Experiences with the Formal Modeling of the Geographical and Tabular Principles of Interlocking Systems

G. Lukács*, B. Farkas**, T. Bartha***

**Budapest University of Technology, Stoczek str. 2, 1111, Budapest, Hungary, E-mail: lukacs.gabor@mail.bme.hu*
***Budapest University of Technology, Stoczek str. 2, 1111, Budapest, Hungary, E-mail: farkas.balazs@outlook.hu*
****Budapest University of Technology, Stoczek str. 2, 1111, Budapest, Hungary, E-mail: bartha.tamas@mail.bme.hu*

Abstract

The development of safety-critical, embedded microcomputer-controlled railway interlocking systems has seen an increasing interest in the use of formal modeling, due to its ability to specify the behavior using mathematically precise logical rules. These model-checked formal specifications increase the likelihood that the design of the interlocking system will be complete and correct.
This paper describes experiences with the formal modeling of the railway interlocking principles (i.e. geographical and tabular approaches) using an example of a simplified terminal area. In this paper we show two well-known modeling techniques (i.e. Petri nets and nondeterministic automata) and we compare the tools employed for the modeling and simulation (PetriDotNet and UPPAAL).
Our long-term goal is to prepare the automatic transformation of the non-formal specifications created by domain engineers into appropriate formal models which can be verified by existing model-checking tools.

KEY WORDS: *safety-critical systems, railway, interlocking, formal methods, transition system, Petri net, automaton*

1. Introduction

The application of formal methods in the railway industry has a long history. Many publications have been written in this area, but the theoretical results of transposition in the daily practice have not yet come to realization. The backgrounds of these are for example the extra work needs, which should be used as background knowledge, or the distrust, which accepts the prepared formal descriptions and analyses.

In this paper we explained a case study (a simplified station interlocking with geographical and tabular approaches) and we describe it with two formal methods, (Petri nets [1] and nondeterministic finite automata [2]). Through the examples of case study we would like to introduce that with proper construction we can create well-used models. We use PetriDotNet [3] (in the following PDN) and UPPAAL [4] tools to realize the interlocking principles. Based on this work we compared and evaluated our experiences with the tools from the point of view of usability in the railway interlocking systems.

2. Formal methods

The formal methods are techniques that we use primarily in the area of information technology. These methods are based on discrete mathematics and mathematical logic. These techniques can be used in the system development including the software and hardware development as well [2]. The semantics and syntax of formal models are well specified, clear and complete, so they do not have equivocal or not defined parts. The cooperation between the participants in the development process does not mean misunderstandings or uncertainty using formal models and specifications instead of textual and ad hoc marking systems which characterize the current engineering practice.

The formal methods support the identification of errors in the early life cycle phases. The executable models can be tested early in their production. Using a suitable analytical tool we can apply model checking allowing full control of the models. When a qualified code generator generates the code, then the proof of the correctness of the models will remain valid as a correctness proof of code in the scope of the verified properties.

The application of formal methods in the railway interlocking systems development is also prescribed by standards (e.g. EN 50128 or EN 50129). These standards [5, 6] classify these techniques as highly recommended requirements.

3. The railway interlocking principles and the case study

Based on constructional structure and the route logics imaging the railway interlocking systems can be classified into two essential groups: geographical and tabular interlocking [7]. In a tabular interlocking system the locking between signals and switches is achieved in form of a locking table that contains all locking conditions for all train routes. Tabular interlocking can be effected by cascade locking or by route-related locking. Cascade locking is not discussed in this article. In a geographical interlocking system the track elements are represented by logical objects connected to each other in form of the track layout.

