

Modeling and Verification for Track Circuit Encoding in Train Control Center Based on UML and TA

Lei Yuan*

State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China
Corresponding author: lyuan@bjtu.edu.cn

Lijuan Wang

State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China
Email: wljqc@gmail.com

Dewang Chen

State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China
Email: dwchen@bjtu.edu.cn

Amie Albrecht

Scheduling and Control Group, Centre for Industrial and Applied Mathematics, University of South Australia, Mawson Lakes 5095, Australia
Email: amie.albrecht@unisa.edu.au

Abstract—The correct implementation of Train Control Center (TCC) software has great significance on the safe operation of high-speed railways. There are some problems in the currently used timed automata (TA) method for modeling TCC software, such as subjectivity and uncertainty in the modeling. In order to better verify the features of TCC software, this paper presents a new modeling method which combines Unified Modeling Language (UML) and Timed Automata (TA). As track circuit encoding function is a key function in TCC software, this paper takes it as an example to show the effectiveness of the proposed modeling method. Firstly, the UML class diagrams and state diagrams of TCC software were established based on the detailed analysis on TCC software. Secondly, the TA models are built according to the corresponding rules between UML and TA model. Finally, TA models were simulated and validated in the validation tool, UPPAAL. The simulation results show that TCC software can meet its functional requirements and performance requirements.

Index Terms—TCC software, timed automata, UML, UPPAAL, track circuit encoding

I. INTRODUCTION

According to the general technical specifications of CTCS (Chinese Train Control System), CTCS system consists of ground equipments and on-board equipments [1-2]. In this specification, train control center (TCC) is one of the ground equipments and the core safety equipment in CTCS which is directly related to safety running of train [3]. Meanwhile, TCC is composed of TCC software and some hardware devices. Therefore,

modeling and verification of TCC software have great significance on the safe operation of high-speed railways. In this paper, we focus on the correct implementation of TCC software which is related to the running safety of high-speed trains [4].

There are some researchers who proposed a safety evaluation method for TCC software based on simulation and debugging [5]. However, simulation and debugging cost a lot of time and cannot assure that all conditions are verified. Meanwhile, other researchers used Petri net to model and verify TCC software [6]. But there are many elements existing in Petri net models, which make the net difficult to be understood [9-12].

To overcome the shortcomings of the above studies, we are trying to combine UML (Unified Modeling Language) and TA (Time Automata) to model TCC software in this paper. UML is a common graphical description modeling method for object-oriented development. Since UML can provide favorable conditions for describing systems on the space, it is an important tool for analyzing software in engineering at present. A graphic depiction of system on the space is able to clearly describe the system's internal attributes [7]. To some extent, sequence diagrams in UML are rigor with integrity which has been proven in practical applications. TA is another modeling method which is commonly used to model the behaviors of real-time system. It defines real-time behaviors of real-time systems by defining time language [8]. Once TA and UML modeling method are combined together to validate the design of software, the subjectivity and uncertainty in models can be effectively avoided which is caused by TA

modeling method used alone. This combining method takes full use of UML modeling characteristics and describes software system in a more objective way. In this paper, UML and TA method are combined together to model track circuit encoding process, a key function of TCC software. UML and TA models are built at first in this paper. Then, the functions and performance of the built models are verified in a model validation tool, UPPAL.

II. ESTABLISHMENT OF UML MODEL

A. Establishment of UML Class Diagram for Track Circuit Encoding Process

In this paper, we take the encoding process in ZPW-2000 track circuit as an example. ZPW-2000 is an unisolated frequency shift auto-blocking system which is commonly used in China. The track circuit encoding function is a key function in TCC software, which achieves its encoding function based on the analysis of TS (track side) information and CI (computer interlock) approaching information[13-15].

