A Petri nets based decision support tool for railway traffic conflicts forecasting and resolution

S. Ricci & A. Tieri

Department of Hydraulics, Transport and Roads (DITS),
Sapienza - University of Rome, Italy

Abstract

The full paper describes current results and applications of an ongoing research concerning the operational simulation of complex railway systems based on working criteria and operational rules of interlocking and line signalling systems. Modelling is based on object oriented approach by the Petri nets formalism, that allows high processing openness and modular building without breaking away from the safety systems logic. The development of a full model includes the integration of a Common module to be applied to any layout and interlocking technology, a Timetable module capable to simulate the operational program, a Movement Inspector module to manage traffic priorities automatically and finally a Top Level module where a specific layout and its routes are modelled. The software tool used is a specific software for extended Petri Nets development, validation and simulation (Faber©), that allows modularity in Petri nets definition and manages variables and types declaration, being endowed with a C language interpreter. The tool could be used both for regularity forecasting and in decision support systems for the management of conflict resolution. For these goals the model was largely tested on the Roma-Formia railway, an Italian high density traffic line. The tests include simulations of traffic regularity monitoring and forecasting in both un-perturbed and perturbed conditions, by generating delays of sets of trains along the line and by verifying the response of the system in comparison with measured delays.

Keywords: Petri nets simulation, railway traffic management, automatic overtaking.

1　Introduction

The possibility to improve the capacity of railway lines, honouring the regularity levels fixed by the railway undertakings, may be verified or at least estimated with good approximation, during the planning phase through simulation models of railway operation capable to evaluate the potential criticalities, that obstruct the achievement of the above mentioned transport service levels.

The models should usefully simulate the process with all boundary conditions imposed by infrastructure, signalling and control systems.

Moreover they should be applicable for as large as possible sets of different railway systems, i.e. they should have high flexibility in the system representation.

The research aims at building a model with described characteristics, joining in already achieved simulation modules of the operation of single elements of railway operation control systems (switches, track circuits, signals, etc.), through the reconstruction of their logical connections, using Petri nets graphic language.

So far, with the software Faber$^©$ capable of drawing and simulating hierarchical, extended and coloured Petri nets, were built a complex station model based on ACC and relay interlocking systems and various line models.

During the research, the same formalism was used to design a model for traffic regularity analysis, switching from a micro to a macro approach applied to railway operation.

Final goal of this research is a simulation model structuring based on Petri nets for line capacity and railway operation regularity, capable to support the effectiveness and efficiency evaluation of infrastructural and/or operational improvements to the railway system.

2　Architecture of the railway model for regularity analysis

The model was built using Petri nets to represent elements of operation signalling and control systems according to an object oriented approach.

The railway line (Roma-Formia double track high traffic line was used for model calibration), is divided in finite elements that, conveniently designed and assembled, can be repeated and composed to describe any railway system with similar signalling and control equipment [1].

Like shown in Figure 1, the model includes three modules (Common module, Movement Inspector and Timetable, inserted in a fourth module, the Top Level, in which is represented the infrastructure layout, with the objects of Common module and the route functions that trains will follow during their running over the railway line.

2.1　Common module

All physical elements of the signalling and control systems are designed into the common module.

The elements represented in the regularity model are: track circuit, switch, advanced signal, block signal.
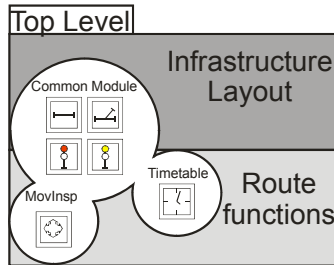
Figure 1:   Model structure.

These classes of objects are drawn through many subnets, each with a particular function.

They are developed so that can have all possible options along the line; in this way it is possible to use the same class also for different situations, without re-drawing objects for particular conditions.

Therefore, for instance, the unique class Track circuit can be used indifferently in both traffic directions, last or intermediate track circuit of a route, in the operations of freedom check, route interlocking and running.

