# TRAFFIC AND INTERLOCKING SIMULATION IN RAILWAY OPERATION: THEORY AND PRACTICAL SOLUTIONS

Edit BARANYI*, Gábor RÁCZ**, Géza SZABÓ*, and Balázs SÁGHI*

*Department of Control and Transport Automation
Budapest University of Technology and Economics
H–1111 Budapest, Bertalan L. u. 2., Hungary
Tel.: +36 1 463 1013
e-mail: {baranyi|szabo|saghi}@kaut.kka.bme.hu
**Stellwerk Engineering Ltd.
H–1141 Budapest, Gödöllői u. 165., Hungary
Tel.: +36 309 910 990

## Abstract

The society demands high safety level, efficiency and economical operation of the railway traffic. One of the key factors to achieve these aims is the human – the personnel operating and controlling railway traffic as well as the personnel supervising track side and on-board safety equipment, such as signaling and interlocking systems, point machines etc. To ensure the low level of human errors, efficient training programs and supporting tools have to be created. Some of the supporting tools are simulators, simulating the traffic, the interlocking or the on-board man-machine interface for different purposes. In the paper, the way how simulators can be designed as re-configurable ones also the data base serving as basic information storage for simulation will be explained.

*Keywords:* interlocking, signaling, traffic simulation, training.

## 1. Introduction

In safety critical system such as transport systems the protection against human errors plays a significant role. Human error can be committed in the different phases of the life-cycle: during the system specification, during the development and in the longest phase of the life-cycle: during the operation. In the railway operation, several safety-critical tasks are assigned to the operators and not controlled (because of cannot be controlled) by the signalling and interlocking systems. A lot of these tasks is necessary in situations happened very rarely (emergency situations). Since these situations are unfamiliar to the operators, and the demand of good interaction is high, pre-training is necessary to provide information and practice to solve these situations [1].

The European standards concerning railway safety (leading with EN50126, about the specification and demonstration of railway reliability, availability, safety and maintainability) forces the efforts on training and re-training of operational personnel. The main equipment for the training of railway signalling personnel has become for nowadays the computer based interlocking, signalling and traffic

simulation systems [2, 3]. It is nowadays desirable that the operator and the maintenance personnel possess necessary knowledge of the equipment from the first moment of its operation. This is unavoidable to an untroubled, smooth and safe traffic operation. In case of extraordinary, troubled situations, the rapid recognition and solution of the given problem is required to avoid accidents. Computer-based simulation systems are capable to help in the above mentioned tasks.

In the paper, first the general requirements will be analyzed, which must be fulfilled by any railway operation purposed simulation system. With help of an example it will be demonstrated that if the simulated signalling functions are realized in a formalized, easily configurable way, the system provides additional possibilities too.

## 2. Functions and Requirements

According to the above mentioned requirements, the computer based railway simulation serves the following main goals:

- Training and retraining of the operative and control personnel that takes part in the management of the railway traffic;
- Training of the control personnel in case of failures of the signalling system.
- Check of the executability of operation plans;
- Preparation of the solution of conflict situations (e.g. in case of planned reconstruction works or other limitations);
- Training of the locomotive drivers;
- Test of new interlocking functions/options already in the design phase;
- Capacity/usage test of planned tracks;
- Capacity/usage analysis of existing tracks, for examination of necessary reconstructions;
- Try different locations for different trackside objects already in the design phase.

According to the goals described above, an ideal simulation tool must provide the following functions:

- Map of the simulated area for the simulation system;
- Simulation of the interlocking and the signalling system;
- Simulation of the trackside elements;
- Simulation of train and shunting movements;
- Display of the Man-Machine Interface (MMI) of the signalling system;
- Evaluation of simulation scenarios;
- Simulation of the onboard equipment of the locomotive(s);
- Simulation of different train control system (like ETCS).

The introduced functions are nowadays generally fulfilled by several railway simulation systems. However, in connection with the appearance of the high-speed railway traffic, and because of the unification trends inside the EU, a new requirement on simulation systems is that the user can easily and quickly change the simulation environment, in order to follow modifications of the real equipment and/or the user interface. Traditionally this can be carried out exclusively by the developer of the simulator. This problem can be solved by the development of such a *simulation development system* that allows the user

- to modify the functions of the interlocking system;
- to develop new object types, e.g. new signals;
- to edit of new symbols and to apply them in the simulation system;
- to modify the attributes of different objects;
- to modify the MMI;
- to set unique, railway specific parameters;
- to add new modules to the system, e.g. 3D locomotive driver display;
- to add ETCS equipment;
- to integrate the locomotive drivers' and the operators' training;
- to supplement the hardware of the system, according to the requirements;
- to configure the system always optimally, according to the given training purposes.

