Safety critical software construction using CPN modeling and B method's proof

Zakaryae Boudi, El Miloudi El Koursi, Simon Collart-Dutilleul

▶ **To cite this version:**

Zakaryae Boudi, El Miloudi El Koursi, Simon Collart-Dutilleul. Safety critical software construction using CPN modeling and B method's proof. SESA 2014, Software Engineering and Systems Architecture, Dec 2014, Tétouan, Morocco. 4p. hal-01263465

HAL Id: hal-01263465
https://hal.science/hal-01263465v1

Submitted on 27 Jan 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Safety critical software construction using CPN modeling and B method's proof

Zakaryae Boudi; El Miloudi El Koursi and Simon Collart-Duttilleul

French Institute of Science and Technology for Transport, Development, and Networks

IFSTTAR-COSYS-ESTAS, Villeneuve d'ascq

Abstract: *Meeting the strict safety requirements in critical software development is today crucial for the safety-related industrial environment, especially railways. To be able to prove that all safety properties are captured in the system requirements and software specifications, as well as that the final software product satisfies all specifications, a formal approach is the most convenient. Indeed, the use of formal means of description in the development process was highly recommended by the CENELEC standard. Accordingly, this paper presents by means of a level-crossing gate controller case study, the practical use of Colored Petri Nets transformation into B abstract machines for safety critical software development.*

Keywords: Software development, Colored Petri Nets, B method, Formal verification, Railway safety

1 Introduction

Transportation sectors, especially railway systems and aviation, are known to be the primary domains to apply formal methods in software design and validation [1]. For instance, in France, the functional requirements of the SACEM system present in RER Line A in Paris were formally constructed in the B language [2] as well as for the automatic train system of the metro line 14 which was the first driverless metro line in Paris [3]. In this context, one of the ultimate goals of this work is to show in a practical manner the benefit from incorporating Colored Petri Nets in safety critical software development process using a conversion methodology aiming to provide B abstract machine preserving all the relevant information contained in the Petri Net models (fig. 1). Thereby, not only the understanding of the system requirements is reinforced, but an additional input is made available for the use of the B method validation tools, particularly when the considered system modeling is much easier using the graphical form of Petri Nets. In fact, the common use of Petri Nets in railway requirements modeling within industrial and research stakeholders [4, 5, and 6] makes the combination of their graphical modeling power and the B formal verification tools very interesting.

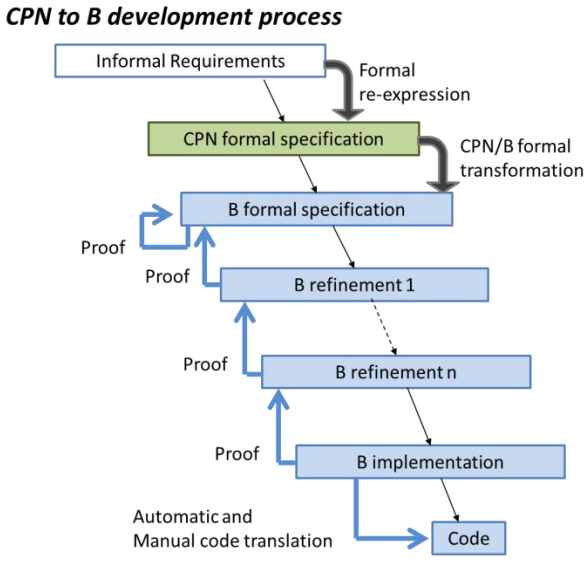

Fig. 1. Waterfall CPN to B development process

The transformation rules of the Colored Petri Nets conversion to the B abstract machines are developed as part of the national French project called “PERFECT” (Performing Enhanced Rail Formal Engineering Constraints Traceability) [7]. This scientific research project focus is to provide formal techniques and approaches in order to determine, if possible, the compliance between the ERTMS (European Rail Traffic Management System) specifications and National railway safety properties.

At first, this paper will introduce the study case example consisting of a level crossing gate controller and present the corresponding CPN model. Afterwards, the obtained B machines will be described and the safety properties introduction and verification will be shown. Finally, some interesting perspectives of this work will be stated.

