the system is specifically defined and connected. For small and simple systems this method is efficient, accurate and reliable as the programmer can easily analyze all the connections and put all data into the model.

The second method involves the creation of subsystems or modules. These modules are stored in the program database and can be accessed and easily incorporated into the model. This feature is very useful for creating large models where some elements of the model appear more than once. For simulation of railway lines, modules correspond to isolated sections of track. The model is created by entering modules in order as in the real system, connect them to one another and with stores in the model. Such a process of modeling takes more time during the initial programming, but enables that these created modules could be used to model any system where there are similar processes, and by creating additional modules it is possible to model large number of railway systems.

C. Modules in Petri net simulations

Basic modules which are used in modeling any railway line are block section and switch. Block sections can be either line block sections or track sections in stations. Basic principle is the same for both kinds of sections, each module has two input and two output pins for both directions. They are used as an external connection to the rest of the model. Aside from those, there are three more external connections for stores time, info and stanje. Time is connected to main time store which provides time reference, stanje is connection to a store which provides information about occupancy of the section and info is connection to a store which provides information about section length and maximum speed for all train types. When token enters the module, data is written in store stanje so section is marked as occupied. Occupancy time is calculated by dividing section length with section speed, which is provided from external store info. That way module is universal and can be used for any bidirectional block section. For block with only one possible direction module is simpler and includes only one subsystem. Module for bidirectional section is shown in Figure 1.

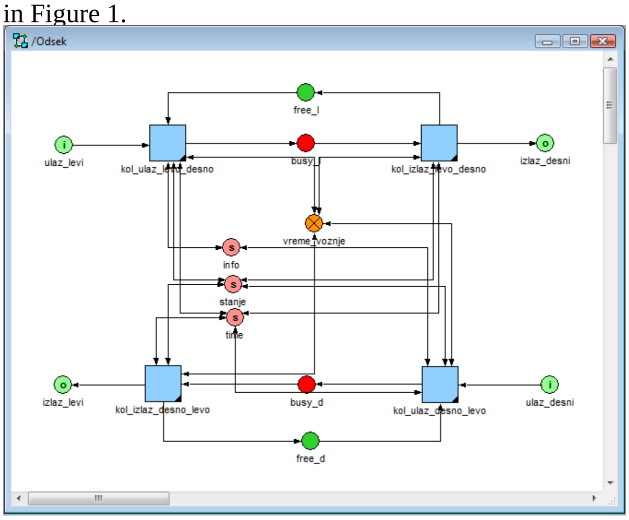

Figure 1. Bidirectional block section module

Module for switch is a bit different since it has a one entrance (input) and two exits (output). It has two branches, one for normal position of the switch and other for reverse. As in block section it has three stores, stanje, info and time which have same function as in block section, but also one more store put. Purpose of this additional store is to set the position of a switch to normal or reverse. There are two types of modules, one for diverging and one for converging switch. Diverging switch when observed from left to right has one input and two outputs while converging switch has two inputs and one output while they are similar in other aspects. (Figure 2)

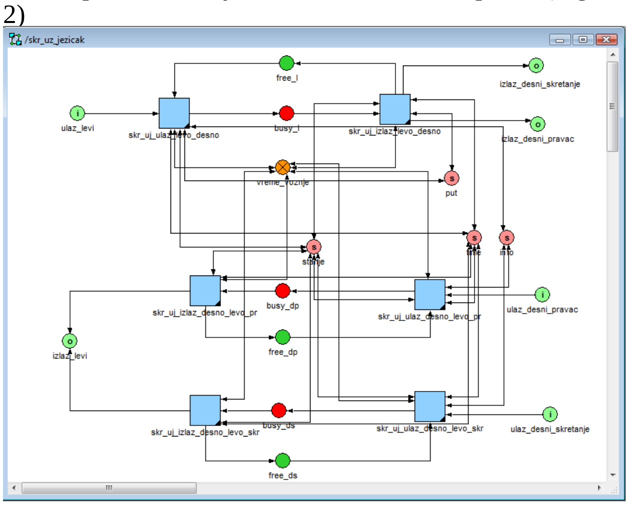

Figure 2. Module for diverging switch

These modules represent basic building blocks for railway infrastructure model. They can be further modified to fulfil any specific need.