Practically, the classes of objects are as much as possible "universal" and this fact allows the railway line representation simply "cloning" and linking with dedicated portsets the objects.

2.2 Timetable module

In this module the subnets that manage the train departures according to desired operational timetable are drawn.

It is possible to modify the timetable simply changing some parameters in a text file into the simulation environment of Faber$^©$.

This opportunity are very useful in the study of perturbed timetable, when, for instance, it is possible to impose on a train a fixed delay and to verify what happens to the global traffic on simulation.

2.3 Movement inspector module

The model is equipped with an automatic overtaking function for different train types.

The Movement Inspector module plays the role of manager of this overtaking system, comparing in prearranged line sections the train arriving before with those arriving after.

Obeying to a structured priority order, this module commands to the first train a route change with stop in the first available station if the second train has a planned overtaking, but it cannot reach its operational speed due to the line occupation by the first train.

The model the can also work in manual mode, i.e. not activating the Movement Inspector Module.

Practically the model is thought like a decision support tool and for this reason the user can freely choose to manage the overtaking system both manually, like often happens in a real Movement Inspector, and automatically with a train category system, so that it is possible to compare real and simulated delays.

2.4 Top level

The Top Level is the place where all the objects designed in the previous modules are stored.

In this class there are two types of functions:

- the first is to represent the physical layout of the railway line, with all the elements of the net;
- the second is to draw the different route functions existing along the line and in the stations.

2.5 Used token types

In order to allow the information movement across the model subnets, an important property of Extended Petri nets is used, i.e. a token can be defined like STRUCT variable becoming a real data record.

This because in Faber$^{©}$ is possible to insert codes in C or C++ in each transition.

This means the possibility to constrain the transition firing to come true of a programmed condition, one between usual types IF-THEN-ELSE, WHILE-DO, CASE-DO or, on the contrary, to perform a particular action, for instance to modify values of STRUCT variable represented by a token located in an input place of a transition when the same transition fires.

In the model three types of tokens are represented: TRAIN, MA and NUL.

The TRAIN token is used in the subnets of track circuits and switches that simulate the train running; it is a STRUCT variable defined by following simple variables:

- ID_TRAIN, the ID code of train;
- CAT, the category of train (high speed, national, regional, freight);
- TIME, absolute time of running;
- DIR, direction of traffic;
- L, length of the block section where is the train;
- PROG, progressive kilometre of the line where is the train;
- DIST, distance from the next station;
- SPEED, current speed of the train;
- PR, priority order, binary variable that communicates to the train to give way to the train after, changing the first possible route and waiting for the next train with the right of way;
- ITINxL, route in x station, left side;
- ITINxR, route in x station, right side.

Obviously, the number of simple variables needed to define the TRAIN token increases according to the number of stations increase.

The MA token, i.e. the Movement Authority, enters in all subnets representing logical functionalities applied on train different from running.

Therefore it is in the subnets representing freedom check of a route, route interlocking, release of route-locking, communication with signals, etc.

Also MA is a STRUCT variable, defined by the following simple variables:

- ID_TRAIN, the ID code of train;
- DIR, direction of traffic;
- ITINxL, route in x station, left side;
- ITINxR, route in x station, right side.

Therefore, if the number of station increases also increase the simple variables to insert in MA.

The last type of token existing in the model is the simple variable NUL, used only to activate some transitions with logical conditions independent from train actions, but necessary for model functioning.

2.6 Model output

The simulation environment has a visual simulator and an output text file.

In the simulator it is possible to select every page or subnet and to check the tokens movements and the system evolving through the colours that are set up in phase of net building.

In the text file there are the information desired by programmer and in particular the time of simulation, the ID object code, the action and the token.

In Figure 2 it is possible to see a simulation window: a station (Torricola) during the running (fuchsia objects) of a train on a locked route (green objects).

Figure 2:   Simulation environment and output file.

At the right of the same figure there is the output text file: it reports the actions, in particular freedom check, interlocking, track occupation, running (column 3), of the train number 2381 (column 4) over each object (column 2), and the time of the action (column 1).