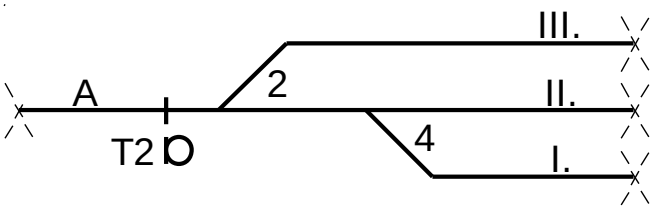

Fig.1 Case study: The topology

The topology of the case study is shown in Fig. 1. The topology introduces a simple part (start point) of a station whit three tracks (I., II., III.) and one shunt signal (T2). Our goal was to select (artificially construct) a topology which allows to set several train route. This is done with two switches (2, 4). Trains can be traversed from the open line (A) to one of the tracks (I., II., III.) on the station. The open line is marked with A section, which is not part of the models.

The realization of the route-related locking principle is summarized in the Table 1. The first part of the table gives the exclusions between the routes; the second part of the table gives the state of elements (e.g. switches, signals) which are required to lock the route.

Table 1

Case study: Route matrix

	A-I.	A-II.	A-III.	Points	
				2	4
A-I.	–	|	|	+	-
A-II.	|	–	|	+	+
A-III.	|	|	–	-	

Notation:
| : conflicting routes locked by plain locking
|| : conflicting routes locked by special locking
– : main diagonal in route matrix
+ : points locked in normal position
- : points locked in reverse position

Note. We only provided the notation what is necessary for the understanding the case study.

The realization of the geographical principle is summarized in the Fig. 2 and Fig 3. The case study contains nine objects, one start point, three target points (which are virtual targets), three tracks and two switches (see Fig. 2). Instead of the route-related locking principle the switches do not have predefined end positions. The necessary end position of switches always depends on the current route. The simplified operation of geographical principle is shown in Fig. 3. First we have to choose a Start and a Target point, after that the interlocking system executes the route search between the chosen Start and Target points. If the correct route is selected (designated), then all affected elements perform the relevant operations (e.g. locking). If all of this has been completed successfully (all relevant preconditions are true) then there is no further condition for the setting of the signal and the shunting movement of train (Fig. 3 Occupancy).

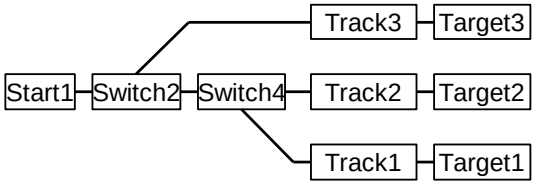

Fig. 2 Case study: The logical object and their connections

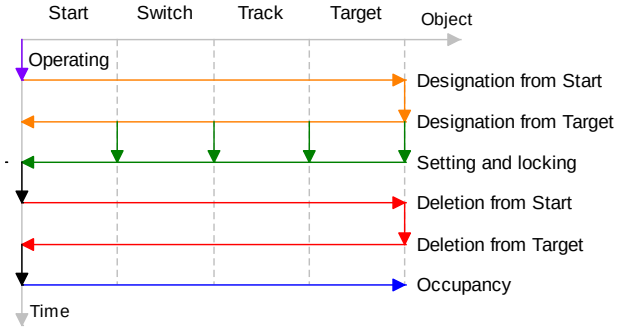

Fig. 3 Case study: The logical objects and their functions depending on time

We provide a simplified notation (see Fig. 4 and Fig. 5) to understand the models. This will be used consistently in the descriptions of the models.

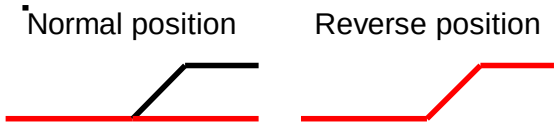

Fig. 4 Case study: The switch notation in the route-related locking principle

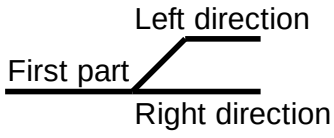

Fig. 5 Case study: The switch notation in the geographical principle

4. Modeling of the railway interlocking principles

In this Chapter we summarize the modeled functions, the constraints and simplifications of modeling. Finally, we briefly describe the basic operation of the models.