According to the track circuit encoding rules in ZPW-2000, the encoding process is divided into four modules in UML class diagrams. The first module is the main logic processing of encoding which achieves the trigger of encoding, process control and the controlling of encoding direction. It consists of two classes, GetCode and CycControl. The second module is TS information processing which achieves the judgment of track circuit state and the sorting of occupied sections. It includes two classes, GetBlockState and GetBlockFromDB. The third

module is interlock information processing which realizes the obtaining of approach information and determines the type of approach. This module includes only one class, GetRouteState. The final module, the encoding module, accomplishes the encoding of low frequency for each section. It is composed of eight classes and each class respectively achieves encoding of up or down line, up or down driving and front or after station. Since the encoding logical is the same for the different driving directions and different lines and there are some differences in the encoding of front or after station, we just modeled on front and after station in the context of down line and down driving to illustrate the modeling process. Therefore, this paper divided this process into two classes, GetCodeRearXX and GetCodeAheadXX.

B. Establishment of UML State Diagram for Track Circuit Encoding Process

UML state diagram is used to show the behaviors and states owned by an object. The state of an object is determined by its current actions and conditions. In addition, state diagram displays the possible states that an object may be transferred into led by status changing [16-17].

This section established UML state diagram for track circuit encoding process in ZPW-2000. State diagram on each function point which is considered as an object is respectively established. Detecting whether occupied section is in the jurisdiction is shown as an example in this section. The established state diagram is displayed as follows:

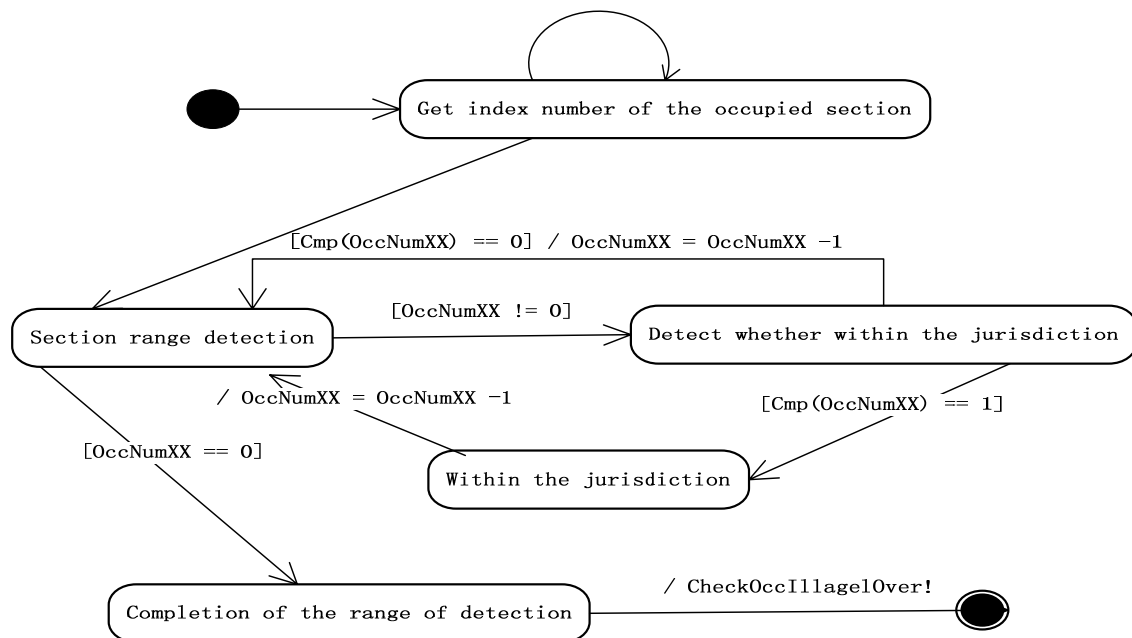


Figure1. UML state diagram for detecting whether occupied section is in jurisdiction

The above state diagram is invoked by ‘obtaining occupied information from TS’. This state diagram provides the function of occupied detecting. It tests whether the occupied section is valid and within TCC

jurisdiction or not by comparing the acquired occupied section number with section number in line data.

III. ESTABLISHMENT OF UML&TA MODEL

In accordance with corresponding rules between UML and TA, each member automat is established corresponding to the established UML state diagram. Therefore, TA models are divided into four modules according to the corresponding relationship rules. Each module makes up a timed automata network which is made up of multi-member automat. However, each member automat is corresponding to an UML state diagram[18-20].