3   Model application to railway traffic conflict resolution

In comparison with the past models [1, 2] the subnets of different classes of objects have been modified to be able to interact with the new module Movement Inspector.

To show better the potentiality of this new module as decision support tool for the increase of traffic regularity, some examples of possible applications, with or without the manager of automatic overtaking will be shortly described.

Examples and figures concern the first model application on the Roma-Formia, Italian railway line with 10 stations, many hundreds of routes, almost 700 elements between track circuits and switches and with 63 trains running in the morning peak period timetable from 7:00 to 10:00.

The line was perfect for model calibration both for high criticality in terms of traffic regularity and because it has some morphological complexities (e.g. two branch single track lines towards Terracina and Nettuno and tracks run in both directions).

Moreover, the model was tested on Roma Ostiense, large and complex station with 15 running tracks and a electronic interlocking system (ACC).

3.1 Standard mode

In case of standard mode functioning, the possible traffic and stabling conflicts are solved through a simple 'first in first out'.

This mode may be useful in phase of timetable planning on new line or timetable upgrading on existing line with new traffic demand or to verify circulation criticalities. In fact it is possible to simulate the theoretical operation timetable forcing the trains entrance into the railway line and starting the simulation. Then it is possible the comparison between the output file and the theoretical timetable and to check/correct conflicts or border line situations.

The flexibility of the output file makes available any information about logical states of railway elements and trains and allows the data transferring in graphical output and synthetic index usual for railway world (e.g. the representation of the graphical timetable on whole line, with possible warning of conflicts).

In figure 3 a portion of a simulated timetable from Roma to Formia is shown.

When all the conflicts are detected, ion this mode it is possible to solve them "manually", i.e. to delay the retarding trains and to run the simulation, with an iterative process that stops itself when ever the timetable perturbation is considered acceptable. This operation is very complex and strongly dependent on "sensitivity" of the movement authority. To decrease the stochastic effects due to human factor, a system of automatic overtaking with criteria defined into the Movement Inspector class of object has been conceived.

3.2 Traffic conflict resolution mode

The problem of traffic conflicts resolution in a generic line can be approached with a fixed goal (e.g. the minimization of mean delay) for the elimination of conflicts in each station of the line. This operation requires the preventive definition of the problem constraints, (e.g. a fixed priority order between different classes of train).

The model is very useful because it allows to change constraints and goals, so far an optimized solution is found as a result of many simulation run.
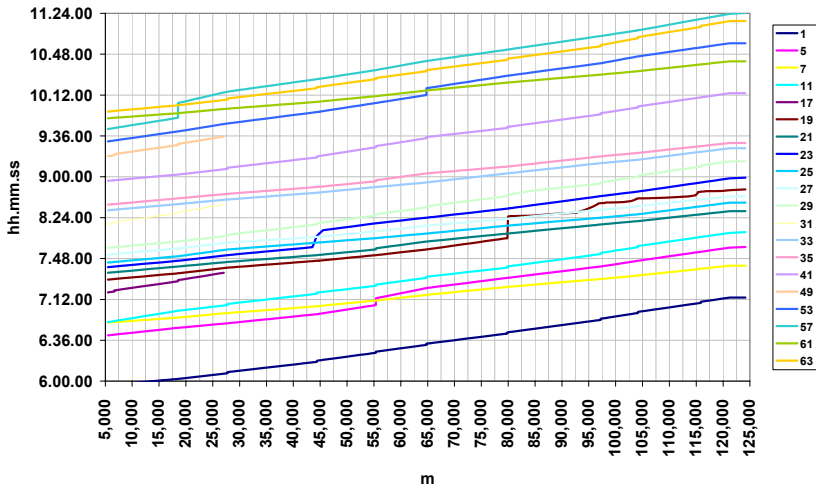
Figure 3:   Part of graphic timetable from simulation results.

Should our goal be the minimization of the delay generated by the conflict between two trains entering the same platform of a station, the model can execute the algorithm in figure 4.

