2117

Validating the Specification of Automatic Train Protection and Block System

Guo XIE, Tomoya Kuroda, Hiroshi Mochizuki, Sei Takahashi, Hideo Nakamura
(College of Science and Technology, Nihon University. 7-24-1 Narashino-dai, Funabashi, Chiba)

In the development of automatic train protection and block (ATPB) railway signalling system, in order to guarantee the safety and reliability of its software system, formal method is adopted to analyze specification formally. Firstly, in order to improve the accuracy of translating original specification into formal specification, dynamic state translation is extracted to express the internal operation mechanism and state changes for every component, e.g. point, signals, trains. Followed by a UML model is created for a comprehensive and object-oriented analysis of the ATPB. Thirdly, a rigorous specification of system is established in VDM++ unambiguously. Lastly, the internal consistency is guaranteed by discharging the proof obligations. The satisfiability is validated by the systematic testing.

Keywords : railway signalling system, ATPB, formal methods, specification

1. Introduction

Considering the severe situations that regional train lines are facing, a novel railway signalling system, i.e. Automatic Train Protection and Block (ATPB) [1], is proposed to aid in reconstructing and improving the efficiency of regional train lines. Railway system is a safety-critical system, and its software system belonging to Software Safety Integrity Level (SSIL4), whose failure could potentially result in loss of human life or damage to environment. Therefore, considerable attentions have to be paid to it. Traditionally, the software safety was guaranteed by testing. However, the cost is high even errors can be found. More important, the test approach is always not inadequate for railway system, which has a high number of possible evolution paths and failure modes [2]. On the other hand, as recommended by international standards, railway systems, such as level-crossing control, digital ATC, and interlocking signaling, are analyzed by various formal methods (FMs), such as B, VDM, and Petri Nets.

In our project, the specification of ATPB is originally written by Japanese, and may contain contradictions and ambiguities too. Considering the numerous advantages of FMs, we intended to apply it to the ATPB. However, owing to various difficulties [3], almost all of these applications just focus on some critical subsystems. For example, rare work formalized onboard system and its communication with control center by GPS at same time, that makes there is few reference and available experience. In addition, until now, most existing works on formal method mainly focused on an existing formal method. However, in fact, the system properties must be stated precisely and formally by natural language firstly (i.e. Japanese in our project) before formal analysis. In that case, how to guarantee the formal specification is fully consistent with the original Japanese specification is a huge challenge. M. A. Ferreira and his group noticed this problem firstly, and a new strategy, i.e. "All-in-one" is proposed [4] to analyze the specification of software. In their approach, the PF-notation is adopted to check the architectural design. However, the notation of PF is abstract, and difficult to translate natural language specification to a formal form. After that, another formal language "alloy" is adopted to check the un-interpreted and unexpected counterexamples. Also, it is just effective to an existing model, and the challenge of building formal model from natural specification is still remaining, which is a key factor that limits the application of formal method into industrial practice.

Faced with the issues above, in our project, firstly the dynamic state transition is created to express the internal mechanism to help system designers check and analyze the system parameters quantitatively. Moreover, it improves the railway engineers and software experts achieving a common understanding on system operation of ATPB. Then the object-oriented modeling approach UML is adopted to express the structure, behavior, architecture of ATPB by use case, component diagram, class diagram, activity diagram, and so on. It is comprehensive, graphical, easy to use, and has to be one of the most widely used analysis tools in software engineering. Thirdly, considering that there is no counterpart in textual style in UML mode, the formal language VDM++ is adopted to formalize and verify the specification. It is selected as the following reasons: the purpose of this paper is to describe the function requirement of ATPB, that is the strength of VDM++, and what is more important is that it is similar to that of object-oriented programming languages (e.g. C++, Java): structural aspects of software are specified using classes and instance variables, that makes it easy to use for developers. Meanwhile, it is supported by an industry-strength tool, VDMTools, which offers syntax and type checking and proof obligation generation capabilities.

The rest of this paper is organized as follows: The overall structure of the ATPB system is shown in section 2. Followed by the process of establishing dynamic state transition is introduced in section 3. Section 4 creates the UML model for ATPB. Section 5 introduces the formal specification and analysis of ATPB by VDM++. Lastly, section 6 is the conclusion and future work.

2. System Structure

The ATPB system is a centralized radio railway system. It is a combination of ATP (Automatic Train Protection) and Block Control System. The main structure of this system is shown in Fig. 1. It consists of two parts, the onboard system and control center. The former is composed of: (1) state detection; (2) automatic train protection (ATP); (3) communication subsystem; and (4) record system. The later is mainly composed of: (1) communication system; (2) interlocking system; and (3) traffic operation management, which creates and revises train timetables.