2 Case-study: LX gate controller

2.1 Description

The example we consider in this work, for the matter of space and clearness, consists of a simple level-crossing gate controlling device. The level-crossing gate can have only two possible states; it is either up or down. Apart from the case of a train passage, the normal gate position is the “up” one. When an arriving-train is detected through the train-arriving sensor, which is launched by the passage of the train toward the level-crossing, the controlling device commands the lowering of the gate. As soon as the rolling away of the train, the leaving-train sensor is launched and the controller commands the raising of the safety gate. Of course, the safety property

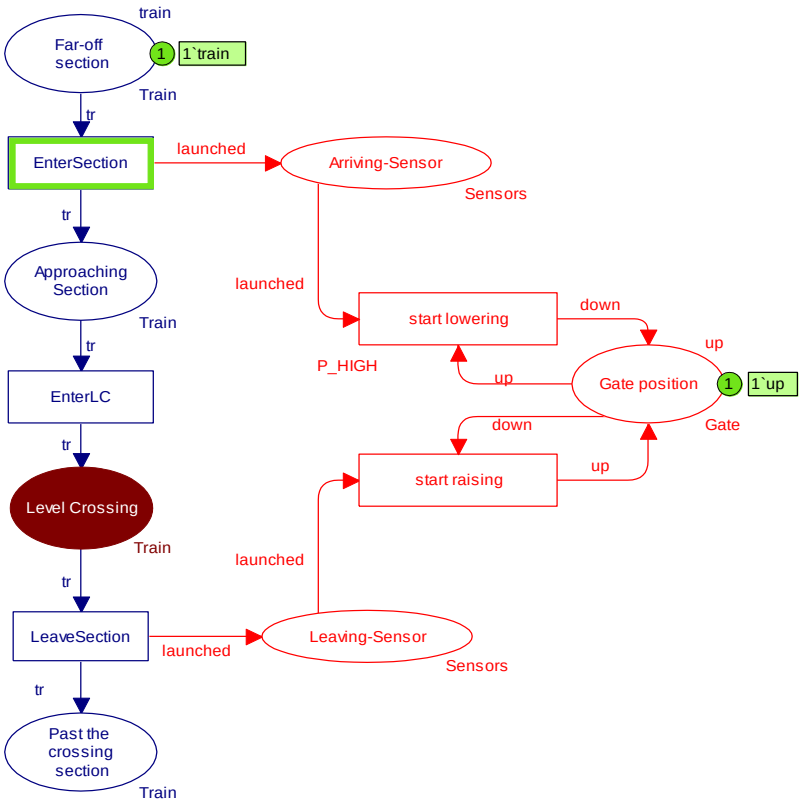

Fig. 2. Level-crossing gate controller CPN model

to check in this example is that the gate has to be in the "down" position if a train is present in the danger zone, i.e., between the two sensors.

2.2 LX gate controller CPN model

As presented in figure 2, the system specification is modeled using a Colored Petri Net. The modeling tool used in this study is the CPN-tools software platform. The main architects behind the tool are Kurt Jensen, Soren Christensen, Lars M. Kristensen, and Michael Westergaard [9].

2.3 The corresponding B machine

Each Place of the CPN model has some corresponding variables in the output B abstract machine. These variables denote the state of the place and the occurrence of the tokens in it. Similarly, the transitions states are also represented using variables in the B machine. The evolution of Petri Net states is mapped using the B machine operations. Figure 3 shows extracts from the obtained B machine. The next section will describe the use of the latter in safety properties verification.

2.4 Safety properties verification

The first property to check is that the invariant corresponding to the Petri Net place representing the level-crossing gate is always taking one state at the time, either up or down. The B invariant of such property is expressed by the following expression:

(occ_up_Gate + occ_down_Gate=1)

Furthermore, the safety invariant relevant to the gate position when a train is within the level-crossing area is expressed by:

(occ_train_LevelCrossing =1 => occ_down_Gate =1)