This shows the following possible solutions:

- solution 1: the trains do not change their routes, the train B does not delay, the train A is delayed as much as no other trains are delayed;
- solution 2: the trains do not change their routes, the train B does not delay, the train A has the minimum delay and consequently other trains are delayed;
- solution 3: the trains do not change their routes, the train A does not delay, the train B has the minimum delay and consequently other trains are delayed;
- solution 4: the trains do not change their routes, the train A does not delay, the train B is delayed as soon as no other trains are delayed;
- solution 5: the train A changes his platform, the train B does not change his route, the train A is delayed as much as no other trains are delayed;
- solution 6: the train A changes his platform, the train B does not change his route, the train A has the minimum delay and consequently other trains are delayed;
- solution 7: the train B changes his platform, the train A does not change his route, the train B has the minimum delay and consequently other trains are delayed;
- solution 8: the train B changes his platform, the train A does not change his route, the train B delays as much as no other trains are delayed.

3.2.1 Example of automatic overtaking subnet
When in the model is active the class Movement Inspector, it manages the automatic system of overtaking.
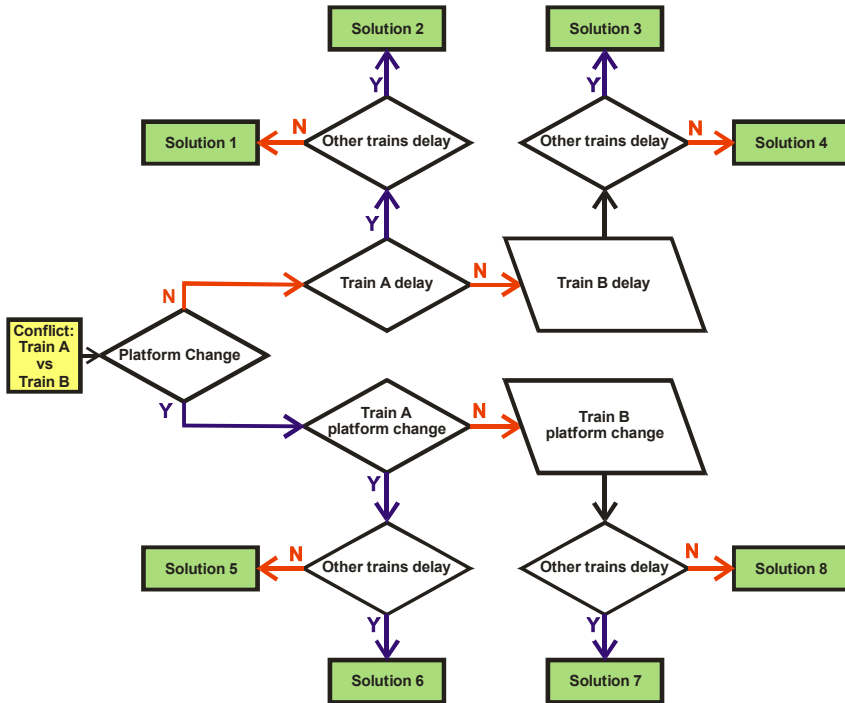
Figure 4: Decision making tree for conflict resolution between two trains.

In this module each train is compared with the train running before and after in order to verify if and when an overtaking is necessary.

The track circuits corresponding to platforms of stations are linked to Movement Inspector, so that as soon as the TRAIN tokens exit from this track circuit and enter in the next one, they are intercepted and move in a subnet of Movement Inspector, that sorts the different TRAIN tokens in other subnets depending on direction an kilometric progressive of the line.
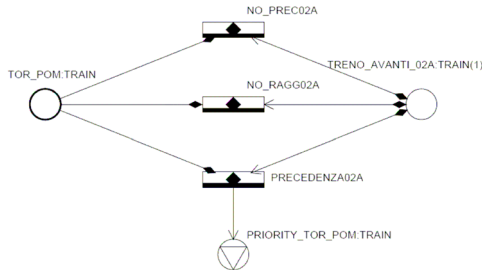

Figure 5: Part of subnet 'Overtaking TOR_POM' of class Movement Inspector.