We take ‘CI information processing module’ for description. Four member timed automat corresponding to four UML state diagrams are respectively established as ‘Obtaining approach’s status’, ‘Detecting the effectiveness of approach’, ‘Getting the approach’s flag’ and ‘Determining the type of approach’. However, the TS information processing timed automata network model is built according to the definition of TA. They are GetRouteMsg||CheckRouteI||GetRouteFlag||JudgeRouteType which are referred to U&A_CIMsgHandle model. They were shown in Figure 2 (a, b, c, d):

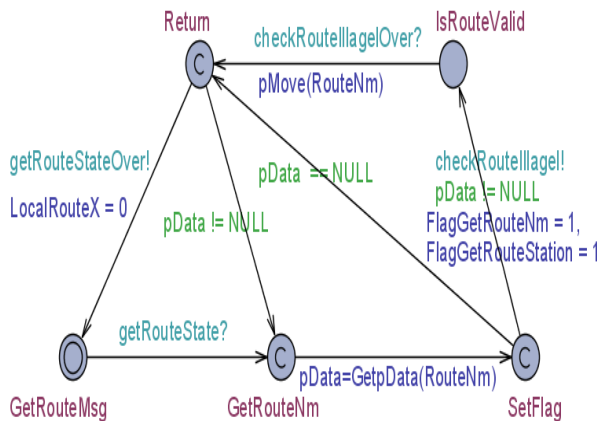


Figure 2(a) The TA model of ‘Getting status of the approach’

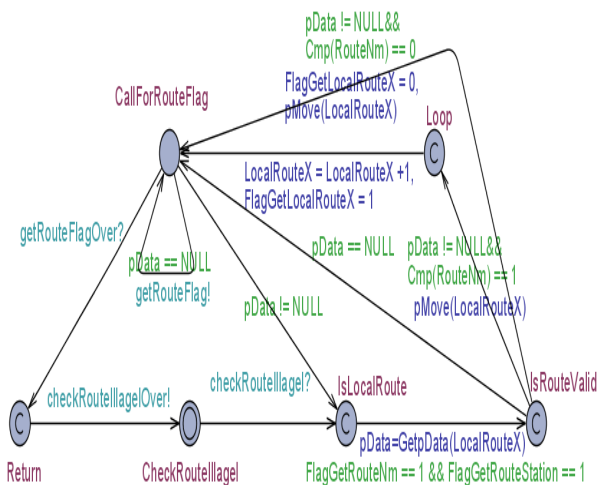


Figure 2(b) The TA model of ‘detecting effectiveness of the approach’

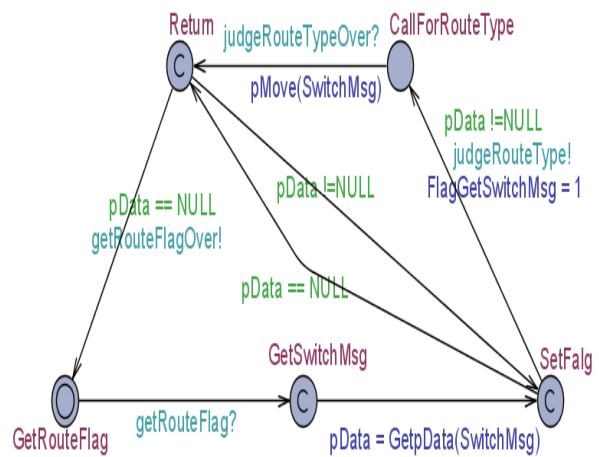


Figure 2(c) The TA model of ‘Getting the approach flag’

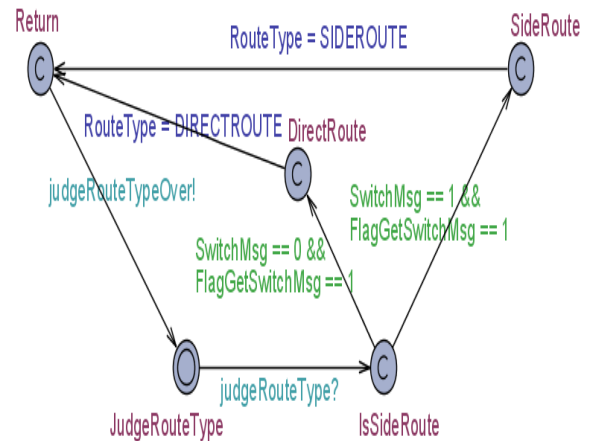


Figure 2(d) The TA model of ‘determining type of the approach’

Where the identifier at the end of the signal “!” means the signal is sent once the conversion occurs. And the identifier at the end of the signal “?” represents the conversion occurs once the signal is received.