If these functions are realized, the system provides the followings:

- The user can easily modify the complement of the following functions:
  - operation of trackside elements;
  - edition of new elements (outlook and operation);
  - interlocking functions;
  - realize a new type of interlocking system;
  - modify or create a new MMI for the interlocking system;
  - modify train or shunting movements;
  - exchangeable train movement dynamics;
  - connect different types of interlockings.
- It is possible to append new program modules to the systems, e.g.
  - 3D display;
  - connection with other devices (e.g. road signals).

## 3. Structure of the Simulation System

As the example for the realisation of the theory, we will introduce the `RailCAD` simulation system [4], which currently realizes the following five basic functions:

1. Edit of the trackside objects (outlook and behaviour) in a railway station;

2. Production of the operation description of the edited railway station, based on the edited station database and the interlocking description file;
3. Simulation of the interlocking system and that of the trackside objects, including the failure behaviour;
4. Simulation of vehicle movements;
5. Simulation of the onboard equipment of the locomotives.

To assist the use of these basic functions, some additional tools are also available, like

- create new symbols or edit existing ones;
- database correctness check
- route collection tool.

Furthermore, the simulation system provides possibility to simulate the traffic according to a predefined timetable, with consideration of *Slow!* signals and with consideration of special track characteristics. The simulation may run online, according to a predefined scenario, with programmed events, like train movements, failures etc. The handlings can be evaluated after the simulation. It is also possible to simulate the transition between different train control systems.

The primary aim of the `RailCAD` system is to support the use (including modifications and extensions) of the simulation system without deep knowledge in programming techniques.

The whole system consists of several segments, which communicate with each other by a computer network. This means, that only those segments are necessary to run, which are really needed to the given tasks, and a further advantage is, that external segments, modules can be connected as well.

The output of each segment is a simple text file, which can be modified by the user, if certain rules are kept. This allows the user to establish very special, station specific behaviour in the interlocking system. The output files act as input files for other segments. With help of special initialization files special parameters can be defined, such as route types, route components, maximal flank protection distance and shunting speed limit.

The system consists of the following segments:

- Core programme.
- User interface programme.
- Symbol editor to establish new objects, their properties and their symbols, furthermore to modify of existing objects.
- Station editor to establish the station description database.
- Station check to check the correctness of the station database.
- Route generator collects the possible routes in an edited railway station
- Interlocking simulation, which simulates the operation of the interlocking system according to the operation description text files.

In the following, the task and the operation of the different program segments will be introduced.

The *gobserver* is the core system component of the `RailCAD` system. This component stores all the information, which must be available to any program segment. The *gobsym* segment is the symbol editor, which serves for the definition of new and the modification of existing elements or objects (*Fig. 1*).
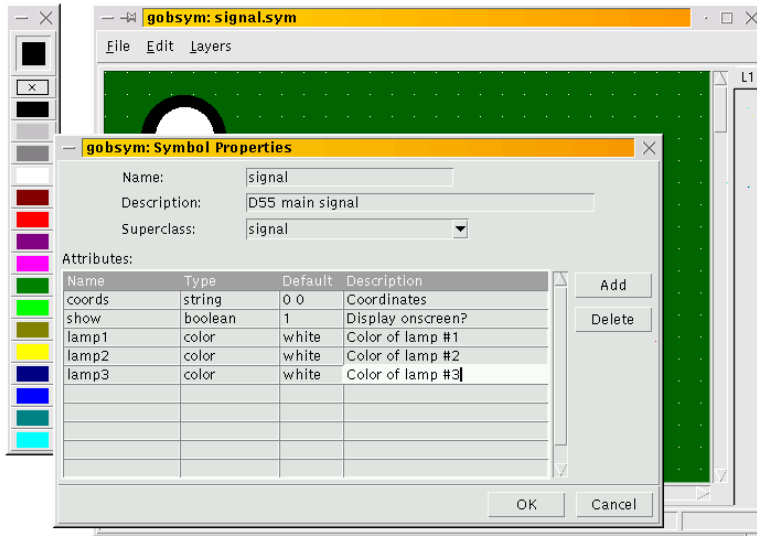


*Fig. 1.* Gobsym – the symbol editor

The *gobcad* segment – also called station editor – is used for the preparation of the database of a given railway station. The so called *gobui* (*Fig. 2*) is used for the handling and the display of the state of the trackside objects and other states. The *gobui* is connected to the *gobserver*.