These subnets, similar to that shown in figure 7, perform the comparison between trains.

In particular the net in figure 7 manages all the train running between two stations (Torricola and Pomezia) in a direction (towards Formia).

The TRAIN token intercepted in line arrives in the common place TOR_POM.

In the TRENO_AVANTI02A place is the TRAIN token corresponding to the train ahead.

At this point the three transitions make the comparison according to the following criteria:

   a.   the transition NO_PREC02A fires if the train behind has not right of way; this event happens if the train ahead is of higher category, if it run with a higher average speed, or if it is going to run in the next station a secondary route, different from the main line;
   b.   the transition NO_RAGG02A fires if, also with right of way, the train behind does not reach the train ahead before the next station;
   c.   the transition PRECEDENZA02A fires if the train ahead has a right of way and is fast enough to reach it before the next station.

The reaching condition of the train ahead by train behind is the following:

$$t_b < t_f + d \times \left(\frac{1}{v_f} - \frac{1}{v_b} \right),$$

with t_b transit time of the train behind in the interception section, t_f transit time of the train ahead, d distance from interception section to the next section, v_f average speed of the train forward, v_b average speed of the train behind.

The firing of one of the three transitions causes in the TRENO_AVANTI02A place the substitution of the TRAIN token with the token TRAIN that before was in the TOR_POM place; this last token is now ready for the comparison with the next train.

The transition firing of PRECEDENZA02, like the figure 7 shows, creates also one TRAIN token in the output place of the class Movement Inspector named PRIORITY_TOR_POM.

This token represents the train behind and has a value in the PR variable that means "give overtaking".

The output place is linked on the Top Level to the track circuit located immediately before the next station (in this case Pomezia).

In the track circuit the new TRAIN token substitutes the other that arrives from the line running with the order of changing the routes inside the Pomezia station and waiting for the running of the train behind.

So the automatic overtaking is carried out.

This control mechanism is active on all the stations, in both directions and on two-way operated tracks.

A last fundamental consideration is that overtaking criteria above described are ever modifiable; this means creating more scenarios with different automatic overtaking systems and it is possible to choose the better criteria, which minimizes the average delay of the trains or that reaches other goals.

4   Conclusion

The proposed model can be applied in the studies of railway traffic regularity, for example comparing real measured delays with those simulated by the model in the same traffic conditions and finding the causes of specific criticalities.

The flexibility of outputs file allows the detailed analysis of each element of the line, so that it is possible to have an exhaustive framework of traffic problems depending on railway infrastructure.

Moreover the model can be effectiveness used as decision support tool both in phase of planning of new line with new operation timetable and in phase of improvement of existing lines with changed traffic conditions.

Finally the main model application with an automatic overtaking system is the support to traffic management decisions finalised to minimise the delays.

In fact, using more criteria for setting up the system, in case of delays of each train it is possible to define a decisional tree to limit the perturbation on whole traffic.

The proposed model has been applied on the test line Roma-Formia.

The goal is to describe a decision procedure that, depending on delay of each train in each station, allows an optimised traffic management.

References

[1] Ricci, S., Tieri, A., *Railway traffic regularity monitoring and forecasting: the use of the Petri Nets models*, Proc. of the 6th Symposium, 2007 FORMS/FORMAT 2007 – Braunschweig, 118–128.
[2] Impastato S., Ricci, S., Tieri, A., *Monitoring and forecasting of railway traffic regularity by a Petri nets model*, Proc. of the 4th Conference Challenges in Transport and Communication – Pardubice, 57–63, 2006.
[3] Malavasi, G., Raponi, P., Ricci, S., Spellucci, C., *Simulation of a complex railway station with a Petri Nets based interlocking model*, Proc. of the Workshop on software specification of safety relevant transportation control tasks – Braunschweig, 11–22, 2002.
[4] Malavasi, G., Ricci, S., *Railway Traffic Simulation by means of a Petri Net model. Computers in Railways VIII*, WIT Press, Southampton, 407–415, 2002.