D. *Application of Petri net models*

Petri nets can be used to model number of infrastructure elements. Simplest task is to model open line or block sections. Model is created just by adding section modules and connecting them with one another. More advanced application is modelling of a railway junction and adjoining block sections [1] [6]. Besides using modules for block sections and switches it requires much more elements for functions such as route setting, route releasing, etc. Most advanced task is modeling a railway station. Depending of the size of the station number of elements can be significantly different. Main factor in model size and complexity is number of possible routes in station. Smaller stations in that aspect are not much different from junctions since number of possible routes is generally small. Larger stations provide much more complexity, especially stations with many connecting lines.

V. SIMULATION OF STATION MJÖLBY

As an example of Petri net application for complex railway junction, Petri net model of station Mjölby was created.

A. *Location and importance of station Mjölby*

Station Mjölby is located in southeastern Sweden on Southern main line (Södra stambanan) which connects Stockholm with Malmö. It is one of most important railway lines in Sweden, connecting central and northern Sweden to southern Sweden, Denmark and rest of the Europe. It is a double track line, congested with many types of trains such as fast X2000, local commuter trains such as Östgötapendeln and many freight trains. Station Mjölby is also a junction from which line to Hallsberg, largest marshaling yard in Sweden diverges. On top of it, station Mjölby is terminal station for many local passenger trains from Linköping and Hallsberg. This adds to more than 600 scheduled trains daily from which around 300 are in service regularly. Track diagram of station Mjölby as seen on display in dispatching center Norrköping is shown in Figure 3 [7].

Figure 3: Track diagram of station Mjölby

Station was mostly reconstructed as a part of a double track project from Mjölby to Hallsberg in preparation of large increase in freight traffic from that line. Project was very soon stopped, freight traffic did not increase and as a result Mjölby is a station with lots of capacity and possibilities for parallel movements.

B. *Petri net model of station Mjölby*

Modules which were described in previous section and in [6] were used as a basis for constructing a model of station Mjölby. Some modifications were performed, mainly adding characteristic of types of trains (X2000, pendel_tranås, pendel_mjölby and godståg). Modules are placed and connected to each other in the order in which tracks are located in station. They are also connected to stores time, info, stanje and put for switches. All stores except time need predefined initial value. For store stanje initial condition is 0 (free) and for store put it depends of switch position, it is usually 0 (normal position). Definition of store info a bit more complicated. In each store type info is necessary to define the name of section, length in meters and speed for all four types of trains (X2000, pendel_tranås, pendel_mjölby and godståg) in meters per second. After connecting basic elements it is necessary to define dispatching logic.

Since Mjölby is a terminating station for number of local passenger trains (pendel_mjölby), special section module was created for tracks on which these trains terminate. For all other train types this section acts as a normal block section but for terminating trains, instead of passing them trough, it returns those trains (tokens) to the point of entry.

Dispatching logic in station Mjölby is not limited to one or two processors, but is incorporated in almost every main signal (signal processor). This allows on the one hand simplification of the code and on the other hand much more flexibility in creating a model. There's the fact that there are a number of different possible train routes in Mjölby and often only entrance routes are used for passenger trains, therefore was necessary to disperse dispatching logic on multiple processors. This is done so that all entrance and most intermediate signals are presented with a separate processor [7].

There are number of other elements which are necessary for proper functioning of this model. Those include section clearance (release) processors and some additional processors. Due to the large number of crossovers, short-track parts and complex track diagram on the one hand, and longer trains than some main tracks on the other hand, section release logic need to be solved differently. It is defined in modules that that train (token) entrance is registered in the section store as a condition 2 (occupancy), but the clearance (release) of sections, due to factors mentioned above is not performed in the processor inside the module, but in special release processors outside the module. They are variously arranged in the model and carry out route release function of various numbers of module sections. In general, section release processors are arranged so that they can approximate the length of the train factor in the best way. Therefore an approximation needs to be performed where some points are grouped so sections are cleared when train (token) enters last section (point), some short sections are not cleared immediately upon exiting the train from that section but only when train leaves next section, etc.

Additional processors mostly include platform processors and train number changing processors. Platforms processors are used to simulate stopping time of certain passenger trains in the station, and only on the tracks that have a platform next to it. Number changing processors are used next to terminating track modules and they change number of terminating trains before their return towards destination station. There are also few additional stores like sig which is used to store information relating to the signal aspect and prolaz which is used to transmit information to intermediate signals if the route is already formed over that signal by another (usually the entrance) signal or it needs to be set when train enters a section in front of the intermediate signal.