4.1. The modeled interlocking functions

The primary purpose of delimiting features was to show all the basic functions in the simplest form in the models. We modeled the selected functions for both railway interlocking principles:

1. route selection/designation,
2. setting and locking of track elements (e.g. switches, tracks),
3. locking of the route,
4. deletion the designation of the locked route,
5. setting the signal clear,
6. movement of a shunting train,
7. unlocking of the route (and the possibility of setting a new route).

Note that this Chapter is also a modeling constraint as well (see Chapter 4.2.).

4.2. Constraints and limitations of modeling

Our constraint comes from the specialties of the case study and they are partly our own requirements which are provisions against the complexity. All of these constraints and limitations can be resolved, and the models can be extended with them.

1. We did not use timings because our goal was to model the process. (The PDN and UPPAAL tool support the timings, too.)
2. Only one direction can be set routes: from starting point (Fig. 1 A) to one of three tracks (Fig. 1. I., II. or III.) The models are not handling the opposite direction movements.
3. We only modeled the shunting movements (we did not model the normal train movements). This makes the simplification that we do not have to model the occupancies to the locking of the track elements. We just modeled the occupancy information which is necessary for unlocking (track elements, route).
4. We did not model slip routes because the topology (Fig. 1) does not allow this.
5. We did not model flank protections because the topology (Fig. 1) does not contain the necessary elements.
6. We did not model the function of route storing.
7. We did not model the operating of interlocking (exception is the selection of start and target points).
8. We neglected some checks of real railway interlocking systems which refer to the route closing and locking.
9. We neglected all malfunctions and wrong operations (e.g. false occupancy).
10. We have modeled the movement of short train because of unlocking (taking into account the features of the PDN and UPPAAL tools).

4.3. Modeling the route-related locking interlocking principle

In this Chapter we give a short, general description about the modeling of route-related locking principle which is the same for the two tools (PDN and UPPAAL). There is a well defined initial state in the models. The first step is choosing the desired route which is implemented randomly in the models. The following parts of the model check the rules of route matrix (see Table 1) and adjust them. The route matrix can be mapped one in one into a Petri net or automata taking into account some simple rules. The modeling of routes (see Fig. 1 and Table 1) is very similar so we write about them in general. First the models set the switch(es), lock them and lock the tracks, too. Finally the models lock the selected route. Then the T2 shunt signal shows clear aspect and the shunting train movement can be performed. The unlocking of the locked route happens with the help of the dissolving sections. The models must perform the following sequence on the section: clear – occupied – clear. If the sequence is correct the track element and the route will be unlocked. After that it is possible to choose and set a new route.

4.4. Modeling the geographical interlocking principle

In this Chapter we give a short, general description about the modeling of geographical principle which is the same for the two tools (PDN and UPPAAL). In this case the focus is on the connection of the track elements between which communication is involved. The track elements themselves are able to do the interlocking functions (see Chapter 4.1). The first step is choosing the start and target points which process is implemented randomly in the models (see Fig. 3). In the next step a designation link moves all the way from start point to target point and after that from target

point to start point (see Fig. 3). If a switch gets the designation from the two directions, it can change its position (when it is needed) and then it can be locked. If a track element gets the designation from the two directions, it can be closed. In the next step a locking link moves all the way from target point to start point (see Fig. 3). If the route is locked the designation is no longer necessary, so it can be deleted. Thereafter the T2 shunt signal shows clear aspect and the shunting train movement can be performed. As the train moves it unlocks all track elements one by one. After all elements are unlocked it is possible to choose and set a new route.

5. Experiences with the Petri nets

During the modeling, we avoided the use of extended Petri nets (e.g. inhibitor edge, test edge etc.) to ensure that the models can be automatically verified. Because of the complexity of the case study we use hierarchical Petri nets to provide a transparency of the models. Further advantage of the hierarchical Petri nets is that we can separately and independently edit the subnets. We use coarse transition to ensure the relationships of the subnets. The coarse transitions are always part of the main net and hide the functionality that we implemented in the subnet.