The U&A_CIMsgHandle model achieves parsing of CI information packet and judgment of the approach type in Figure 2(a, b, c, d). The model in figure 2(a) obtains the opening, locking and unlocking state of approach from CI information packet. The model in figure 2(b) judges the effectiveness of the approach by comparing the approach table. It is mainly to prevent interlock information error and handle errors as local TCC does not recognize the approach. The model in figure 2(c) is used to obtain approach flag, including the pros or cons information of turnouts and the signal information along with the approach etc. The model in Figure 2(d) gets the approach’s type by comparing approach flag, such as pick-up/depart approach, main line/side line approach.

IV. SIMULATION AND VERIFICATION FOR UML&TA MODEL

A. Simulation and Verification of the Model

UPPAAL is an integrated tool for modeling, validation, verification of the real-time systems modeled as TA networks. Its typical application areas are real-time controllers and those where timing aspects are critical in particular. Once the UPPAAL is used to simulate the system, specific or random running process in system can be observed. Interactions between each model along and system running process could be displayed as well as the location, time and state changing of single model in the run-time.

Simulation for model test is incomplete. That means it could prove the presence of errors, but not able to find the errors which do not occur in simulation. As we all know, it is unrealistic to do endless model simulation. By use UPPAAL, we can find some obvious errors in TCC model at the beginning of verification. However, the UPPAAL is used to validate subtle and complex conversions in the models as well as validate system's characteristics based on quickly search of entire system state space in this section.

BNF (Backus-Naur Form) language is used to verify the functional of TCC model in UPPAAL. Functional verification of TCC software is divided into logic, sequence and fault-security three parts. However, the main verification functions and formally logical expression according to BNF grammars are as follows:

1. Logical functions:

- 1) All the occupied sections are encoded when there are multiple occupied sections:
 $E \langle \text{CheckTwoOccGap.EncodeGap} \quad \text{imply}(\text{GetOccMsgFromTS.OccNumXX} > 1)$
- 2) Protective codes are encoded for the pit mouth without approaching:
 $E \langle \text{EncodeFirstOccRear.EncodeHU} \quad \text{imply} (\text{GetRouteMsg.FlagGetRouteStation} = 0)$
- 3) Encoding message for active transponder according to the departure routes:
 $E \langle (\text{FunMsgBuilder.BuildETCSS}_3_FJZ \&\& \text{FunMsgBuilder.BuildCTCS2}_3_FJZ) \text{ imply} (\text{ScnMsgBuilder.SideOut})$

2. Fault-security functions:

- 1) If it is failure to get segment information at the time of track circuit encoding, then alarm and exit:
 $(\text{GetOccMsgFromTS.GetOccNum} \&\& \text{GetOccMsgFromTS.getOccNm}) \text{ --} \> (\text{TCC.Exit} \&\& \text{TCC.Alarm})$
- 2) If the occupied section does not meet the three-point check, then alarm and exit:
 $((\text{GetOccID.CmpOccNm}() - \text{Temp.OccIndex}) \geq 2) \text{ --} \> (\text{TCC.Exit} \&\& \text{TCC.Alarm})$

3. Sequence functions:

- 1) Encoding track circuit after the track segment and the approaching information have been gotten:
 $A[](\text{not}(\text{GetOccIDAhead.CheckOccNum})) \text{ imply} (\text{not}(\text{GetOccMsgFromTS.Return} \&\& \text{GetRouteMsg.Return}))$

- 2) The time of receiving approach is earlier than the time of station encoding:
 $A[](\text{not}(\text{GetOccIDAhead.CheckOccNum})) \text{ imply} (\text{not}(\text{GetRouteMsg.Return}))$