Upon completion of all definitions, it is necessary to integrate all elements. In addition to linking all the elements, order of connection must be considered, for example signal processor with a section stores or processor for release with the section stores. The result is a completed model of station Mjölby (Figure 4).

Figure 4. Completed model of station Mjölby

After developing model, to launch the simulation it is necessary to define the initial value, i.e. trains. The program allows input train data from external sources (txt file) which must be written in necessary form with a specific sequence of data. Trains must be sorted by time of entry into the system and other information must be sorted in alphabetical order.

VI. SIMULATION RESULTS AND DISCUSSION

A. *Simulation parameteres*

Simulation of station Mjölby was performed with two sets of input data. In first set train arrival was according to schedule and in other set train arrival was randomly generated according to proper distribution for different train types. For random arrivals ten combinations were generated and accordingly ten simulations were performed. For scheduled arrivals ten simulation were also performed but with same results since only one set of input data was used.

Since ExSpect has the ability to export data into an Excel file, in this model it is defined that the data is entered into an Excel file from each channel between the sections (or groups of sections in case of switches) and from each section store stanje. In this case Excel was chosen between other formats due to suitable processing and well known file format. Information from each channel is written into one xls file (izlaz.xls), which serves mostly to review the functioning of the whole system, of delays, the most used routes, while data from each store is entered into separate file which can be later merged into single file for comparison and better view of section occupancy time. Same principle applies to stores put which export data about switch positions. Every change in store or place is recorded along with time of change. For every place which token (train) passes, all train properties (token color) are exported to xls file. Every entry includes data about train that usually do not change like train length, train type, destination and data which changes at each place that train passes like current train path and current position. All entries include simulation time at which they were recorded.

B. Simulation results

Goal of this simulation was to determine critical places or points of possible conflict in station but also impact of track element malfunctions on train delay. Same methodology was used both for scheduled and random train arrivals. Both tasks were performed using data from export xls file.

There are several criteria used to determine points of possible conflict. First was to compare frequently used routes and determine which are conflicting. Second, to determine which switches were most frequently used as part of different routes. Next was to determine occupancy time of those switches, both with train and train route. Another criterion was to determine number of train routes per switch. And as a last criterion, minimal time between consecutive passes of different trains over switch was measured.

Data about frequently used routes was gathered from main xls file (izlaz.xls) which recorded all data about trains through system. All data is written in one sheet chronologically using simulation time. Since every route has predefined switches which are included in that route, simply by comparing conflicting routes it is possible to determine which switches are most used by conflicting routes. Therefore, switches which were most used by conflicting routes were likely to be places of possible conflict. (Figure 5)

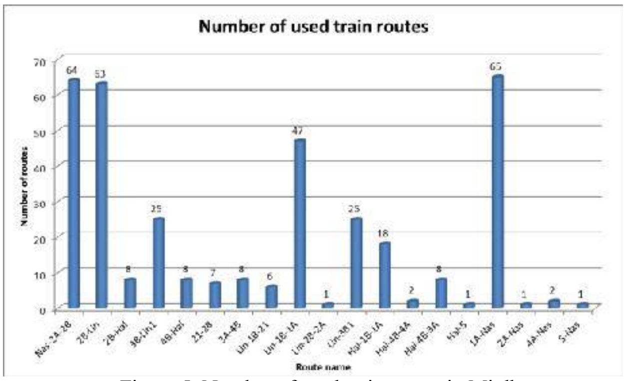

Figure 5. Number of used train routes in Mjölby

Switch occupation time is determined from each switch store. There are three possible conditions of a switch, and each condition associated with numeric value. Those conditions are free (value 0), occupied by route (value 1) and occupied by train (value 2). All values are written in separate xls file for each switch, together with time of condition change in simulation. Overall occupancy time is obtained by adding time difference between each change of condition. Switches with longest occupancy time are likely candidates for place of possible conflict. Another way to determine which switch could be place of possible conflict is to compare number of train routes per switch or in other words number of trains that cross each switch. That information is obtained from izlaz.xls simply by counting how many times each switch appears in file (Figure 6).