In the next section we will only describe the switch subnets (only an instance of the switch) because of the large size of the models. We will show and compare the switch subnets using the route-related locking and geographical principles, too.

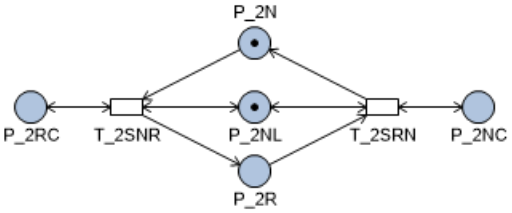

Fig. 6 Case study: The PDN model of the switch (switch subnet, route-related locking principle)

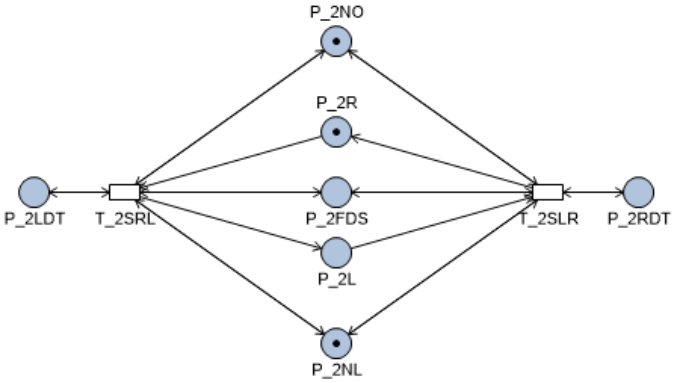

Fig. 7 Case study: The PDN model of the switch (switch subnet, geographical principle)

The initial state of the switch according to route-related locking principle (see Fig. 6) is that it is not locked (token on the P_2NL place) and it is in the normal position (token on the P_2N place). Note that other starting states are also possible, e.g. token on the P_2NL and P_2R places say the switch is in the reverse position. The position of the switch is adjusted by the two transitions (T_2SNR and T_2SRN).

The precondition of T_2SNR (Set from Normal to Reverse position) transition firing are P_2RC (Reverse Command) and P2_NL. If these preconditions are true (there is a token on the P_2RC and P_2NL places) and the switch is in the normal position (a token on the P_2N place) then the switch can change its position.

The precondition of T_2SRN (Set from Reverse to Normal position) transition firing are P_2NC (Normal Command) and P2_NL. If these preconditions are true (there is a token on the P_2NC and P_2NL places) and the switch is in the reverse position (a token on the P_2R place) then the switch can change its position.

Note that we did not model the occupancy of switch.

The initial state of the switch according to geographical principle (see Fig. 7) is that it is not locked (token on the P_2NL place), it is not occupied (token on the P_2NO place) and it is in the right direction (token on the P_2R place). Note that other starting states are also possible, e.g. token on the P_2NL, P_2NO and P_2L places say the switch is in the left direction. The position of the switch is adjusted by the two transitions (T_2SRL and T_2SLR).

The precondition of T_2SRL (Set from Right to Left direction) transition firing are P_2LDT (the Left direction is Designated from Target), P_2FDS (the First part of switch is Designated from Start), P2_NL and P_2NO. If these preconditions are true (there is a token on the P_2LDT, P_2FDS, P_2NL and P_2NO places) and the switch is in the right direction (a token on the P_2R place) then the switch can change its position.

The precondition of T_2SLR (Set from Left to Right position) transition firing are P_2RDT (the Right direction is Designated from Target), P_2FDS (the First part of switch is Designated from Start), P2_NL and P_2NO. If these preconditions are true (there is a token on the P_2RDT, P_2FDS, P_2NL and P_2NO places) and the switch is in the left direction (a token on the P_2L place) then the switch can change its position.