The *route generator* collects the possible routes in a railway station (*Fig. 3*), after that the *station check* segment has checked the correctness of an edited railway station.

The task of the *interlocking simulator* program is to simulate the operation of the interlocking system, based on the text file descriptions (*Fig. 4*). A separate segment is used to simulate train and shunting movements (*Fig. 4*). To each simulated train a so called *MMI* segment can be coupled, which simulates the onboard equipment on the vehicles. This module enables both automatic and manual train driving (*Fig. 5*).

Since the single program segments can be distributed to several computers the number of the interlocking personnel (*gobui*) and the locomotive driver workstation (*MMI*) can be defined according to the actual requirements of the training.
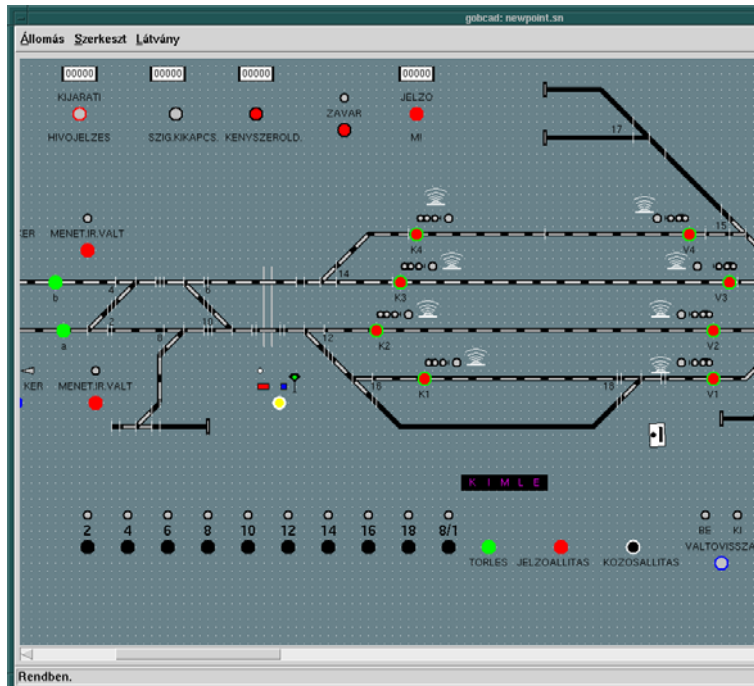
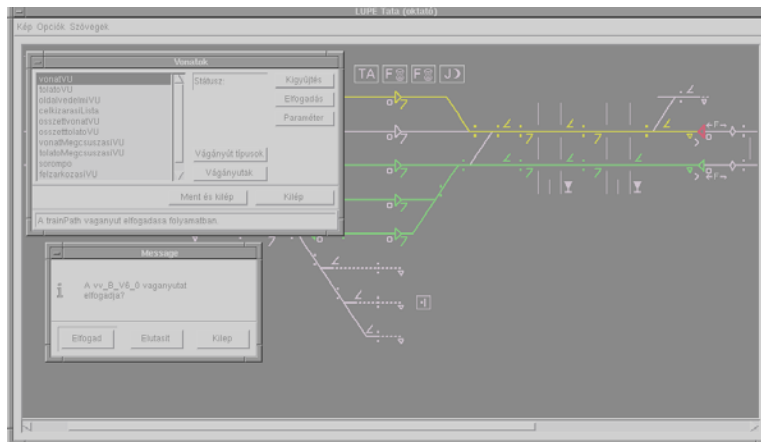Fig. 2. Gobui – user interface
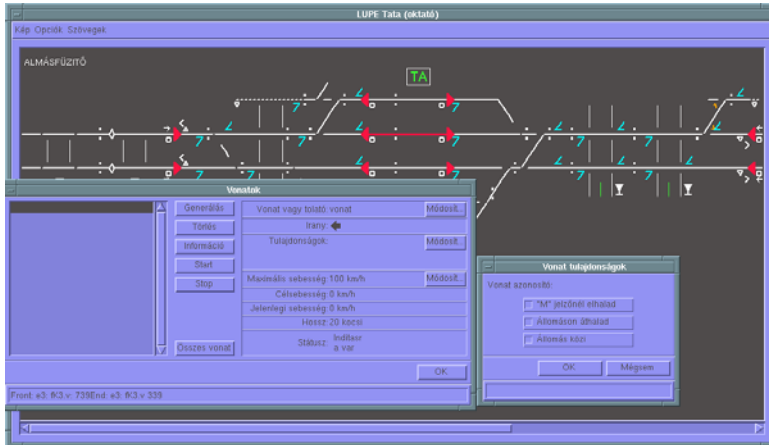