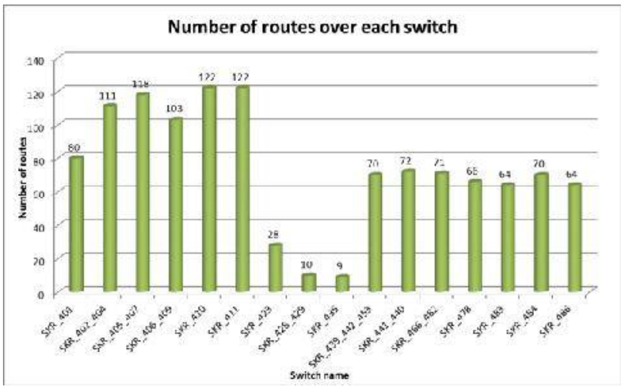

Figure 6. Number of train routes per switch

Minimal time between consecutive passes of different trains over switches was performed only on switches which were by previous methods determined to be likely places of possible conflict. It was also measured from izlaz.xls by comparing time difference between each passing over certain switches. All time differences were grouped into time intervals. For each switch and each interval, number of occurrences was shown. (Figure 7).

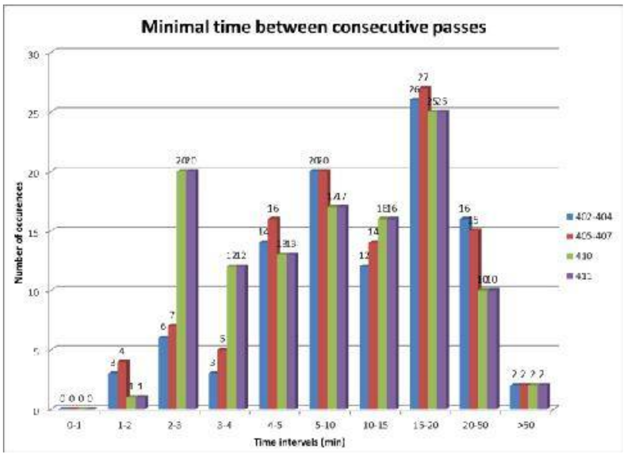

Figure 7. Number of minimal times between consecutive passes

Same methodology was used in case of random train arrivals. With ten different sets of input data, ten different simulation results were obtained. Difference between them was not significant, all results were with approximately 10% variation. For example, number of routes over switches for first five simulations with random train arrivals is shown in Figure 8.

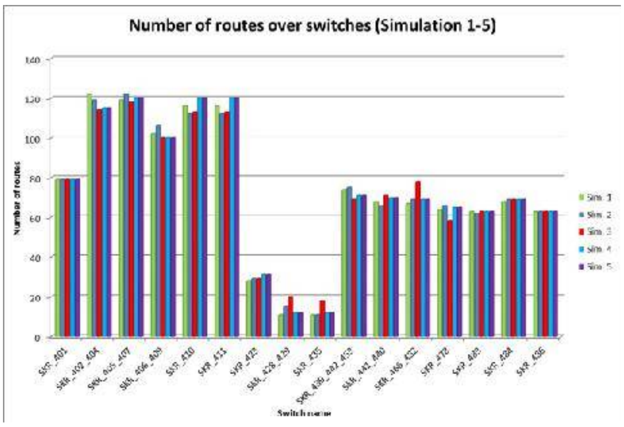

Figure 8. Number of routes over switches in case of random train arrivals

As it can be seen results are only marginally different then in case of scheduled train arrivals. Similar results were obtained for frequently used routes, switch occupation time and minimal time between consecutive passes of different trains over switches.

Second goal was determining impact of track element malfunctions on train delay. Track elements included both track sections and switches. Two kinds of malfunctions were considered, one was malfunction of isolated section which can occur both in blocks and switches, and other was malfunction of switch mechanism. Reasons and probabilities of isolated section malfunctions were not considered since that requires much broader analysis of interlocking device in station Mjölby and its reliability. Analysis of possibility of switch mechanism malfunction was performed by analyzing their usage, or number of throws per switch. Number of throws per switch is obtained from each switch store put, which holds information about point position. As switch can be in two positions, normal and reverse, store put can also be in two conditions, normal (value 0) and reverse (value 1). Number of throws is determined by counting number of changes from 0 to 1 and vice versa. Total number of throws is shown in Figure 9.