Note that there may be such a case that the switch is not need setting but also can be locked immediately.

6. Experiences with the UPPAAL specific finite nondeterministic automata

The UPPAAL specific finite nondeterministic automata (in the following FNA) are composed from states and state transitions. The state transitions may be the actions by the system (internal events) or external events from the environment. In the simplest case the modeling of the FNA consists of modeling of the states (circles) and states transitions (directed edges), which makes the model easy to read. During the modeling, we avoided the use of simple FNAs (e.g. we did not use parameterized FNAs).

With one FNA you can model only one system element or process. To describe the entire system we have to model the links between the elements. The communication between the FNAs can be done in several ways. The most basic communication form is the synchronization which is based on the principle of handshake.

In the next section we will only describe the function of the switch (the setting of the switch) because of the large size of the models. We will show and compare the switch FNAs using the route-related locking and geographical principles, too. The notation is similar as in Chapter 5.

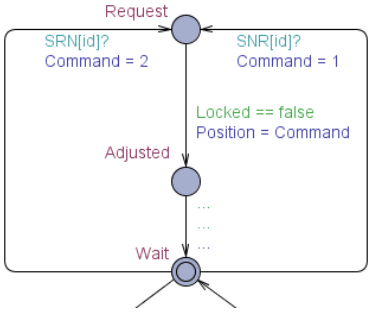

Fig. 8 Case study: The UPPAAL model of the switch (switch FNA, route-related locking principle)

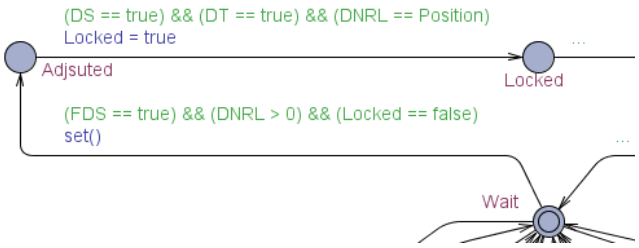

Fig. 9 Case study: The UPPAAL model of the switch (switch FNA, geographical principle)

The setting of the switch according to route-related locking principle starts from a Wait state which is an initial state (see Fig. 8). The initial state is represented by a double circle in the UPPAAL. This FNA is in the waiting state as long as one of the two synchronization (SRN[id]? or SNR[id]?) will not apply. If the SRN[id]? synchronization is applies then the value of the Command variable is update (Command = 2) which means (see Fig. 10) that the switch must be set in a straight position. Note that the SNR[id]? synchronization works same.

```
bool Locked = false;     // locked state of switch (false: not locked, true: locked)
int Route = 0;           // route in which the switch is locked
int [0,2] Command = 0;   // command of setting (0: No command 1: Reverse command, 2: Normal Command)
int [1,2] Position = 2; // position of the switch (1: Reverse, 2: Normal)
```

Fig. 10 Case study: Definition of the variables (switch FNA, route-related locking principle)

If the FNA is in the Request state and the precondition (the guard) is fulfilled (Locked == false) the position of the switch is updated according to the given Command (Position = Command). After that the switch is set to the Adjusted state. Note that on the Fig. 8 we did not detailed the closing and locking of the switch.

The setting of the switch according to geographical principle starts from a Wait state (see Fig. 9). If the guard ((FDS == true) && (DNRL > 0) && Locked == false) becomes true then the set() function does the setting of switch (see Fig. 11 to understand the abbreviations and the set() function). After that the switch is set to the Adjusted state. Note that we can see the locking process of the switch on the Fig. 9, too.

```
bool DS = false;         // Designated from Start (true: designated, false: not designated)
bool DT = false;         // Designated from Target (true: designated, false: not designated)
bool Locked = false;     // switch is Locked (true: locked, false: not locked)
bool Occupied = false;   // switch is Occupied (true: occupied, false: not occupied)
bool FDS = false;        // First part of switch is Designated form Start (true: designated, false: not designated)
int [0,2] DNRL = 0;      // one direction is Designated from target (0: Not designated, 1: Left, 2: Right)
int [1,2] Position = 2; // Position of switch (1: Left, 2: Right)

void set()               // setting the switch
{
   if ((Occupied == false) && (Closed == false) && (FDS == true) && (DNRL > 0)) Position = DNRL;
}
```