[No. 2117] 日本機械学会 第19回鉄道技術連合シンポジウム講演論文集 [2012-12. 5～7. 東京]

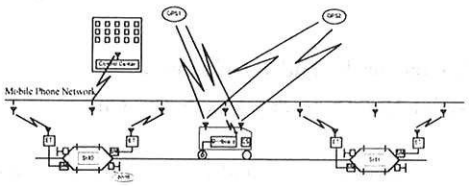

Fig. 1 Structure of ATPB System

The specification above is not an exhaustive description of ATPB system. Then considering that the system safety is the most important thing, a discussion will be made on system specification in the following by formal method.

3. State Transition

The state transition is expressed to graphically show the changes of state for every object, and to support a common understanding of operation mechanism. There are three steps to achieve it.

(I) Determine physical structure

This step creates the topology graph for rail lines, including signals, stations, points, etc. For example, according to the state change of train, the railroad is divided into three sections, i.e. "home", "departure", and "block".

(II) Extract mathematical model

After determined the objects range, a mathematical expression is extracted. It is a static and abstract expression of the system structure, by which can establish a quantitative statement between train, signals, and communication spots.

(III) Establish state transition

Based on the representation above and the Japanese specification, the state transition is established to show the train running operation graphically as well as rigorously. It is mainly made up of three parts: (1) State of train; (2) Jump condition; and (3) Event or message. It plays as a communication bridge between system engineers and developers. It also can help railway signal engineers to check the human errors, such as whether the model or original specification is consistent with their expectation.

4. UML Model of ATPB

Based on the natural language and state transition, the ATPB is modeled with UML, through use case for modeling the business processes, sequence diagram for modeling message passing between object, state diagram for modeling the behavior of objects, class diagram for modeling the static structure, and deployment diagram for modeling the static structure. In this paper, the sequence diagram for train approaching station shown in Fig. 2 is demonstrated as an example. The processing sequence is graphical, comprehensive and easy to use. It doesn't mean that all of these diagrams are indispensable. Actually, according to different system, the requirement of diagrams is also different.

5. Formal Specification and Analysis

The ATPB is formalized in VDM++ this section. The VDM++ model and UML class diagram has the same data structure, that means that they will have the same name of class, variables and operations. According to the original specification, the formal specification is built. As an example, class *SpeedCompare* is expressed as Fig. 3. Where Factor = 0.1 is a constant, which indicates the buffer capacity for the speed. The union type state defines three speed states: (1) If the speed is under 90% (1 - Factor) of the maximum speed limit, i.e. permitted speed limit, the safe speed signal <NormalSpeed> will be returned; (2) If a speed of up to 90% but less than the maximum speed limit, i.e. warning speed, the <WarningSpeed> will be returned. (3) If the speed exceeds the maximum speed limit, the signal <OverSpeed> will be returned.

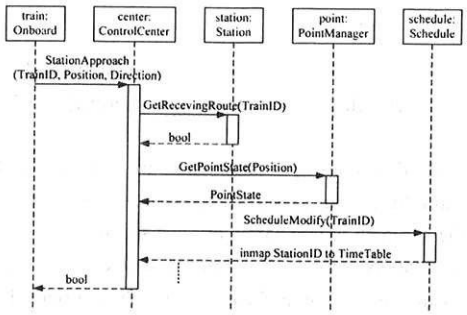

Fig. 2 Sequence Diagram for Train Approaching Station

```
class SpeedCompare
types
public State = <NormalSpeed> |
               <WarningSpeed> | <OverSpeed>;
values
 Factor: real = 0.1;
functions
 public SpeedState: real*real -> State
 SpeedState (speed, maxspeed) ==
 if speed < (1 - Factor)*maxspeed
 then <NormalSpeed>
 elseif
  (1 - Factor)*maxspeed <= speed and
  speed < maxspeed
 then <WarningSpeed>
 else <OverSpeed>;
end SpeedCompare
```

Fig. 3. Specification of *SpeedCompare*

After the VDM++ model was established, the syntax errors were checked by tools, the proof obligations are discharged to ensure internal consistency, and the systematic testing is implemented to ensure satisfiability.

6. Conclusion and Future Work

This paper proposed a new strategy to analyze the ATPB system by taking and utilizing the advantage of an integrated verification tool-chain. More work will be implemented on the communication between trains, control center and stations.

Reference

1) H. Nakamura: "Developments towards the safest railway in the world through sophisticated train control system", IEICE Technical Report, Vol.110, No.472, pp.21-28, (2011)
2) C. Bernardeschi, et al: "A formal verification environment for railway signaling system design", J. Formal Methods System Design 12, pp.139–161 (1998)
3) A. van Lamsweerde: "Formal Specification: a Roadmap," in A. Finkelstein, editor, The Future of Soft. Eng., ACM Press, 2000, pp.149-159.
4) M. A. Ferreira and J. N. Oliveira: "An integrated formal methods tool-chain and its application to verifying a file system model", In SBMF '09, vol. 5902 of LNCS, pp. 153–169. Springer, 2009.