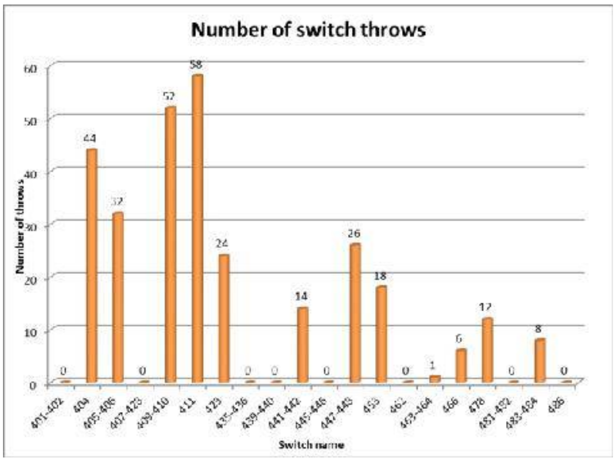

Figure 9. Number of throws per switch

Impact of track element malfunction on train delay was tested by changing the state of observed element from free into busy at the start of simulation. Tracks and switches whose malfunction would block entire station were not analyzed. Analysis was performed by comparison of delays in case of malfunction and delays in normal operating conditions. Results showed small delays, around 15-20 seconds when malfunction was on through main tracks and up to one hour when malfunction was on one of the terminating tracks, namely 3B. Those high delays were only observed on one type of train, local terminating trains from Hallsberg in case they could not use their designated terminating track 3B.

VII. Conclusion

High level petri nets proved to be powerful tool for creating models of railway infrastructure where greatest advantage is their flexibility. Most important advantage is possibility of creating modules for any part of railway infrastructure. Mostly used are modules of repeating or frequently used elements like track section or switch, but even complete station could be used as a module. Since modules are not predefined but created by user, they can also be modified for specific purposes. Block section for terminating trains is a good example, where standard block section module is modified to return certain type of train i.e. terminating train to a point of entry instead of passing them trough.

Another flexibility is dispatching logic, which could be defined as centralized in one processor or as in this paper decentralized to several processors. Route setting rules can be defined in number of ways, using practically any train attribute as a route setting criterion. Train attributes are also custom category defined by user and they can be defined in number of formats (integer, real, string…). Number of other processors and transition rules can be easily defined and incorporated into model.

As for station Mjölby simulation showed that it is well projected with lots of possible routes and possibilities for parallel movements. Number of places of possible conflict is low, both with scheduled and random train arrivals. Impact of track circuits and switches malfunctions produces only small delays. In general, it is hard to find any part of the station which could be source of conflict situations. Reasons may lay in fact that data for this research is from 2008, and in that period there were not so many train using line to and from Hallsberg. Therefore, functions of this station as a junction were not so pronounced and it mostly served as a terminating mainline station. Today is a bit different situation, there are much more passenger train which use line to Hallsberg. And for a future research it would be interesting to test a model with current train schedule and at the same time determine if Mjölby is really well projected as this research showed.

Acknowledgment

This paper is supported by The Ministry of Educations and Science of the Republic of Serbia, within the research projects No. 36012.

References

[1] S. Milinković, M. Marković, S. Vesković, M. Ivić, N. Pavlović, A fuzzy Petri net model to estimate train delays, Simulation Modelling Practice and Theory 33, 143-157

[2] Petri Nets for Dynamic Event-Driven System Modeling, Jiacun Wang

[3] Simulation of rail traffic - Methods for timetable construction, delay modeling and infrastructure evaluation, Hans Sipilä, Ph. D. thesis, Stockholm 2015

[4] Barber, F., Abril, M., Salido, M. A., Ingolotti, L., Tormos, P., Lova, A. 2007. Survey of automated Systems for Railway Management. *Technical Report DSIC-II/01/07.*Technical University of Valencia.

[5] http://www.exspect.com/

[6] Milinković, S., Vesković, S., Marković, M., Ivić, M., Pavlović, N. (2010). Simulation model of a railway junction based on Petri Nets and fuzzy logic. *Selected Proceedings of the 12th World Conference on Transport Research Society*, 11-15 July Lisbon, Portugal.

[7] Jeremić, D. Utvrđivanje konfliktnih tačaka i parametara rada stanice Mjölby simulacionim modelom Petri mreža, Diplomski rad, Saobraćajni fakultet, Beograd 2010