Fig. 3. Route generator in action

*Fig. 4*. The simulated operation of the interlocking system (in the background) and the generation of a new train (in pop-up windows)



*Fig. 5*. MMI – simulation of the ETCS onboard equipment

## 4.  Formal Description of the Railway Interlocking Operation

The interlocking simulation segment has the task, to interpret and to let run the files that describe the operation of the interlocking system. These are the files that describe the operation rules of the given type of interlocking system (created by the user) and the files that contain the elements of the routes (generated by the *route generator*).

The first one is a special text file that describes the principal behaviour of

the interlocking objects. Furthermore it contains the rules, necessary to the route search. This file is really easy to understand and can be modified as well.

The textual description of the operation of the interlocking system enables to describe any kind of trackside objects and to describe their behaviour as well. This means that special elements or usual elements with some special behaviour can also be inserted into the system. In the following a short example is demonstrated from the discussed description file (remarks in brackets).

*Example 1.* Collecting the elements of a train route

```
pathtype trainPath {  <route type is train route>
list elements =        <list of the elements that play a role in the route logic>
'elemlist(track,pValto_state,pJelzo_role,pFojelzo_role,
   pCel_role)';
                          <tracks, points in correct state, shunting signal, main
                          signal, targets>
                          <definition of other lists>
list elems = 'elemlist(track,pValto_state)';
                          <to detect occupancy, track sections and switches
                          must be collected separately>
list points = 'elemlist(pValto_state)';
                          <state of the switches>
list cross = 'elemlist(alsorompo)';
                          <list of level crossings>
```

*Example 2.* Check of a point, whether it can be locked or not

```
action LockOk {
                          <Checks if the point is in the correct state, is it controlled,
                          is there flank protection? If everything is OK, the variable
                          „lockOK" is set.>
match base.control, base.check, base.  sideok; set lockok;
    case state, yes, yes; values yes;
    case *, * , *; values no;
```

It is important to note, that the description language is not a programming language, but a railway interlocking definition language with defined syntax and semantics, developed for railway simulation purposes.

## 5. Conclusion

In the paper we presented the requirements that shall be fulfilled by configurable simulator systems and the benefits of a formalized, modularized realization. With help of the presented simulation design tool it is possible to realize the simulation of any user interface, any type of interlocking system and any onboard equipment. All of this can be done by the user of the system. Each program segment can be adapted to specific requirements with help of configuration text files. Inputs and outputs of the programs are also text files, which can be edited without deep programming knowledge. The text format enables to interface different programs, developed on different platforms too. The TCP/IP communication of the program segment enables to append external modules to the system, and that the segments can be distributed to multiple computers. The system enables to insert new symbols, and the modification of existing ones. Any stations can be edited, and with help of the interlocking description file, the operation of the interlocking is generated automatically.

## References

[1] ARANYOSI, Z. – MOSÓCZI, L. – RÁCZ, G. – TARNAI, G., Training the Personnel Using Simulator and Training System, *Word Congress on Railway Research WCRR '97 Proceedings*, **A** (1997), pp. 745–751.

[2] BARANYI, E. – RÁCZ, G. – SÁGHI, B., Application of Simulation Systems in Planning and Training of Railway Operation, in: *Formal Methods for Railway Operation and Control Systems. Proc. of Symposium FORMS2003. L'Harmattan*, Budapest, 2003.

[3] DOBROSI, A. – HRIVNÁK, I. – SCHWENDKE, R., Multifunction Simulator Developed For Hamburg/Simulation Based Training and Planning Systems for Railway Technology, *IRJ. International Railway Journal*, UK. **4** (1999), pp. 25–27.

[4] RAILCAD Users Guide. Stellwerk Ltd, 2004.