Fig. 11 Case study: Definition of the variables and functions (switch FNA, geographical principle)

7. A summary of the experiences

The Petri net and the finite nondeterministic automata are suitable for modeling railway interlocking systems. Utilizing their extensions we can make even more robust models, but these models cannot be automatically analyzed for all the tools.

The tool selected for the modeling of Petri nets (PDN) and FNAs (UPPAAL) is suitable for modeling and simulation operation of the route-related locking and geographical principles. The modeling of the geographical principle for both tools was much more complicated and resource intensive than the modeling of route-related locking principle.

We have experienced both advantages and disadvantages with the tools (PDN and UPPAAL). Without the necessity of completeness, we list some examples:

Advantages:

- Both tools are the same in the aspect of learnability, interpretability, utility.
- Built-in model checker and easy-use editing interface is available for both tools alike.
- etc.

Disadvantages:

- Help is not available in the used version of the PDN tool which makes it more difficult to answer the issues that came up during the editing of the models.
- etc.

In this paragraph we briefly summarize our experiences with the modeling of the interlocking principles. In the geographical models all information is at the track elements. The track elements communicate with their neighbors and transfer the locally stored information. In contrast, the route-related locking models have a central logic which can verify the status of each track elements and give commands to the track elements to change their status.

Our models and their detailed presentation can be downloading from [8].

8. Conclusions

The challenges of the modeling in the practical life are usually very large, difficult and various. To solve these problems, it is essential to choose the corresponding tool or tools. During the work we started to collect and categorize those aspects which can be affected the outcome of the selecting tools. We give a not complete shortlist about the aspects without the category and weight: built-in model checker, manageability of the editing interface, simulations skills, etc. This area may be the subject of further research.

Another aspect of the problem in the previous paragraph is a design question: Is it possible to decompose the system and model? How can we decompose the system and models? The large systems are confusing for the people, but the formal methods – i.e. modeling and model checking – support the understanding of the system, the clarification of the specification and the creation of a proper decomposition.

The modeling that we discussed in this article will allow covering additional research areas. It is possible to analyze explicit properties of the Petri nets (e.g. boundedness, deadlock freedom, reversibility, pureness) or model checking is available for both tools (PDN and UPPAAL) so we are able to verify the fulfillment of the functional and safety requirements on the models.

References

1. **Murata, T.** 1989. Petri Nets: Properties, Analysis and Application. Proceedings of the IEEE, Vol. 77 No. 4: 542-599.
2. **Ésik, Z.; Gombas, É.; Németh L.Z.** 2011. Hardver- és szoftverrendszerek verifikációja. TYPOTEX, ISBN 978-963-279-497-6
3. **PetriDotNet.** Available from: https://inf.mit.bme.hu/research/tools/petridotnet
4. **UPPAAL.** Available from: http://www.uppaal.org/
5. **MSZT.** 2011. MSZ EN 20128. Vasúti alkalmazások. Távközlési, biztosítóberendezési és adatfeldolgozó rendszerek. Szoftverek vasúti vezérlő- és védelmi rendszerekhez
6. **MSZT.** 2003. MSZ EN 50129. Railway applications – Communication, signaling and processing systems – Safety related electronic systems for signaling
7. **ERA.** Glossary of railway terms. Available from: http://www.era.europa.eu/document-register/documents/glossary%20of%20railway%20terminology-selection-%20en-fr-de.pdf.pdf
8. Our models and their detailed presentation. Available from: http://kjit.bme.hu/index.php/hu/folap/17-kutatas/352-formalis-modszerek-kutatocsoport