This invariant denotes also that the arriving sensor is launched by the rolling of a train. Theses invariants are added to the "INVARIANTS"

```
MACHINE LX_Gate

SETS
Gate = {up,down};
Sensors= {launched};
Train = {train}

VARIABLES
state_FarOffSection,state_ApproachingSection,state_LevelCrossing,state_Past
TheCrossingSection,
state_ArrivingSensor, state_LeavingSensor,
state_Gate,
occ_up_Gate, occ_down_Gate,
occ_train_FarOffSection,occ_train_ApproachingSection,
occ_train_LevelCrossing, occ_train_PastTheCrossingSection,
occ_launched_ArrivingSensor,occ_launched_LeavingSensor,
enabled_EnterSection,enabled_EnterLC,enabled_LeaveSection,enabled_StrartRai
sing,enabled_StartLowering
```

```
OPERATIONS
Op_Enabled_EnterSection=
PRE (train |-> 1):state_FarOffSection
THEN enabled_EnterSection := TRUE
END;
Op_Enabled_EnterLC=
PRE (train |-> 1):state_ApproachingSection & occ_down_Gate=1
THEN enabled_EnterLC := TRUE
END;
```

Fig. 3. Extracts from the obtained B abstract machine

clause in the obtained B machines. Thereby, the model checking of the B method tools “ProB” and “Atelier B” proof are applied to the machine. In this example, both tools validated the trueness of the system safety design since the proof had reached a rate of 100% (fig.4).

Component Status for LX_Gate

AutoProved C:/Program Files (x86)/Atelier B free 4.1.0/bbin/win32/workspace/LevelCrossingGateController/LX_Gate.mch

	nPO	nPRi	nPRa	nUn	%Pr
Initialisation	8	0	8	0	100
Op_Enabled_EnterSection	0	0	0	0	100
Op_Enabled_EnterLC	0	0	0	0	100
Op_Enabled_LeaveSection	0	0	0	0	100
Op_Enabled_StartLowering	0	0	0	0	100
Op_Enabled_StrartRaising	0	0	0	0	100
Op_Fired_EnterSection	12	0	12	0	100
Op_Fired_EnterLC	8	0	8	0	100
Op_Fired_LeaveSection	12	0	12	0	100
Op_Fired_StartLowering	12	0	12	0	100
Op_Fired_StrartRaising	12	0	12	0	100
LX_Gate	**64**	**0**	**64**	**0**	**100**

Fig. 3. “Atelier B” Proof results

3 Conclusion and future work

In the scope of safety critical software development, this paper presented, through a case study, the practical use of the combination of Colored Petri Nets and the B method in the verification process. The present work, as part of the PERFECT project, has a number of perspectives; the first one is the development of an elaborate transformation rules catalogue and the definition of a refinement methodology applicable to the obtained B machines. Besides, the project aims at providing a tool for the automatic transformation based on the theoretical rules.

References

[1] Havelund, K., Lowry, M., Park, S., Pecheur, C., Penix, J., Visser,W., White, J-L.: Formal Analysis of the Remote Agent Before and After Flight. In The Fifth NASA Langley Formal Methods Workshop, Virginia, June 2000.

[2] Guiho, G., Hennebert, C.: SACEM Software Validation. In ICSE ’90: Proceedings of the 12th international conference on Software engineering, pages 186–191, Los Alamitos, CA, USA, 1990. IEEE Computer Society Press.

[3] Behm, P., Desforges, P., Meynadier, J-M., Météor: An Industrial Success in Formal Development. In Didier Bert, editor, B‘98: Recent Advances in the Development and Use of the B Method, number 1393 in Lecture Notes in Computer Science. Springer, 1998.

[4] Boudi, Z., El-Koursi, EM., Collart-dutilleul, S., Khaddour, M.: High level Petri net modeling for railway safety critical scenarios. Proc. of 10th FORMS/FORMAT symposium on formal methods, pp. 65- 75. Braunchweig, 2014.

[5] Sun, P., Collart-dutilleul, S., Bon, P. : Formal modeling methodology of French railway interlocking system via HCPN, 14th International conference on Railway Engineering De-sign and Optimization, Rome, Italy, 2014.

[6] Antoni, M.: Formal validation method for computerized railway interlocking systems. In Computers Industrial Engineering, 2009. CIE 2009. International Conference on, pages 1532–1541, 2009.

[7] Lanusse, A., Ferlin, A., Ben-Ayed, R., Sun, P., Boudi, Z., Sallak, M. Preliminary methodological propositions for certification support using formal methods within a model driven approach (Propositions méthodologiques préliminaires pour le support à la certification à l’aide de méthodes formelles selon une approche orienté modèles). Technical report, deliverable D2.1.1, the PERFECT project, 2014.

.