The formal logical expressions corresponding to each function point are imported in UPPAAL. Then the UPPAAL validates each function by quickly searching state space in models. At last, the verification results indicate that the above logical expressions are satisfied as shown in figure 3:

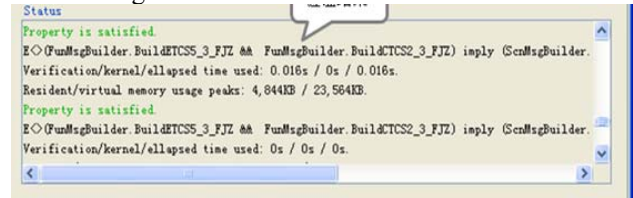


Figure3. The verification results in UPPAAL

B. Inferring and Detecting of the TCC Software Error

Error analysis of software's verification results is helpful to find more problems which are existing in the software. Further on, we could take actions on the error protection of software. As security software in the train control system, TCC software has higher quality requirements than other software. This paper detects the following types of inferred errors for TCC software in this paper:

1. Logical errors

Logical errors express as deadlock in model validation which is detected by statement 'A[] not deadlock'. If deadlock appears in model validation, logical errors should be considered at first.

2. Pointer defense class

In order to illustrate pointer defense issue, validation of TA model 'Getting segment information from current line' is shown as an example. The following statement is used to verify:

```
E <> (GetCurDrcBlockName.Return imply GetCurDrcBlockName.pData == NULL);
```

The verification result is 'Property is not satisfied'. This error manifest as failing to obtain track segment information in TCC software, which makes pointer of track section become empty. However, the following results will become unpredictable. System halted is optimistic scenario. So it is essential for security software to do error inference detection. 'If pData is empty, then jump' is used to do error protection in 'getting segment information of the current line TA model' in figure 4.

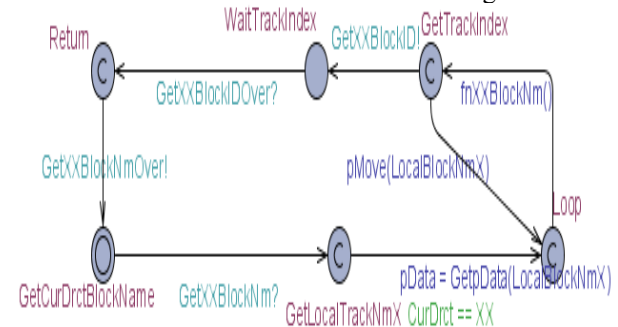


Figure 4(a) Unprotected model of 'getting segment information of the current line

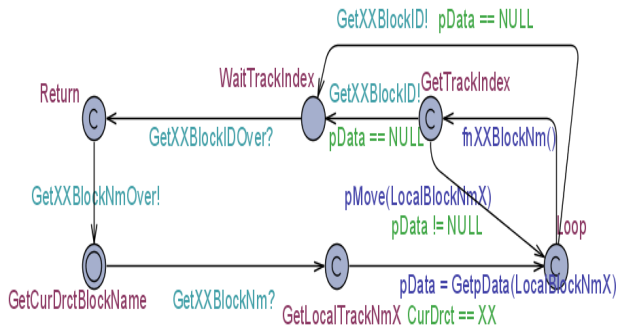


Figure 4(b) defense model of 'getting segment information of the current line'

3. Unreachable branch

Unreachable branch can be detected by statement 'A [] not deadlock'. Incomplete consideration of software in design process would lead to unreachable branch. If the conditions are set incomplete, UPPAAL can verify as being deadlock.

We take the TA model of 'detecting first occupied' as an example to show this software error. The validation result of the model shown in figure 5(a) is deadlock. According to single-step debugging in the UPPAAL simulator, we can know that the position of deadlock in the state 'Count'. $Cmp(IsLocalOcc)$ is identically equal to 1 and $LocalBlockIDX$ is identically equal to 0. When the deadlock occurs, it leads to that transfer conditions from the 'Count' state to any state can't be satisfied.

When the state is in deadlock, the detection section count variable $LocalBlockIDX$ is equal to 0 after the detection of section in jurisdiction have been completed and $Cmp(IsLocalOcc)$ is equal to 1 if an occupied section has been detected. There is no corresponding branch and processing when TCC software is in the above described state. Once the detection of section in jurisdiction has been completed and occupied section was detected, $FlagIsLocalOcc$ should be assigned for detecting occupation and the current state should be transferred to the return state. Therefore, the modified model is shown in Figure 5(b).

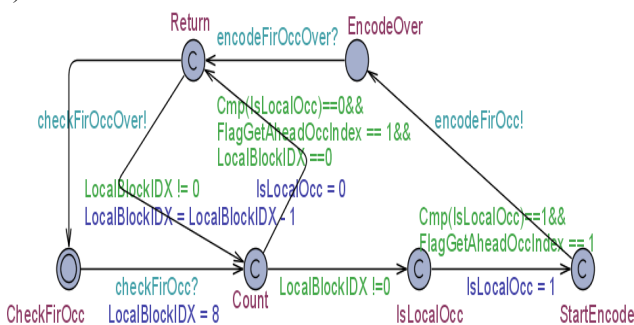


Figure 5(a) The TA model of 'detecting first occupied'

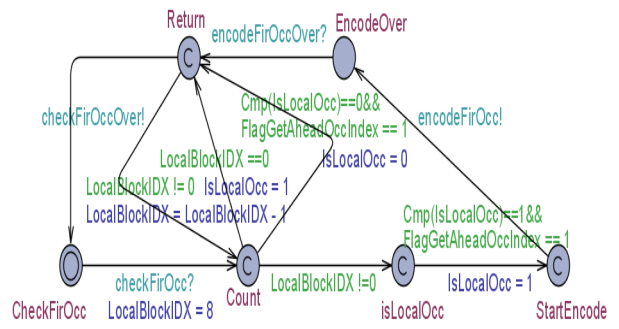


Figure 5(b) The revised TA model of 'detecting first occupied'

V. SUMMARY

UML&TA modeling method are combined together to model track circuit encoding process in TCC software. It can effectively avoid the subjectivity and uncertainty of model which is caused by timed automata modeling used alone. The object-oriented features in UML and the automatic validation function in TA are fully utilized in this combining method. UML&TA models with higher convincing degrees are obtained through UML models. Validation of UML&TA models is divided into logic, sequence and failure-security, which make the verification more comprehensive. In order to solve software encoding problem, this paper presents an error inference and detection method which improves the performance of TCC software.

Firstly, this paper established UML class diagrams and UML state diagrams for track circuit encoding process in TCC software. Secondly, UML&TA models were established according to corresponding rules between the built UML model and TA. Finally, this paper formalized the statement verification on UML&TA models by quickly searching of model state space. In addition, this paper inferred and detected three types of errors which are likely to exist in track circuit encoding process.

ACKNOWLEDGMENT

This work is partially supported by the State Key Laboratory of Rail Traffic Control and Safety under Grant RCS2014ZT06, by Fundamental Research Funds for the Central Universities under Grant 2012JBM016, by State Key Laboratory of Rail Traffic Control and Safety under Grant RCS2014ZZ02.

REFERENCES

- [1] Ning B, Tang T, Qiu K, et al. CTCS-Chinese Train Control System. *Advanced Train Control Systems* (2004): 1.
- [2] Ministry of Railway. *Passenger Line train control center technical specifications, V0.6*. 2008.
- [3] Hairong Dong, Bin Ning, Baigen Cai, Zhongsheng Hou. Automatic train control system development and simulation for high-speed railways. *Circuits and Systems Magazine, IEEE*, 2010, 10(2): 6-18.
- [4] Hui Du, Weiting Yu. A MEASUR and RUP Combined Business Modeling Method. *Journal of Computers* 6.6(2011).
- [5] Lars Jansen, M. Meyer Zu Horste, E.Schnieder. Technical issues in modelling the European Train Control System

- (ETCS) using coloured Petri nets and the Design/CPN tools. (1998).
- [6] MM Zu Hörste, E Schnieder. Modelling and simulation of train control systems using Petri nets. FMRail Workshop. 1999, 3.
- [7] Wumei Tang, Bin Ning, Tianhua Xu, Lin Zhao. Scenario-based modeling and verification for CTCS-3 system requirement specification. Computer Engineering and Technology (ICCET). 2010 2nd International Conference on. IEEE, 2010, 1: V1-400-V1-403.
- [8] Yu Liu, Tao Tang. Research on the Method of Interoperability Test for the Onboard Equipment of CTCS-3 Train Control System of Chinese Railway. Autonomous Decentralized Systems (ISADS), 2011 10th International Symposium on. IEEE, 2011: 415-419.
- [9] Liu Shao Yin. Software development formal engineering methods. Tsinghua University Press, 2008.
- [10] Peng Yu Jing. Complex system requirements for formal. Yangzhou University, 2009.
- [11] Zhang You Sheng, Li Xiong. Research on software development model. Computer Engineering and Applications, 2006, 42(3): 109-115.
- [12] Zhang Hai Fan. Introduction to Software Engineering. Tsinghua University Press, 1998.
- [13] Gao Rui . Railway system in of CTCS the application. Scientific era, 2012 (11): 93-93.
- [14] Ji Xue Sheng, Li Kai Cheng. System Assessment of CTCS-3 Train Control System. Railway Signaling & Communication, 2009 (6): 1-5.
- [15] Zhang Shu Guang. Overall technical program of CTCS-3 Train Control System[M]. Beijing: China Railway Press, 2008:122-123.
- [16] Chunyu Miao. Dynamic Slicing Research of UML Statechart Specifications. Journal of Computers 6.4 (2011).
- [17] Zhao Wei, Chunhe Xia, Yang Luo, Xiaochen Liu, Weikang Wu. An Approach for Description of Computer Network Defense Scheme and Its Simulation Verification. Journal of Computers 9.2 (2014).
- [18] Di Tommaso Pasquale, Francesco Flammini, Armando Lazzaro, Raffaele Pellicchia, Angela Sanseviero. The simulation of anomalies in the functional testing of the ERTMS/ETCS trackside system. High-Assurance Systems Engineering, 2005. HASE 2005. Ninth IEEE International Symposium on. IEEE, 2005: 131-139.
- [19] Zhu Hui Zhong, Zhang Ya Ping, Jiang Xiao Bing. GSM-R communication technology and application. Railway Press, 2005.
- [20] Xiangzheng Xu. Research of Electrified Railway Harmonic Suppression Based on H-infinity Control. Journal of computers 7.12 (2012).

Lei Yuan was born in 1978. He received the M.S. degree in School of Electronic and Information Engineering, Beijing Jiaotong University, China in 2004. He is now working at the State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing. His research interests include the method of design, simulation and test of train control system.

Lijuan Wang was born in 1989. She received the B.S. degree in College of Information Science and Technology, Southwest Jiaotong University, China in 2012. She is working toward the M.S. degree with the Electronics and Information Engineering, Beijing Jiaotong University. Her research interests include machine learning and high-speed railways.

Dewang Chen was born in 1976. He received the B.S. degree in Mechanical and Electrical Engineering and the M.S. degree in Control and Automation from Harbin Engineering University in 1998 and 2000, respectively, and the Ph.D. degree in Control Theory and Control Engineering from the Institute of Automation, Chinese Academy of Sciences in 2003. He is a Professor with the State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University since 2011. His current research interests include intelligent control, machine learning, soft computing, optimization and their applications in intelligent transportation systems and railway systems.

Amie R. Albrecht is a Lecturer in Operations Research in the School of Information Technology and Mathematical Sciences at the University of South Australia. She obtained a B.IT (Honors) in computing and mathematics (2002) and a Ph.D. in mathematics (2009) from the same institution. Amie is a member of the Scheduling and Control Group (SCG) in the Centre for Industrial and Applied Mathematics (CIAM). Her predominant research focus is on improving the operational and energy efficiency of rail transport. This includes: design and automated scheduling of train services and track maintenance, infrastructure design, and improving timekeeping and reducing energy consumption with optimal train driving strategies.